User's Manual

32

RX671 Group

User's Manual: Hardware

RENESAS 32-Bit MCU
RX Family／RX600 Series

Renesas Electronics
www.renesas.com

Rev.1.10 Apr 2022

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving 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, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, 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 and impractical, you are responsible for evaluating the safety of the final products or systems 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. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

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

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

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time 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 time 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 time 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 time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance 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 the 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.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is 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. Additionally, 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.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

How to Use This Manual

1. Objective and Target Users

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

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

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

The following documents have been prepared for the RX671 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	RX671 Group Datasheet	R01DS0373EJ
User's manual: Hardware	Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, and timing charts) and descriptions of operation	RX671 Group User's manual: Hardware	This User's manual
User's manual: Software	Detailed descriptions of the CPU and instruction set	RX Family RXv3 Instruction Set Architecture User's Manual: Software	R01US0316EJ
Application Note	Notes on Printed Circuit Board Patterns	RX Family Hardware Design Guide	R01AN1411EJ
	Examples of applications and sample programs	—	—
Renesas Technical Update	Preliminary report on the specifications of a product, document, etc.	—	—

2. Description of Registers

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

X.X.X ... Register

Address(es): xxxx xxxxh

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

x: Undefined

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

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

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

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

3. List of Abbreviations and Acronyms

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

Contents

R01UH0899EJ0110
Rev.1.10
Apr 15, 2022

120-MHz 32-bit RX MCU, on-chip double-precision FPU, 707 CoreMark,
up to 2-MB flash memory (supporting the dual bank function), 384-KB SRAM,
various communications interfaces, including SD host interface, Quad SPI, and CAN, Capacitive touch sensing unit,
12-bit A/D converter, RTC, Encryption function, Serial sound interface, Remote control signal receiver

Features

■ 32-bit RXv3 CPU core
- Maximum operating frequency: 120 MHz
  Capable of 707 CoreMark in operation at 120 MHz
- Double-precision 64-bit IEEE-754 floating point
- A collective register bank save function is available.
- Supports the memory protection unit (MPU)
- JTAG and FINE (one-line) debugging interfaces

■ Low-power design and architecture
- Operation from a single 2.7- to 3.6-V supply
- Battery supply of backup power allows continued operations of the RTC and the backup registers.
- Four low-power modes

■ On-chip code flash memory
- Supports versions with up to 2 Mbytes of ROM
- No wait cycles at up to 60 MHz or when the ROM cache is hit, one-wait state at up to 120 MHz
- User code is programmable by on-board or off-board programming.
- Programming/erasing as background operations (BGOs)
- A dual-bank structure allows exchanging the start-up bank.

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

■ On-chip SRAM
- 384 Kbytes of SRAM (no wait states)
- 4 Kbytes of standby RAM (backup on deep software standby)

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

■ Data transfer
- DMACAb: 8 channels
- DTCb: 1 channel
- EXDMACa: 2 channels

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

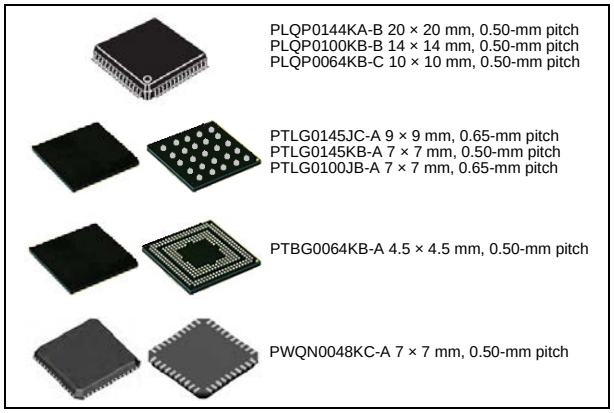
■ Clock functions
- External crystal resonator or internal PLL for operation at 8 to 24 MHz
- A sub-clock oscillator connectable to a 32.768-kHz crystal resonator
- Internal 240-kHz LOCO and HOCO selectable from 16, 18, and 20 MHz
- 120-kHz clock for the IWDTa

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

■ Independent watchdog timer
- Operates with the 120-kHz clock frequency generated by the dedicated low-speed oscillator

■ Useful functions for IEC60730 compliance
- Oscillation-stoppage detection, frequency measurement, CRCA, IWDTa, self-diagnostic function for the A/D converter, etc.
- Register write protection function that protects important registers against overwriting

■ Remote control signal receiver



■ Various communications interfaces
- PHY layer (up to 2 channels) for host/function or OTG controller (1 channel) with full-speed USB 2.0 transfer
- CAN (compliant with ISO11898-1), incorporating 32 mailboxes (up to 2 channels)
- SCIk and SCIh with multiple functionalities (up to 13 channels)
  Choose from among asynchronous mode, clock-synchronous mode, smart-card interface mode, simplified SPI, simplified I^2C, and extended serial mode.
- SCIm with 16-byte transmission and reception FIFOs (up to 2 channels)
- Up to two RSCIs with Manchester encoding and HBS functionality
- The I^2C bus interfaces RIIC and RIICHS for transfer at up to 3.4 Mbps (up to 3 channels), and the RIICHS also supports high-speed mode.
- Single I/O RSPId (3 channels), single I/O RSPIA (1 channel), and quad QSPIX (1 channel). The QSPIX supports fetching from serial flash memory.
- SD host interface (1 channel) with a 1- or 4-bit SD bus for use with SD memory or SDIO
- Serial sound interface supporting various audio data formats, including I^2S

■ Up to 25 extended-function timers
- 16-bit TPUa, MTU3a
- 8-bit TMRb (4 channels), 16-bit CMT (4 channels), 32-bit CMTW (2 channels)

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

■ Temperature sensor for measuring temperature within the chip

■ Capacitive touch sensing unit
- Self-capacitance method: A single pin configures a single key, supporting up to 17 keys
- Mutual capacitance method: Matrix configuration with 17 pins, supporting up to 64 keys

■ Encryption function
- Trusted Secure IP (TSIP)
  AES128/192/256, TDES, ARC4, RSA, ECC,
  True-random number generator (TRNG), SHA1, SHA224, SHA256, MD5, GHASH, Prevention of the illicit copying of keys

■ Up to 114 pins for general I/O ports
- 5-V tolerance, open drain, input pull-up, switchable driving ability

■ Operating temp. range
- D-version: –40°C to +85°C
- G-version: –40°C to +105°C

1. Overview

1.1 Outline of Specifications

Table 1.1 lists the specifications in outline, and Table 1.2 give a comparison of the functions of products in different packages.
Table 1.1 is an outline of maximum specifications, and the peripheral modules and the number of channels of the modules differ depending on the number of pins on the package. For details, see Table 1.2, Comparison of Functions for Different Packages.

Table 1.1 Outline of Specifications (1/10)

Classification	Module/Function	Description
CPU	CPU	• Maximum operating frequency: 120 MHz • 32-bit RX CPU (RXv3) • Minimum instruction execution time: One instruction per state (cycle of the system clock) • Address space: 4-Gbyte linear • Register set of the CPU General purpose: Sixteen 32-bit registers Control: Ten 32-bit registers Accumulator: Two 72-bit registers • 113 instructions Instructions installed as standard: 111 Basic instructions: 77 Single-precision floating-point operation instructions: 11 DSP instructions: 23 Instructions for register bank save function: 2 • Addressing modes: 11 • Data arrangement Instructions: Little endian Data: Selectable as little endian or big endian • On-chip 32-bit multiplier: 32 × 32 → 64 bits • On-chip divider: 32 / 32 → 32 bits • Barrel shifter: 32 bits
	FPU	• Single precision (32-bit) floating point • Data types and floating-point exceptions in conformance with the IEEE754 standard
	Double-precision floating point coprocessor	• Double-precision floating-point register set Double-precision floating-point data registers: 16, each with 64-bit width Double-precision floating-point control registers: Four, each with 32-bit width • Double-precision floating-point processing instructions: 21 • Notifying the interrupt controller of double-precision floating-point exceptions
	Register bank save function	• Fast collective saving and restoration of the values of CPU registers • 16 save register banks
Memory	Code flash memory	• Capacity: 1 Mbyte/1.5 Mbytes/2 Mbytes • ROM cache: 8 Kbytes • 60 MHz ≤ No-wait cycle access 120 MHz ≤ 1-wait cycle access • Instructions hitting the ROM cache or operand = 120 MHz: No-wait access • On-board programming: Four types • Off-board programming (parallel programmer mode) • Instructions are executable only for the program stored in the TM target area by using the Trusted Memory (TM) function and protection against data reading is realized. • A dual-bank structure allows programming during reading or exchanging the start-up areas
	Data flash memory	• Capacity: 8 Kbytes • Programming/erasing: 100,000 times
	Unique ID	• 16-byte unique ID for the device
	RAM	• Capacity: 384 Kbytes • 120 MHz, no-wait access
	Standby RAM	• Capacity: 4 Kbytes • Operation synchronized with PCLKB: Up to 60 MHz, two-cycle access

Table 1.1 Outline of Specifications (2/10)

Classification	Module/Function	Description
Operating modes		• Operating modes by the mode-setting pins at the time of release from the reset state Single-chip mode Boot mode (for the SCI interface) Boot mode (for the USB interface) Boot mode (for the FINE interface) • Selection of operating mode by register setting Single-chip mode On-chip ROM disabled extended mode On-chip ROM enabled extended mode • Endian selectable
Clock	Clock generation circuit	• Main clock oscillator, sub clock oscillator, low-speed/high-speed on-chip oscillator, PLL frequency synthesizer, and IWDT-dedicated on-chip oscillator • The peripheral module clocks can be set to frequencies above that of the system clock. • Main-clock oscillation stoppage detection • Separate frequency-division and multiplication settings for the system clock (ICLK), peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD), flash-IF clock (FCLK) and external bus clock (BCLK) The CPU and other bus masters run in synchronization with the system clock (ICLK): Up to 120 MHz QSPIX runs in synchronization with ICLK at Up to 120 MHz. Peripheral modules of MTU, RSPI, SCIm, RSPIA, RSCI, and RIICHS run in synchronization with PCLKA, which operates at up to 120 MHz. Other peripheral modules run in synchronization with PCLKB: Up to 60 MHz ADCLK in the S12AD (unit 0) runs in synchronization with PCLKC: Up to 60 MHz ADCLK in the S12AD (unit 1) runs in synchronization with PCLKD: Up to 60 MHz Flash IF run in synchronization with the flash-IF clock (FCLK): Up to 60 MHz Devices connected to the external bus run in synchronization with the external bus clock (BCLK): Up to 60 MHz • Multiplication is possible with using the high-speed on-chip oscillator (HOCO) as a reference clock of the PLL circuit • External clock input frequency: 30 MHz (max) • Clock output function
Reset		Nine types of reset • RES# pin reset: Generated when the RES# pin is driven low. • Power-on reset: Generated when the RES# pin is driven high and VCC = AVCC0 = AVCC1 rises. • Voltage-monitoring 0 reset: Generated when VCC = AVCC0 = AVCC1 falls. • Voltage-monitoring 1 reset: Generated when VCC = AVCC0 = AVCC1 falls. • Voltage-monitoring 2 reset: Generated when VCC = AVCC0 = AVCC1 falls. • Deep software standby reset: Generated in response to an interrupt to trigger release from deep software standby. • Independent watchdog timer reset: Generated when the independent watchdog timer underflows, or a refresh error occurs. • Watchdog timer reset: Generated when the watchdog timer underflows, or a refresh error occurs. • Software reset: Generated by register setting.
Power-on reset		If the RES# pin is at the high level when power is supplied, an internal reset is generated. After VCC = AVCC0 = AVCC1 has exceeded the voltage detection level and the specified period has elapsed, the reset is cancelled.
Voltage detection circuit (LVDA)		Monitors the voltage being input to the VCC = AVCC0 = AVCC1 pins and generates an internal reset or interrupt. • Voltage detection circuit 0 Capable of generating an internal reset The option-setting memory can be used to select enabling or disabling of the reset. Voltage detection level: Selectable from three different levels (2.94 V, 2.87 V, 2.80 V) • Voltage detection circuits 1 and 2 Voltage detection level: Selectable from three different levels (2.99 V, 2.92 V, 2.85 V) Digital filtering (1/2, 1/4, 1/8, and 1/16 LOCO frequency) Capable of generating an internal reset • Two types of timing are selectable for release from reset An internal interrupt can be requested. • Detection of voltage rising above and falling below thresholds is selectable. • Maskable or non-maskable interrupt is selectable Voltage detection monitoring Event linking

Table 1.1 Outline of Specifications (3/10)

Classification	Module/Function	Description
Low power consumption	Low power consumption function	• Module stop function • Four low power consumption modes Sleep mode, all-module clock stop mode, software standby mode, and deep software standby mode
	Battery backup function	• When the voltage on the VCC pin drops, power can be supplied to the backup power area from the VBATT pin. • Backup power area: Sub-clock oscillator Realtime clock Backup register Tamper detection • Detection of low voltage in the backup power area
Interrupt	Interrupt controller (ICUE)	• Peripheral function interrupts: 256 sources • External interrupts: 16 (pins IRQ0 to IRQ15) • Software interrupts: 2 sources • Non-maskable interrupts: 8 sources • Sixteen levels specifiable for the order of priority • Method of interrupt source selection: The interrupt vectors consist of 256 vectors (128 sources are fixed. The remaining 133 vectors are selected from among the other 128 sources.)
External bus extension		• The external address space can be divided into eight areas (CS0 to CS7), each with independent control of access settings. Capacity of each area: 16 Mbytes (CS0 to CS7) A chip-select signal (CS0# to CS7#) can be output for each area. Each area is specifiable as an 8-, or 16-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 (DMACAb)	• 8 channels • Three transfer modes: Normal transfer, repeat transfer, and block transfer • Activation sources: Software trigger, external interrupts, and interrupt requests from peripheral functions • Transfer space: 4 Gbytes (0000 0000h to FFFF FFFFh excluding reserved areas)
	EXDMA controller (EXDMACa)	• 2 channels Four transfer modes: Normal transfer, repeat transfer, block transfer, and cluster transfer • Single-address transfer enabled with the EDACKn signal • Request sources: Software trigger, external DMA requests (EDREQn), and interrupt requests from peripheral functions
	Data transfer controller (DTCb)	• Three transfer modes: Normal transfer, repeat transfer, and block transfer • Request sources: External interrupts and interrupt requests from peripheral functions • Sequence transfer

Table 1.1 Outline of Specifications (4/10)

Classification	Module/Function	Description
I/O ports	Programmable I/O ports	• I/O ports for the 145-pin TFLGA (0.65-mm pitch) I/O pins: 111 Input pin: 1 Pull-up resistors: 111 Open-drain outputs: 111 5-V tolerance: 20 • I/O ports for the 145-pin TFLGA (0.50-mm pitch) and 144-pin LFQFP I/O pins: 113 Input pin: 1 Pull-up resistors: 113 Open-drain outputs: 113 5-V tolerance: 20 • I/O ports for the 100-pin TFLGA and 100-pin LFQFP I/O pins: 80 Input pin: 1 Pull-up resistors: 80 Open-drain outputs: 80 5-V tolerance: 18 • I/O ports for the 64-pin TFBGA I/O pins: 43 Input pin: 1 Pull-up resistors: 43 Open-drain outputs: 43 5-V tolerance: 8 • I/O ports for the 64-pin LFQFP I/O pins: 44 Input pin: 1 Pull-up resistors: 44 Open-drain outputs: 44 5-V tolerance: 8 • I/O ports for the 48-pin HWQFN I/O pins: 33 Input pin: 1 Pull-up resistors: 33 Open-drain outputs: 33 5-V tolerance: 6
Event link controller (ELC)		• Event signals such as interrupt request signals can be interlinked with the operation of functions such as timer counting, eliminating the need for intervention by the CPU to control the functions. • 99 internal event signals can be freely combined for interlinked operation with connected functions. • Event signals from peripheral modules can be used to change the states of output pins (of ports B and E). • Changes in the states of pins (of ports B and E) being used as inputs can be interlinked with the operation of peripheral modules.
Timers	16-bit timer pulse unit (TPUa)	• (16 bits × 6 channels) × 1 unit • Maximum of 16 pulse-input/output possible • Select from among seven or eight counter-input clock signals for each channel • Input capture/output compare function • Output of PWM waveforms in up to 15 phases in PWM mode • Support for buffered operation, phase-counting mode (two phase encoder input) and cascade-connected operation (32 bits × 2 channels) depending on the channel. • PPG output trigger can be generated • Capable of generating conversion start triggers for the A/D converters • Digital filtering of signals from the input capture pins • Event linking by the ELC
	8-bit timers (TMRb)	• (8 bits × 2 channels) × 2 units • Select from among seven internal clock signals (PCLKB/1, PCLKB/2, PCLKB/8, PCLKB/32, PCLKB/64, PCLKB/1024, PCLKB/8192) and one external clock signal • Capable of output of pulse trains with desired duty cycles or of PWM signals • The 2 channels of each unit can be cascaded to create a 16-bit timer • Generation of triggers for A/D converter conversion • Capable of generating baud-rate clocks for SCI5, SCI6, and SCI12 • Capable of generating operating clock for the remote control signal receiver (REMC) • Event linking by the ELC

Table 1.1 Outline of Specifications (5/10)

Classification	Module/Function	Description
Timers	Compare match timer (CMT)	• (16 bits × 2 channels) × 2 units • Select from among four internal clock signals (PCLKB/8, PCLKB/32, PCLKB/128, PCLKB/512) • Event linking by the ELC
	Compare match timer W (CMTW)	• (32 bits × 1 channel) × 2 units • Compare-match, input-capture input, and output-comparison output are available. • Select from among four internal clock signals (PCLKB/8, PCLKB/32, PCLKB/128, PCLKB/512) • Interrupt requests can be output in response to compare-match, input-capture, and output-comparison events. • Event linking by the ELC
	Watchdog timer (WDTA)	• 14 bits × 1 channel • Select from among 6 counter-input clock signals (PCLKB/4, PCLKB/64, PCLKB/128, PCLKB/512, PCLKB/2048, PCLKB/8192)
	Independent watchdog timer (IWDTa)	• 14 bits × 1 channel • Counter-input clock: IWDT-dedicated on-chip oscillator • Dedicated clock/1, dedicated clock/16, dedicated clock/32, dedicated clock/64, dedicated clock/128, dedicated clock/256 • Window function: The positions where the window starts and ends are specifiable (the window defines the timing with which refreshing is enabled and disabled). • Event linking by the ELC
	Multifunction timer pulse unit (MTU3a)	• 9 channels (16 bits × 8 channels, 32 bits × 1 channel) • Maximum of 28 pulse-input/output and 3 pulse-input possible • Select from among 14 counter-input clock signals for each channel (PCLKA/1, PCLKA/2, PCLKA/4, PCLKA/8, PCLKA/16, PCLK/A32, PCLKA/64, PCLKA/256, PCLKA/1024, MTCLKA, MTCLKB, MTCLKC, MTCLKD, MTIOC1A) 14 of the signals are available for channel 0, 11 are available for channels 1, 3, 4, 6 to 8, 12 are available for channel 2, and 10 are available for channel 5. • Input capture function • 39 output compare/input capture registers • Counter clear operation (synchronous clearing by compare match/input capture) • Simultaneous writing to multiple timer counters (TCNT) • Simultaneous register input/output by synchronous counter operation • Buffered operation • Support for cascade-connected operation • 43 interrupt sources • Automatic transfer of register data • Pulse output mode Toggle/PWM/complementary PWM/reset-synchronized PWM • Complementary PWM output mode Outputs non-overlapping waveforms for controlling 3-phase inverters Automatic specification of dead times PWM duty cycle: Selectable as any value from 0% to 100% Delay can be applied to requests for A/D conversion. Non-generation of interrupt requests at peak or trough values of counters can be selected. Double buffer configuration • Reset synchronous PWM mode Three phases of positive and negative PWM waveforms can be output with desired duty cycles. • Phase-counting mode: 16-bit mode (channels 1 and 2); 32-bit mode (channels 1 and 2) • Counter functionality for dead-time compensation • Generation of triggers for A/D converter conversion • A/D converter start triggers can be skipped • Digital filter function for signals on the input capture and external counter clock pins • PPG output trigger can be generated • Event linking by the ELC
	Port output enable 3 (POE3a)	• Control of the high-impedance state of the MTU waveform output pins • 5 pins for input from signal sources: POE0#, POE4#, POE8#, POE10#, POE11# • Initiation on detection of short-circuited outputs (detection of simultaneous PWM output to the active level) • Initiation by oscillation-stoppage detection or software • Additional programming of output control target pins is enabled

Table 1.1 Outline of Specifications (6/10)

Classification	Module/Function	Description
Timers	Programmable pulse generator (PPG)	• (4 bits × 4 groups) × 2 units • Pulse output with the MTU or TPU output as a trigger • Maximum of 32 pulse-output possible
	Realtime clock (RTCd)*1	• Clock sources: Main clock, sub clock • Selection of the 32-bit binary count in time count/second unit possible • Clock and calendar functions • Interrupt sources: Alarm interrupt, periodic interrupt, and carry interrupt • Battery backup operation • Time capture function (up to 3 pins) • Event linking by the ELC
Communication function	USB 2.0 FS host/ function module (USBb)	• Includes a UDC (USB Device Controller) and transceiver for USB 2.0 FS • Up to two ports • Compliance with the USB 2.0 specification • Transfer rate: Full speed (12 Mbps), low speed (1.5 Mbps) (host only) • Both self-powered mode and bus-powered mode are supported • OTG (On the Go) operation is possible (low-speed is not supported) • Incorporates 2 Kbytes of RAM as a transfer buffer • External pull-up and pull-down resistors are not required
	Serial communications interfaces (SCIk, SCIm, SCIh)	• 13 channels (SCIk: 10 channels + SCIh: 1 channel + SCIm: 2 channels) • SCIk, SCIh, SCIm 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 Start-bit detection: Level or edge detection is selectable. Simple I^2C Simple SPI 9-bit transfer mode Bit rate modulation Double-speed mode • SCIk, SCIh Average transfer rate clock can be input from TMR timers for SCI5, SCI6, and SCI12 Event linking by the ELC (only on channel 5) • SCIh Supports the serial communications protocol, which contains the start frame and information frame Supports the LIN format • SCIm Data can be transmitted or received in sequence by the 16-byte FIFO buffers of the transmission and reception unit • SCIk, SCIm Data match detection Adjustment of the timing of sampling of the RXD signals

Table 1.1 Outline of Specifications (7/10)

Classification	Module/Function	Description
Communication function	Serial communications interfaces (RSCI)	• 2 channels (RSCI10, RSCI11) • 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 • Start-bit detection: Level or edge detection is selectable. • Simple I^2C • Simple SPI • 9-bit transfer mode • Bit rate modulation • Double-speed mode • Event linking by the ELC (only RSCI10) • Supports the serial communications protocol, which contains the start frame and information frame • Supports the LIN format • Data can be transmitted or received in sequence by the 32-byte FIFO buffers of the transmission and reception unit • Manchester encoding is supported. • HBS (home bus system) support mode • Data match detection • Adjustment of the timing of sampling of the RXD signals
	I^2C bus interface (RIICa)	• 3 channels (only channel 0 can be used in fast-mode plus) Communication formats I^2C bus format/SMBus format Supports the multi-master Max. transfer rate: 1 Mbps (channel 0) • Event linking by the ELC
	High-speed I^2C bus interface (RIICHS)	• 1 channel Communication formats I^2C bus format/SMBus format Supports the multi-master Max. transfer rate: 3.4 Mbps • Event linking by the ELC
	CAN module (CAN)	• 2 channels • Compliance with the ISO11898-1 specification (standard frame and extended frame) • 32 mailboxes per channel
	Serial peripheral interface (RSPId)	• 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) Transmit/receive data can be swapped in byte units • Buffered structure Double buffers for both transmission and reception • RSPCK can be stopped with the receive buffer full for master reception. • Event linking by the ELC

Table 1.1 Outline of Specifications (8/10)

Classification	Module/Function	Description
Communication function	Serial peripheral interface (RSPIA)	• 1 channel • RSPI transfer facility Using the MOSI (master out, slave in), MISO (master in, slave out), SSL (slave select), and RSPCK (RSPI clock) signals enables serial transfer through SPI operation (four lines) or clock-synchronous operation (three lines) Capable of handling serial transfer as a master or slave • Data formats Switching between MSB first and LSB first The number of bits in each transfer can be changed to any number of bits from 8 to 16, or to 20, 24, or 32 bits. 128-bit buffers for transmission and reception Up to four frames can be transmitted or received in a single transfer operation (with each frame having up to 32 bits) Transit/receive data can be swapped in byte units • Buffered structure The transmission and reception sections have 4-stage and 32-bit-wide FIFO buffers for the sequential transmission and reception of data. • RSPCK can be stopped with the receive buffer full for master reception. • Event linking by the ELC • Communications protocol: RSPIA supports the Texas Instruments Synchronous Serial Protocol (TI SSP).
	Quad-SPI memory interface (QSPIX)	• 1 channel • This interface can handle fetching from serial flash memory that has an SPI-compatible interface. • It supports extended SPI, dual-SPI, and quad-SPI protocols. • Address width is selectable from among 8, 16, 24, and 32 bits.
	Remote control signal receiver (REMCa)	• 1 channel • Four pattern matching (header, data 0, data 1, and special data detection) • 8-byte receive buffer per unit • The operating clock can be selected from among the PCLK, sub-clock, and TMR.
Serial sound interface (SSIE)		• 1 channel • Full-duplex transmission • Various types of serial audio formatting are supported. • Master and slave operations are supported. • The bit-clock frequency is selectable from among 13 frequencies (1/1, 1/2, 1/4, 1/6, 1/8, 1/12, 1/16, 1/24, 1/32, 1/48, 1/64, 1/96, or 1/128). • Data formats with 8, 16, 18, 20, 22, 24, and 32 bits are supported. • 32-stage FIFO buffers for transmission and reception • Stopping or not stopping the SSILRCK signal on stopping of data transmission is selectable.
SD host interface (SDHI)*2		• 1 channel • Transfer speed: Supports high-speed mode (25 MB/s) and default speed mode (12.5 MB/s) • One interface for SD memory and I/O cards (supporting 1- and 4-bit SD buses) • SD specifications Part 1: Physical Layer Specification Ver. 3.01 compliant (DDR not supported) Part E1: SDIO Specification Ver. 3.00 • Error checking: CRC7 for commands and CRC16 for data • Interrupt requests: Card access interrupt, SDIO access interrupt, card detection interrupt, SD buffer access interrupt • DMA transfer requests: SD_BUF write and SD_BUF read • Support for card detection and write protection

Table 1.1 Outline of Specifications (9/10)

Classification	Module/Function	Description
12-bit A/D converter (S12ADFa)		• 12 bits × 2 units (unit 0: 8 channels; unit 1: 12 channels) • 12-bit resolution (switchable between 8, 10, and 12 bits) • Conversion time 0.48 µs per channel (for 12-bit conversion) 0.45 µs per channel (for 10-bit conversion) 0.42 µs per channel (for 8-bit conversion) • Operating mode Scan mode (single scan mode, continuous scan mode, or 3 group scan mode) Group priority control (only for 3 group scan mode) • Sample-and-hold function Common sample-and-hold circuit included • Sampling variable Sampling time can be set up for each channel. • Digital comparison Method: Comparison to detect voltages above or below thresholds and window comparison Measurement: Comparison of two results of conversion or comparison of a value in the comparison register and a result of conversion • Self-diagnostic function The self-diagnostic function internally generates three analog input voltages (unit 0: VREFL0, VREFH0 × 1/2, VREFH0; unit 1: AVSS1, AVCC1 × 1/2, AVCC1) • Double trigger mode (A/D conversion data duplicated) • Detection of analog input disconnection • Three ways to start A/D conversion Software trigger, timer (MTU, TMR, TPU) trigger, external trigger • Event linking by the ELC
Temperature sensor		• 1 channel • Relative precision: ± 1°C • The voltage of the temperature is converted into a digital value by the 12-bit A/D converter (unit 1).
Capacitive touch sensing unit (CTSUa)		• Detection pin: 17 channels
Safety	Memory protection unit (MPU)	• Protection area: Eight areas (max.) can be specified in the range from 0000 0000h to FFFF FFFFh. • Minimum protection unit: 16 bytes • Reading from, writing to, and enabling the execution access can be specified for each area. • An access exception occurs when the detected access is not in the permitted area.
	Trusted Memory (TM) Function	• Programs in the TM target area in the code flash memory are protected against reading • Instruction fetching by the CPU is the only form of access to these areas when the TM function is enabled.
	Register write protection function	• Protects important registers from being overwritten for in case a program runs out of control.
	CRC calculator (CRCA)	• Generation of CRC codes for 8-/32-bit data 8-bit data Selectable from the following three polynomials $X^8 + X^2 + X + 1$, $X^{16} + X^{15} + X^2 + 1$, $X^{16} + X^{12} + X^5 + 1$ 32-bit data Selectable from the following two polynomials $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$, $X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$ • Generation of CRC codes for use with LSB-first or MSB-first communications is selectable
	Main clock oscillation stop detection	• Main clock oscillation stop detection: Available
	Clock frequency accuracy measurement circuit (CAC)	• Monitors the clock output from the main clock oscillator, sub-clock oscillator, low- and high-speed on-chip oscillators, IWDT-dedicated on-chip oscillator, and PCLKB, and generates interrupts when the setting range is exceeded.
	Data operation circuit (DOCA)	• This handles the comparison, addition, subtraction, comparison in terms of which is larger or smaller, or window comparison of 32-bit values.

Table 1.1 Outline of Specifications (10/10)

Classification	Module/Function	Description
Encryption function	Trusted Secure IP (TSIP)*3	• Access management circuit • Encryption engine Common key encryption: AES (compliant with NIST FIPS PUB 197), TDES, ARC4 Public key encryption: RSA, ECC Hash functions: SHA1, SHA224, SHA256, MD5, GHASH • Other features TRNG (true-random number generator) Prevention from illicit copying of a key
Operating frequency		Up to 120 MHz
Power supply voltage		VCC = AVCC0 = AVCC1 = VCC_USB = 2.7 to 3.6 V, 2.7 ≤ VREFH0 ≤ AVCC0, V_{BATT} = 1.62*4 to 3.6 V
Operating temperature		D-version: –40 to +85°C G-version: –40 to +105°C
Package		145-pin TFLGA (PTLG0145JC-A) 145-pin TFLGA (PTLG0145KB-A) 144-pin LFQFP (PLQP0144KA-B) 100-pin TFLGA (PTLG0100JB-A) 100-pin LFQFP (PLQP0100KB-B) 64-pin TFBGA (PTBG0064KB-A) 64-pin LFQFP (PLQP0064KB-C) 48-pin HWQFN (PWQN0048KC-A)
Debugging interface		JTAG and FINE interfaces

Note 1. When the realtime clock is not used, initialize the registers in the time clock according to description in section 31.6.8, Initialization Procedure When the Realtime Clock is Not to be Used.
Note 2. The product part number differs according to whether or not the MCU includes an SDHI (SD host interface).
Note 3. The product part number differs according to whether or not the MCU includes the encryption function.
Note 4. The low CL crystal unit cannot be used when the V_{BATT} voltage is less than 2.0 V.

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

Functions	Products	RX671				
	Package	145-pin TFLGA (0.65-mm pitch)	145-pin TFLGA (0.50-mm pitch) 144-pin LFQFP	100-pin TFLGA 100-pin LFQFP	64-pin TFBGA 64-pin LFQFP	48-pin HWQFN
Code Flash Memory	Code Flash Memory Capacity	1 Mbyte/1.5 Mbytes/2 Mbytes				
	Dual bank function	Available				
	BGO function	Available				
Data Flash Memory		8 Kbytes				
RAM		384 Kbytes				
Standby RAM		4 Kbytes				
External bus	External bus width	16/8 bits			Not available	
	SDRAM area controller	Available		Not available		
DMA	DMA controller	Ch. 0 to 7				
	Data transfer controller	Available				
	EXDMA controller	Ch. 0 and 1			Not available	
Timers	16-bit timer pulse unit	Ch. 0 to 5				
	Multi-function timer pulse unit 3	Ch. 0 to 8				
	Port output enable 3	Available				
	Programmable pulse generator	Ch. 0 and 1			Not available	
	8-bit timers	Ch. 0 to 3				
	Compare match timer	Ch. 0 to 3				
	Compare match timer W	Ch. 0 and 1				
	Realtime clock	Available				Not available
	Watchdog timer	Available				
	Independent watchdog timer	Available				
Communication function	USB 2.0 FS host/function module	Ch. 0 and 1	Ch. 0		Ch. 0*1	Not available
	Serial communications interfaces (SCIk)	Ch. 0 to 9		Ch. 0 to 3, 5, 6, 8 and 9	Ch. 1 to 3, 5, 8 and 9	
	Serial communications interfaces (SCIm)	Ch. 10 and 11				
	Serial communications interfaces (SCIh)	Ch. 12				
	Serial communications interfaces (RSCI)	Ch. 10 and 11				
	I^2C bus interfaces (RIIC)	Ch. 0 to 2				Ch. 0 and 2
	Hi-speed I^2C bus interfaces (RIICHS)	Ch. 0				
	Serial peripheral interface (RSPI)	Ch. 0 to 2			Ch. 0 and 1	
	Serial peripheral interface (RSPIA)	Ch. 0				
	CAN module	Ch. 0 and 1			Not available	
	Quad-SPI memory interface (QSPIX)	Ch. 0				
	SD host interface (SDHI)	Available				
	Serial sound interface (SSIE)	Ch. 0				
	Remote control signal receiver (REMC)	Ch. 0				
Capacitive touch sensing unit (CTSU)		17 channels + 1 channel (TSCAP)			8 channels + 1 channel (TSCAP)	6 channels + 1 channel (TSCAP)
12-bit A/D converter		Unit 0: 8 channels Unit 1: 12 channels		Unit 0: 8 channels Unit 1: 8 channels	Unit 0: 4 channels Unit 1: 6 channels	Unit 0: 4 channels Unit 1: 4 channels
Temperature sensor		Available				
CRC calculator		Available				
Data operation circuit (DOC)		Available				

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

Functions	Products	RX671				
	Package	145-pin TFLGA (0.65-mm pitch)	145-pin TFLGA (0.50-mm pitch) 144-pin LFQFP	100-pin TFLGA 100-pin LFQFP	64-pin TFBGA 64-pin LFQFP	48-pin HWQFN
Clock frequency accuracy measurement circuit (CAC)		Available				
Trusted Secure IP		Available/Not available				
Event link controller (ELC)		Available				
Battery backup function		Available				Not available
Backup register		Available				
Off-board programming (parallel programmer mode)		Available			Not available	

Note 1. Only supports the function controller.

1.2 List of Products

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

Table 1.3 List of Products (1/3)

Group	Part No.	Package	Code Flash Memory Capacity (byte(s))	RAM Capacity (byte(s))	Data Flash Memory Capacity (byte(s))	Operating Frequency (Max.)	Encryption Module	Operating temperature (°C)
RX671 (D-version)	R5F5671EHDFB	PLQP0144KA-B	2 M	384 K	8 K	120 MHz	Available	–40 to +85
	R5F5671EDDFB	PLQP0144KA-B	2 M	384 K	8 K	120 MHz	Not available	–40 to +85
	R5F5671CHDFB	PLQP0144KA-B	1.5 M	384 K	8 K	120 MHz	Available	–40 to +85
	R5F5671CDDFB	PLQP0144KA-B	1.5 M	384 K	8 K	120 MHz	Not available	–40 to +85
	R5F56719HDFB	PLQP0144KA-B	1 M	384 K	8 K	120 MHz	Available	–40 to +85
	R5F56719DDFB	PLQP0144KA-B	1 M	384 K	8 K	120 MHz	Not available	–40 to +85
	R5F5671EHDFP	PLQP0100KB-B	2 M	384 K	8 K	120 MHz	Available	–40 to +85
	R5F5671EDDFP	PLQP0100KB-B	2 M	384 K	8 K	120 MHz	Not available	–40 to +85
	R5F5671CHDFP	PLQP0100KB-B	1.5 M	384 K	8 K	120 MHz	Available	–40 to +85
	R5F5671CDDFP	PLQP0100KB-B	1.5 M	384 K	8 K	120 MHz	Not available	–40 to +85
	R5F56719HDFP	PLQP0100KB-B	1 M	384 K	8 K	120 MHz	Available	–40 to +85
	R5F56719DDFP	PLQP0100KB-B	1 M	384 K	8 K	120 MHz	Not available	–40 to +85
	R5F5671EHDFM	PLQP0064KB-C	2 M	384 K	8 K	120 MHz	Available	–40 to +85
	R5F5671EDDFM	PLQP0064KB-C	2 M	384 K	8 K	120 MHz	Not available	–40 to +85
	R5F5671CHDFM	PLQP0064KB-C	1.5 M	384 K	8 K	120 MHz	Available	–40 to +85
	R5F5671CDDFM	PLQP0064KB-C	1.5 M	384 K	8 K	120 MHz	Not available	–40 to +85
	R5F56719HDFM	PLQP0064KB-C	1 M	384 K	8 K	120 MHz	Available	–40 to +85
	R5F56719DDFM	PLQP0064KB-C	1 M	384 K	8 K	120 MHz	Not available	–40 to +85
	R5F5671EHDNE	PWQN0048KC-A	2 M	384 K	8 K	120 MHz	Available	–40 to +85
	R5F5671EDDNE	PWQN0048KC-A	2 M	384 K	8 K	120 MHz	Not available	–40 to +85
	R5F5671CHDNE	PWQN0048KC-A	1.5 M	384 K	8 K	120 MHz	Available	–40 to +85
	R5F5671CDDNE	PWQN0048KC-A	1.5 M	384 K	8 K	120 MHz	Not available	–40 to +85
	R5F56719HDNE	PWQN0048KC-A	1 M	384 K	8 K	120 MHz	Available	–40 to +85
	R5F56719DDNE	PWQN0048KC-A	1 M	384 K	8 K	120 MHz	Not available	–40 to +85
	R5F5671EHDBP	PTBG0064KB-A	2 M	384 K	8 K	120 MHz	Available	–40 to +85
	R5F5671EDDBP	PTBG0064KB-A	2 M	384 K	8 K	120 MHz	Not available	–40 to +85
	R5F5671CHDBP	PTBG0064KB-A	1.5 M	384 K	8 K	120 MHz	Available	–40 to +85
	R5F5671CDDBP	PTBG0064KB-A	1.5 M	384 K	8 K	120 MHz	Not available	–40 to +85
	R5F56719HDBP	PTBG0064KB-A	1 M	384 K	8 K	120 MHz	Available	–40 to +85
	R5F56719DDBP	PTBG0064KB-A	1 M	384 K	8 K	120 MHz	Not available	–40 to +85
	R5F5671EHDLE	PTLG0145JC-A	2 M	384 K	8 K	120 MHz	Available	–40 to +85
	R5F5671EDDLE	PTLG0145JC-A	2 M	384 K	8 K	120 MHz	Not available	–40 to +85
	R5F5671CHDLE	PTLG0145JC-A	1.5 M	384 K	8 K	120 MHz	Available	–40 to +85
	R5F5671CDDLE	PTLG0145JC-A	1.5 M	384 K	8 K	120 MHz	Not available	–40 to +85
	R5F56719HDLE	PTLG0145JC-A	1 M	384 K	8 K	120 MHz	Available	–40 to +85
	R5F56719DDLE	PTLG0145JC-A	1 M	384 K	8 K	120 MHz	Not available	–40 to +85

Table 1.3 List of Products (2/3)

Group	Part No.	Package	Code Flash Memory Capacity (byte(s))	RAM Capacity (byte(s))	Data Flash Memory Capacity (byte(s))	Operating Frequency (Max.)	Encryption Module	Operating temperature (°C)
RX671 (D-version)	R5F5671EHDLK	PTLG0145KB-A	2 M	384 K	8 K	120 MHz	Available	–40 to +85
	R5F5671EDDLK	PTLG0145KB-A	2 M	384 K	8 K	120 MHz	Not available	–40 to +85
	R5F5671CHDLK	PTLG0145KB-A	1.5 M	384 K	8 K	120 MHz	Available	–40 to +85
	R5F5671CDDLK	PTLG0145KB-A	1.5 M	384 K	8 K	120 MHz	Not available	–40 to +85
	R5F56719HDLK	PTLG0145KB-A	1 M	384 K	8 K	120 MHz	Available	–40 to +85
	R5F56719DDLK	PTLG0145KB-A	1 M	384 K	8 K	120 MHz	Not available	–40 to +85
	R5F5671EHDLJ	PTLG0100JB-A	2 M	384 K	8 K	120 MHz	Available	–40 to +85
	R5F5671EDDLJ	PTLG0100JB-A	2 M	384 K	8 K	120 MHz	Not available	–40 to +85
	R5F5671CHDLJ	PTLG0100JB-A	1.5 M	384 K	8 K	120 MHz	Available	–40 to +85
	R5F5671CDDLJ	PTLG0100JB-A	1.5 M	384 K	8 K	120 MHz	Not available	–40 to +85
	R5F56719HDLJ	PTLG0100JB-A	1 M	384 K	8 K	120 MHz	Available	–40 to +85
	R5F56719DDLJ	PTLG0100JB-A	1 M	384 K	8 K	120 MHz	Not available	–40 to +85
RX671 (G-version)	R5F5671EHGFB	PLQP0144KA-B	2 M	384 K	8 K	120 MHz	Available	–40 to +105
	R5F5671EDGFB	PLQP0144KA-B	2 M	384 K	8 K	120 MHz	Not available	–40 to +105
	R5F5671CHGFB	PLQP0144KA-B	1.5 M	384 K	8 K	120 MHz	Available	–40 to +105
	R5F5671CDGFB	PLQP0144KA-B	1.5 M	384 K	8 K	120 MHz	Not available	–40 to +105
	R5F56719HGFB	PLQP0144KA-B	1 M	384 K	8 K	120 MHz	Available	–40 to +105
	R5F56719DGFB	PLQP0144KA-B	1 M	384 K	8 K	120 MHz	Not available	–40 to +105
	R5F5671EHGFP	PLQP0100KB-B	2 M	384 K	8 K	120 MHz	Available	–40 to +105
	R5F5671EDGFP	PLQP0100KB-B	2 M	384 K	8 K	120 MHz	Not available	–40 to +105
	R5F5671CHGFP	PLQP0100KB-B	1.5 M	384 K	8 K	120 MHz	Available	–40 to +105
	R5F5671CDGFP	PLQP0100KB-B	1.5 M	384 K	8 K	120 MHz	Not available	–40 to +105
	R5F56719HGFP	PLQP0100KB-B	1 M	384 K	8 K	120 MHz	Available	–40 to +105
	R5F56719DGFP	PLQP0100KB-B	1 M	384 K	8 K	120 MHz	Not available	–40 to +105
	R5F5671EHGFM	PLQP0064KB-C	2 M	384 K	8 K	120 MHz	Available	–40 to +105
	R5F5671EDGFM	PLQP0064KB-C	2 M	384 K	8 K	120 MHz	Not available	–40 to +105
	R5F5671CHGFM	PLQP0064KB-C	1.5 M	384 K	8 K	120 MHz	Available	–40 to +105
	R5F5671CDGFM	PLQP0064KB-C	1.5 M	384 K	8 K	120 MHz	Not available	–40 to +105
	R5F56719HGFM	PLQP0064KB-C	1 M	384 K	8 K	120 MHz	Available	–40 to +105
	R5F56719DGFM	PLQP0064KB-C	1 M	384 K	8 K	120 MHz	Not available	–40 to +105
	R5F5671EHGNE	PWQN0048KC-A	2 M	384 K	8 K	120 MHz	Available	–40 to +105
	R5F5671EDGNE	PWQN0048KC-A	2 M	384 K	8 K	120 MHz	Not available	–40 to +105
	R5F5671CHGNE	PWQN0048KC-A	1.5 M	384 K	8 K	120 MHz	Available	–40 to +105
	R5F5671CDGNE	PWQN0048KC-A	1.5 M	384 K	8 K	120 MHz	Not available	–40 to +105
	R5F56719HGNE	PWQN0048KC-A	1 M	384 K	8 K	120 MHz	Available	–40 to +105
	R5F56719DGNE	PWQN0048KC-A	1 M	384 K	8 K	120 MHz	Not available	–40 to +105
	R5F5671EHGBP	PTBG0064KB-A	2 M	384 K	8 K	120 MHz	Available	–40 to +105
	R5F5671EDGBP	PTBG0064KB-A	2 M	384 K	8 K	120 MHz	Not available	–40 to +105
	R5F5671CHGBP	PTBG0064KB-A	1.5 M	384 K	8 K	120 MHz	Available	–40 to +105
	R5F5671CDGBP	PTBG0064KB-A	1.5 M	384 K	8 K	120 MHz	Not available	–40 to +105

Table 1.3 List of Products (3/3)

Group	Part No.	Package	Code Flash Memory Capacity (byte(s))	RAM Capacity (byte(s))	Data Flash Memory Capacity (byte(s))	Operating Frequency (Max.)	Encryption Module	Operating temperature (°C)
RX671 (G-version)	R5F56719HGBP	PTBG0064KB-A	1 M	384 K	8 K	120 MHz	Available	–40 to +105
	R5F56719DGBP	PTBG0064KB-A	1 M	384 K	8 K	120 MHz	Not available	–40 to +105
	R5F5671EHGLE	PTLG0145JC-A	2 M	384 K	8 K	120 MHz	Available	–40 to +105
	R5F5671EDGLE	PTLG0145JC-A	2 M	384 K	8 K	120 MHz	Not available	–40 to +105
	R5F5671CHGLE	PTLG0145JC-A	1.5 M	384 K	8 K	120 MHz	Available	–40 to +105
	R5F5671CDGLE	PTLG0145JC-A	1.5 M	384 K	8 K	120 MHz	Not available	–40 to +105
	R5F56719HGLE	PTLG0145JC-A	1 M	384 K	8 K	120 MHz	Available	–40 to +105
	R5F56719DGLE	PTLG0145JC-A	1 M	384 K	8 K	120 MHz	Not available	–40 to +105
	R5F5671EHGLK	PTLG0145KB-A	2 M	384 K	8 K	120 MHz	Available	–40 to +105
	R5F5671EDGLK	PTLG0145KB-A	2 M	384 K	8 K	120 MHz	Not available	–40 to +105
	R5F5671CHGLK	PTLG0145KB-A	1.5 M	384 K	8 K	120 MHz	Available	–40 to +105
	R5F5671CDGLK	PTLG0145KB-A	1.5 M	384 K	8 K	120 MHz	Not available	–40 to +105
	R5F56719HGLK	PTLG0145KB-A	1 M	384 K	8 K	120 MHz	Available	–40 to +105
	R5F56719DGLK	PTLG0145KB-A	1 M	384 K	8 K	120 MHz	Not available	–40 to +105
	R5F5671EHGLJ	PTLG0100JB-A	2 M	384 K	8 K	120 MHz	Available	–40 to +105
	R5F5671EDGLJ	PTLG0100JB-A	2 M	384 K	8 K	120 MHz	Not available	–40 to +105
	R5F5671CHGLJ	PTLG0100JB-A	1.5 M	384 K	8 K	120 MHz	Available	–40 to +105
	R5F5671CDGLJ	PTLG0100JB-A	1.5 M	384 K	8 K	120 MHz	Not available	–40 to +105
	R5F56719HGLJ	PTLG0100JB-A	1 M	384 K	8 K	120 MHz	Available	–40 to +105
	R5F56719DGLJ	PTLG0100JB-A	1 M	384 K	8 K	120 MHz	Not available	–40 to +105

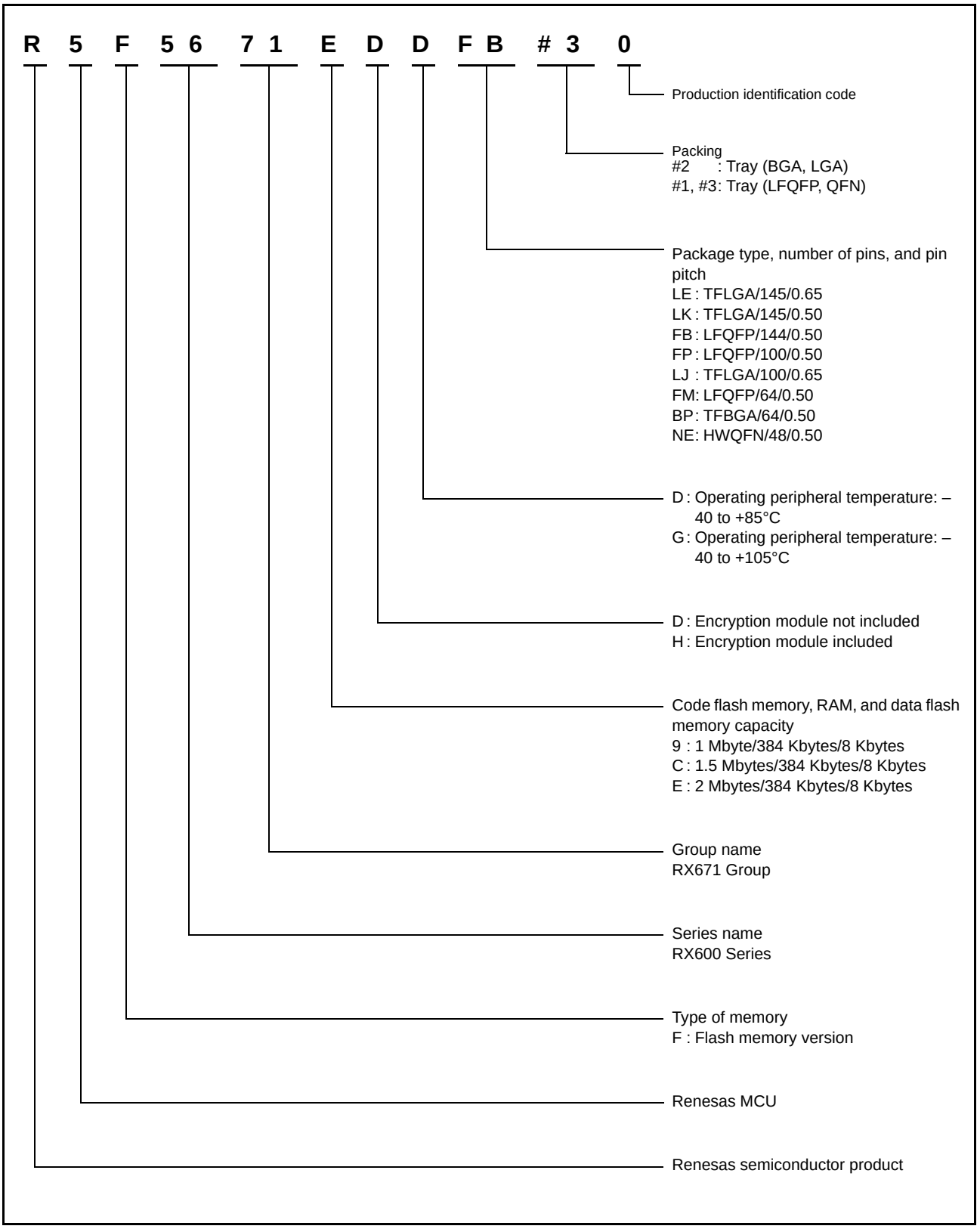
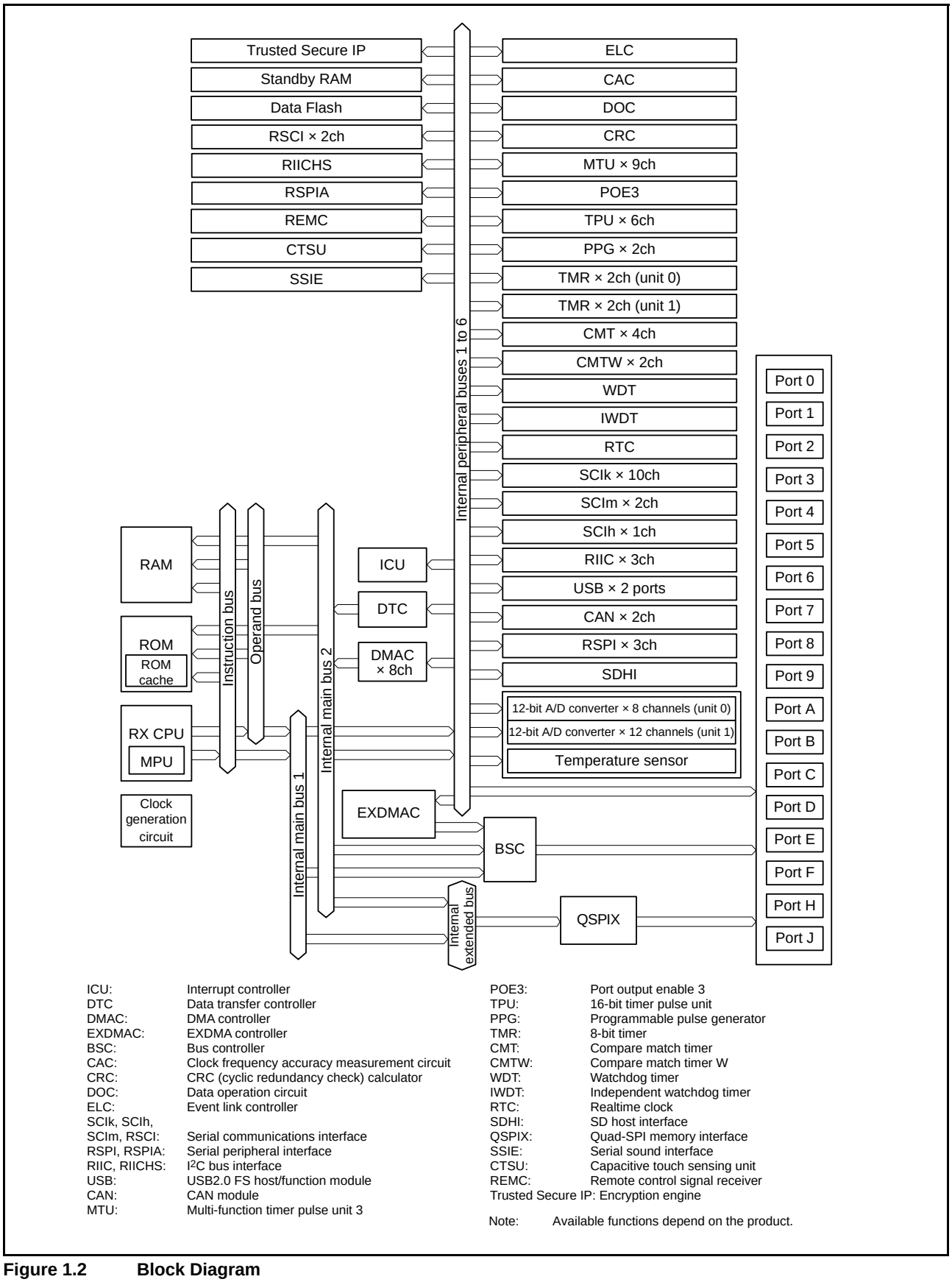


Figure 1.1 How to Read the Product Part Number

1.3 Block Diagram

Figure 1.2 shows a block diagram.



ICU: Interrupt controller
DTC Data transfer controller
DMAC: DMA controller
EXDMAC: EXDMA controller
BSC: Bus controller
CAC: Clock frequency accuracy measurement circuit
CRC: CRC (cyclic redundancy check) calculator
DOC: Data operation circuit
ELC: Event link controller
SCIk, SCIh, SCIm, RSCI: Serial communications interface
RSPI, RSPIA: Serial peripheral interface
RIIC, RIICHS: I^2C bus interface
USB: USB2.0 FS host/function module
CAN: CAN module
MTU: Multi-function timer pulse unit 3
POE3: Port output enable 3
TPU: 16-bit timer pulse unit
PPG: Programmable pulse generator
TMR: 8-bit timer
CMT: Compare match timer
CMTW: Compare match timer W
WDT: Watchdog timer
IWDT: Independent watchdog timer
RTC: Realtime clock
SDHI: SD host interface
QSPIX: Quad-SPI memory interface
SSIE: Serial sound interface
CTSU: Capacitive touch sensing unit
REMC: Remote control signal receiver
Trusted Secure IP: Encryption engine

Note: Available functions depend on the product.

Figure 1.2 Block Diagram

1.4 Pin Functions

Table 1.4 lists the pin functions.

Table 1.4 Pin Functions (1/8)

Classifications	Pin Name	I/O	Description
Digital power supply	VCC	Input	Power supply pin. Connect this pin to the system power supply. Connect the pin to VSS via a 0.1-µF multilayer ceramic capacitor. The capacitor should be placed close to the pin.
	VCL	Input	Connect this pin to VSS via a 0.22-µF multilayer ceramic capacitor. The capacitor should be placed close to the pin.
	VSS	Input	Ground pin. Connect it to the system power supply (0 V).
	VBATT	Input	Backup power pin
Clock	XTAL	Output	Pins for a crystal resonator. An external clock signal can be input through the EXTAL pin.
	EXTAL	Input	
	BCLK	Output	Outputs the external bus clock for external devices.
	SDCLK	Output	Outputs the SDRAM-dedicated clock.
	XCOUT	Output	Input/output pins for the sub clock oscillator. Connect a crystal resonator between XCOUT and XCIN.
	XCIN	Input	
	CLKOUT	Output	Clock output pin.
	EXCIN	Input	External clock input pin for the RTC, battery backup, and REMC
Clock frequency accuracy measurement	CACREF	Input	Reference clock input pin for the clock frequency accuracy measurement circuit
Operating mode control	MD	Input	Pin for setting the operating mode. The signal level on this pin must not be changed during operation.
	UB	Input	USB boot mode enable pin
	UPSEL	Input	Selects the power supply method in USB boot mode. The low level selects self-powered mode and the high level selects bus-powered mode.
System control	RES#	Input	Reset signal input pin. This LSI enters the reset state when this signal goes low.
	EMLE	Input	Input pin for the on-chip emulator enable signal. When the on-chip emulator is used, this pin should be driven high. When not used, it should be driven low.
	BSCANP	Input	Boundary scan enable pin. Boundary scan is enabled when this pin goes high. When not used, it should be driven low.
On-chip emulator	FINED	I/O	Fine interface pin
	TRST#	Input	On-chip emulator or boundary scan pins. When the EMLE pin is driven high, these pins are dedicated for the on-chip emulator.
	TMS	Input	
	TDI	Input	
	TCK	Input	
	TDO	Output	
	TRCLK	Output	This pin outputs the clock for synchronization with the trace data.
	TRSYNC TRSYNC1	Output	These pins indicate that output from the TRDATA0 to TRDATA7 pins is valid.
	TRDATA0 TRDATA1 TRDATA2 TRDATA3 TRDATA4 TRDATA5 TRDATA6 TRDATA7	Output	These pins output the trace information.

Table 1.4 Pin Functions (2/8)

Classifications	Pin Name	I/O	Description
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#, WR1#	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#, BC1#	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
	WAIT#	Input	Input pin for wait request signals in access to the external space
	CS0# to CS7#	Output	Select signals for CS areas
SDRAM interface	CKE	Output	SDRAM clock enable signal
	SDCS#	Output	SDRAM chip select signal
	RAS#	Output	SDRAM row address strobe signal
	CAS#	Output	SDRAM column address strove signal
	WE#	Output	SDRAM write enable pin
	DQM0, DQM1	Output	SDRAM I/O data mask enable signals
EXDMA controller	EDREQ0, EDREQ1	Input	External DMA transfer request pins
	EDACK0, EDACK1	Output	Single address transfer acknowledge signals
Interrupt	NMI	Input	Non-maskable interrupt request pin
	IRQ0 to IRQ15, IRQ0-DS to IRQ15-DS	Input	Maskable interrupt request pins
Multi-function timer pulse unit 3	MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D	I/O	The TGRA0 to TGRD0 input capture input/output compare output/PWM output pins
	MTIOC1A, MTIOC1B	I/O	The TGRA1 and TGRB1 input capture input/output compare output/PWM output pins
	MTIOC2A, MTIOC2B	I/O	The TGRA2 and TGRB2 input capture input/output compare output/PWM output pins
	MTIOC3A, MTIOC3B, MTIOC3C, MTIOC3D	I/O	The TGRA3 to TGRD3 input capture input/output compare output/PWM output pins
	MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D	I/O	The TGRA4 to TGRD4 input capture input/output compare output/PWM output pins
	MTIC5U, MTIC5V, MTIC5W	Input	The TGRU5, TGRV5, and TGRW5 input capture input/dead time compensation input pins
	MTIOC6A, MTIOC6B, MTIOC6C, MTIOC6D	I/O	The TGRA6 to TGRD6 input capture input/output compare output/PWM output pins
	MTIOC7A, MTIOC7B, MTIOC7C, MTIOC7D	I/O	The TGRA7 to TGRD7 input capture input/output compare output/PWM output pins
	MTIOC8A, MTIOC8B, MTIOC8C, MTIOC8D	I/O	The TGRA8 to TGRD8 input capture input/output compare output/PWM output pins
	MTCLKA, MTCLKB, MTCLKC, MTCLKD	Input	Input pins for external clock signals or for phase counting mode clock signals
Port output enable 3	POE0#, POE4#, POE8#, POE10#, POE11#	Input	Input pins for request signals to place the MTU in the high impedance state

Table 1.4 Pin Functions (3/8)

Classifications	Pin Name	I/O	Description
16-bit timer pulse unit	TIOCA0, TIOCB0, TIOCC0, TIOCD0	I/O	The TGRA0 to TGRD0 input capture input/output compare output/PWM output pins
	TIOCA1, TIOCB1	I/O	The TGRA1 and TGRB1 input capture input/output compare output/PWM output pins
	TIOCA2, TIOCB2	I/O	The TGRA2 and TGRB2 input capture input/output compare output/PWM output pins
	TIOCA3, TIOCB3, TIOCC3, TIOCD3	I/O	The TGRA3 to TGRD3 input capture input/output compare output/PWM output pins
	TIOCA4, TIOCB4	I/O	The TGRA4 and TGRB4 input capture input/output compare output/PWM output pins
	TIOCA5, TIOCB5	I/O	The TGRA5 and TGRB5 input capture input/output compare output/PWM output pins
	TCLKA, TCLKB, TCLKC, TCLKD	Input	Input pins for external clock signals or for phase counting mode clock signals
Programmable pulse generator	PO0 to PO31	Output	Output pins for the pulse signals
8-bit timer	TMO0 to TMO3	Output	Compare match output pins
	TMCI0 to TMCI3	Input	Input pins for external clocks to be input to the counter
	TMRI0 to TMRI3	Input	Input pins for the counter reset
Compare match timer W	TIC0 to TIC3	Input	Input pins for CMTW
	TOC0 to TOC3	Output	Output pins for CMTW
Serial communications interface (SCIk)	• Asynchronous mode/clock synchronous mode		
	SCK0 to SCK9	I/O	Input/output pins for the clock
	RXD0 to RXD9	Input	Input pins for received data
	TXD0 to TXD9	Output	Output pins for transmitted data
	CTS0# to CTS9#	Input	Input pins for controlling the start of transmission and reception
	RTS0# to RTS9#	Output	Output pins for controlling the start of transmission and reception
	• Simple I^2C mode		
	SSCL0 to SSCL9	I/O	Input/output pins for the I^2C clock
	SSDA0 to SSDA9	I/O	Input/output pins for the I^2C data
	• Simple SPI mode		
	SCK0 to SCK9	I/O	Input/output pins for the clock
	SMISO0 to SMISO9	I/O	Input/output pins for slave transmission of data
	SMOSI0 to SMOSI9	I/O	Input/output pins for master transmission of data
	SS0# to SS9#	Input	Chip-select input pins

Table 1.4 Pin Functions (4/8)

Classifications	Pin Name	I/O	Description
Serial communications interface (SCIh)	• Asynchronous mode/clock synchronous mode		
	SCK12	I/O	Input/output pin for the clock
	RXD12	Input	Input pin for received data
	TXD12	Output	Output pin for transmitted data
	CTS12#	Input	Input pin for controlling the start of transmission and reception
	RTS12#	Output	Output pin for controlling the start of transmission and reception
	• Simple I^2C mode		
	SSCL12	I/O	Input/output pin for the I^2C clock
	SSDA12	I/O	Input/output pin for the I^2C data
	• Simple SPI mode		
	SCK12	I/O	Input/output pin for the clock
	SMISO12	I/O	Input/output pin for slave transmission of data
	SMOSI12	I/O	Input/output pin for master transmission of data
	SS12#	Input	Chip-select input pin
	• Extended serial mode		
	RXDX12	Input	Input pin for received data
	TXDX12	Output	Output pin for transmitted data
	SIOX12	I/O	Input/output pin for received or transmitted data
Serial communications interface (SCIm)	• Asynchronous mode/clock synchronous mode		
	SCK10, SCK11	I/O	Input/output pins for the clock
	RXD10, RXD11	Input	Input pins for received data
	TXD10, TXD11	Output	Output pins for transmitted data
	CTS10#, CTS11#	Input	Input pins for controlling the start of transmission and reception
	RTS10#, RTS11#	Output	Output pins for controlling the start of transmission and reception
	• Simple I^2C mode		
	SSCL10, SSCL11	I/O	Input/output pins for the I^2C clock
	SSDA10, SSDA11	I/O	Input/output pins for the I^2C data
	• Simple SPI mode		
	SCK10, SCK11	I/O	Input/output pins for the clock
	SMISO10, SMISO11	I/O	Input/output pins for slave transmission of data
	SMOSI10, SMOSI11	I/O	Input/output pins for master transmission of data
	SS10#, SS11#	Input	Chip-select input pins

Table 1.4 Pin Functions (5/8)

Classifications	Pin Name	I/O	Description
Serial communications interface (RSCI)	• Asynchronous mode/clock synchronous mode		
	SCK010, SCK011	I/O	Input/output pins for the clock
	RXD010, RXD011	Input	Input pins for received data
	TXD010, TXD011	Output	Output pins for transmitted data
	CTS010#, CTS011#	Input	Input pins for controlling the start of transmission and reception
	RTS010#, RTS011#	Output	Output pins for controlling the start of transmission and reception
	DE010, DE011	Output	DriveEnable output pins
	• Simple I^2C mode		
	SSCL010, SSCL011	I/O	Input/output pins for the I^2C clock
	SSDA010, SSDA011	I/O	Input/output pins for the I^2C data
	• Simple SPI mode		
	SCK010, SCK011	I/O	Input/output pins for the clock
	SMISO010, SMISO011	I/O	Input/output pins for slave transmission of data
	SMOSI010, SMOSI011	I/O	Input/output pins for master transmission of data
	SS010#, SS011#	Input	Chip-select input pins
	• HBS support mode		
	RXD010, RXD011	Input	Input pin for received data
	TXD010, TXD011, TXDA011, TXDB011	Output	Output pins for transmitted data
I^2C bus interface	SCL0[FM+], SCL1, SCL2, SCL2-DS	I/O	Input/output pins for clocks. Bus can be directly driven by the N-channel open drain
	SDA0[FM+], SDA1, SDA2, SDA2-DS	I/O	Input/output pins for data. Bus can be directly driven by the N-channel open drain
Hi-speed I^2C bus interface (RIICHS)	SCLHS0[FM+/HS]	I/O	Input/output pin for clocks. Bus can be directly driven by the N-channel open drain
	SDAHS0[FM+/HS]	I/O	Input/output pin for data. Bus can be directly driven by the N-channel open drain
USB 2.0 host/function module	VCC_USB	Input	Power supply pin
	VSS_USB	Input	Ground pin
	USB0_DP, USB1_DP	I/O	Input or output USB transceiver D+ data.
	USB0_DM, USB1_DM	I/O	Input or output USB transceiver D- data.
	USB0_EXICEN, USB1_EXICEN	Output	Connect to the OTG power IC.
	USB0_ID, USB1_ID	Input	Connect to the OTG power IC.
	USB0_VBUSEN, USB1_VBUSEN	Output	USB VBUS power enable pins
	USB0_OVRCURA/ USB0_OVRCURB, USB1_OVRCURA/ USB1_OVRCURB	Input	USB overcurrent pins
	USB0_VBUS, USB1_VBUS	Input	USB cable connection/disconnection detection input pins
CAN module	CRX0, CRX1, CRX1-DS	Input	Input pins
	CTX0, CTX1	Output	Output pins

Table 1.4 Pin Functions (6/8)

Classifications	Pin Name	I/O	Description
Serial peripheral interface	RSPCKA-A/RSPCKA-B/ RSPCKB-A/RSPCKB-B/ RSPCKC-A/RSPCKC-B	I/O	Clock input/output pins
	MOSIA-A/MOSIA-B/ MOSIB-A/MOSIB-B/ MOSIC-A/MOSIC-B	I/O	Input or output data output from the master
	MISOA-A/MISOA-B/ MISOB-A/MISOB-B/ MISOC-A/MISOC-B	I/O	Input or output data output from the slave
	SSLA0-A/SSLA0-B/ SSLB0-A/SSLB0-B/ SSLC0-A	I/O	Input or output pins for slave selection
	SSLA1-A/SSLA1-B/ SSLB1-A/SSLB1-B/ SSLC1-A, SSLA2-A/SSLA2-B/ SSLB2-A/SSLB2-B/ SSLC2-A, SSLA3-A/SSLA3-B/ SSLB3-A/SSLB3-B/ SSLC3-A	Output	Output pins for slave selection
Serial peripheral interface (RSPIA)	RSPCK0-A/RSPCK0-B	I/O	Clock input/output pins
	MOSI0-A/MOSI0-B	I/O	Input or output data output from the master
	MISO0-A/MISO0-B	I/O	Input or output data output from the slave
	SSL00-A/SSL00-B	I/O	Input or output pins for slave selection
	SSL01-A/SSL01-B, SSL02-A/SSL02-B, SSL03-A/SSL03-B	Output	Output pins for slave selection
Quad-SPI memory interface	QSPCLK-A/QSPCLK-B	Output	Clock output pins
	QSSL-A/QSSL-B	Output	Output pins for slave selection
	QIO0-A/QIO0-B, QIO1-A/QIO1-B, QIO2-A/QIO2-B, QIO3-A/QIO3-B	I/O	Data input/output pins
Serial sound interface	SSIBCK0	I/O	SSIE serial bit-clock pin
	SSILRCK0	I/O	LR clock
	SSITXD0	Output	Serial data output pin
	SSIRXD0	Input	Serial data input pin
	AUDIO_CLK	Input	External clock pin for audio (input for an oversampling clock)
SD host interface	SDHI_CLK-A/SDHI_CLK-B/ SDHI_CLK-C	Output	SD clock output pins
	SDHI_CMD-A/SDHI_CMD-B/ SDHI_CMD-C	I/O	SD command output, response input signal pins
	SDHI_D3-A/SDHI_D3-B/ SDHI_D3-C to SDHI_D0-A/ SDHI_D0-B/SDHI_D0-C	I/O	SD data bus pins
	SDHI_CD	Input	SD card detection pin
	SDHI_WP	Input	SD write-protect signal
Realtime clock	RTCOUT	Output	Output pin for 1-Hz/64-Hz clock
	RTCIC0 to RTCIC2	Input	Time capture event input pins
Battery backup	TAMPI0 to TAMPI2	Input	Input pins for detecting tampering
Remote control signal receiver (REMC)	PMC0-DS	Input	Input pin for external pulse signal

Table 1.4 Pin Functions (7/8)

Classifications	Pin Name	I/O	Description
12-bit A/D converter	AN000 to AN007, AN100 to AN111	Input	Input pins for the analog signals to be processed by the A/D converter
	ADTRG0#, ADTRG1#	Input	Input pins for the external trigger signals that start the A/D conversion
	ANEX0	Output	Extended analog output pin
	ANEX1	Input	Extended analog input pin
Capacitive touch sensing unit (CTSU)	TS0 to TS16	I/O	Electrostatic capacitance measurement pins (touch pins).
	TSCAP	—	Connect to the VSS via a decoupling capacitor (10 nF) for stabilizing the internal voltage. The capacitor should be placed close to the pin.
Analog power supply	AVCC0*1	Input	Analog voltage supply pin for the 12-bit A/D converter (unit 0). Connect this pin to a branch from the VCC power supply. Connect the pin to AVSS0 via a 0.1-µF multilayer ceramic capacitor. The capacitor should be placed close to the pin.
	AVSS0*1	Input	Analog ground pin for the 12-bit A/D converter (unit 0). Connect this pin to a branch from the VSS ground power supply. Connect the pin to AVCC0 via a 0.1-µF multilayer ceramic capacitor. The capacitor should be placed close to the pin.
	VREFH0	Input	Analog reference voltage supply pin for the 12-bit A/D converter (unit 0). Connect this pin to VCC if the 12-bit A/D converter is not to be used.
	VREFL0	Input	Analog reference ground pin for the 12-bit A/D converter (unit 0). Connect this pin to VSS if the 12-bit A/D converter is not to be used.
	AVCC1*1	Input	Analog voltage supply and reference voltage supply pin for the 12-bit A/D converter (unit 1). This pin also supplies the analog voltage to the temperature sensor. Connect this pin to a branch from the VCC power supply. Connect the pin to AVSS1 via a 0.1-µF multilayer ceramic capacitor. The capacitor should be placed close to the pin.
	AVSS1*1	Input	Analog voltage supply and reference voltage supply pin for the 12-bit A/D converter (unit 1). This pin also supplies the analog ground voltage to the temperature sensor. Connect this pin to a branch from the VSS ground power supply. Connect the pin to AVCC1 via a 0.1-µF multilayer ceramic capacitor. The capacitor should be placed close to the pin.

Table 1.4 Pin Functions (8/8)

Classifications	Pin Name	I/O	Description
I/O ports	P00 to P03, P05, P07	I/O	6-bit input/output pins
	P12 to P17	I/O	6-bit input/output pins
	P20 to P27	I/O	8-bit input/output pins
	P30 to P37	I/O	8-bit input/output pins (P35: input pin)
	P40 to P47	I/O	8-bit input/output pins
	P50 to P56	I/O	7-bit input/output pins
	P60 to P67	I/O	8-bit input/output pins
	P70 to P77	I/O	8-bit input/output pins
	P80 to P83, P86, P87	I/O	6-bit input/output pins
	P90 to P93	I/O	4-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
	PF5	I/O	1-bit input/output pins
	PH1, PH2	I/O	2-bit input/output pins
	PJ3, PJ5	I/O	2-bit input/output pins

Note: Note the following regarding pin names. For details, see section 1.6, List of Pin and Pin Functions.
- When a letter “-A”, “-B”, etc. to indicate group membership is appended to the pin name, each pin is recommended to use in combination with the pins in the same group.
  All RSPI, RSPIA, QSPIX, SDHI AC timings are measured in combination with the pins in the same group.
- When the pin functions have “-DS” appended to their names, they can also be used as triggers for release from deep software standby.
- RIIC and RIICHS pin functions that have [FM+] appended to their names support fast-mode plus.

Note 1. When neither the 12-bit A/D converter nor temperature sensor is to be used, connect the AVCC0 and AVCC1 pins to VCC, and the AVSS0 and AVSS1 pins to VSS.

1.5 Pin Assignments

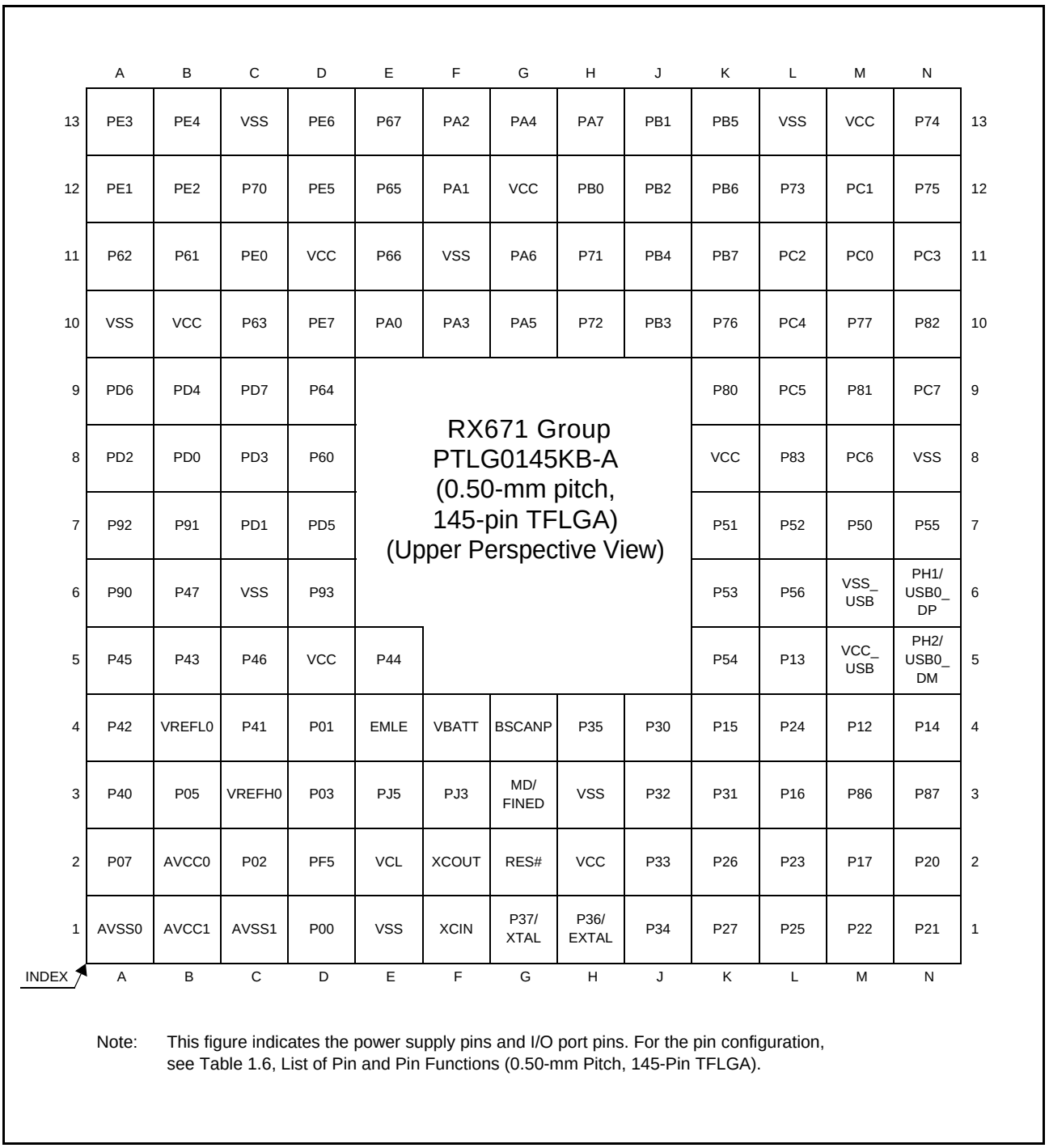
1.5.1 145-Pin TFLGA (0.65-mm Pitch)

	A	B	C	D	E	F	G	H	J	K	L	M	N	
13	PE3	PE4	VSS	PE6	P67	PA2	USB1_ DP	USB1_ DM	PB1	PB5	VSS	VCC	P74	13
12	PE1	PE2	P70	PE5	P65	PA1	VSS_ USB	VCC_ USB	PB2	PB6	P73	PC1	P75	12
11	P62	P61	PE0	VCC	P66	PA4	PA6	PA7	PB4	PB7	PC2	PC0	PC3	11
10	VSS	VCC	P63	PE7	PA0	PA3	PA5	PB0	PB3	P76	PC4	P77	P82	10
9	PD6	PD4	PD7	P64	RX671 Group PTLG0145JC-A (0.65-mm pitch, 145-pin TFLGA) (Upper Perspective View)					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	PH1/ USB0_ DP	6
5	P45	P43	P46	VCC	P44					P54	P13	VCC_ USB	PH2/ 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	AVCC1	AVSS1	P00	VSS	XCIN	P37/ XTAL	P36/ EXTAL	P34	P27	P25	P22	P21	1
INDEX	A	B	C	D	E	F	G	H	J	K	L	M	N	

Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.5, List of Pin and Pin Functions (0.65-mm Pitch, 145-Pin TFLGA).

Figure 1.3 Pin Assignment (0.65-mm Pitch, 145-Pin TFLGA)

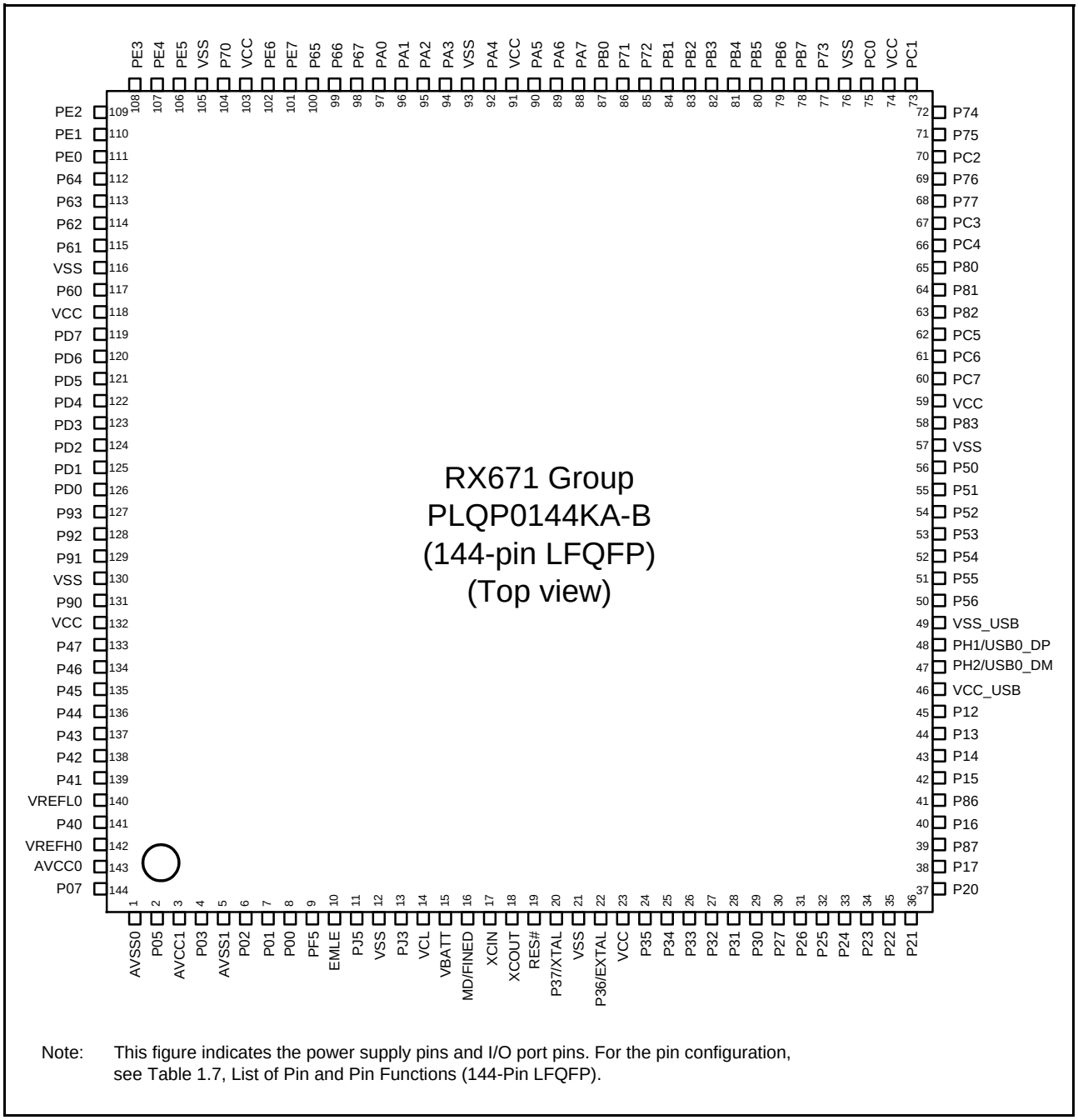
1.5.2 145-Pin TFLGA (0.50-mm Pitch)



Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.6, List of Pin and Pin Functions (0.50-mm Pitch, 145-Pin TFLGA).

Figure 1.4 Pin Assignment (0.50-mm Pitch, 145-Pin TFLGA)

1.5.3 144-Pin LFQFP



Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.7, List of Pin and Pin Functions (144-Pin LFQFP).

Figure 1.5 Pin Assignment (144-Pin LFQFP)

1.5.4 100-Pin TFLGA

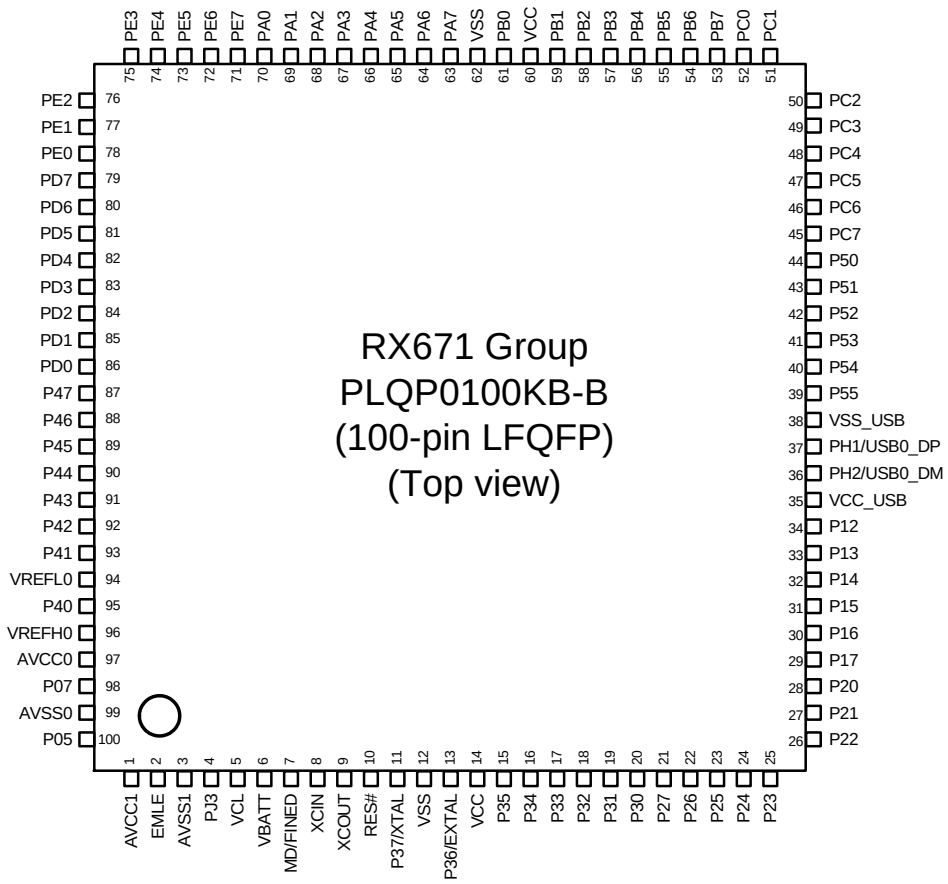
RX671 Group
PTLG0100JB-A (100-pin TFLGA)
(Upper Perspective View)

	A	B	C	D	E	F	G	H	J	K	
10	PE2	PE3	PE4	PA0	PA3	VSS	VCC	PB7	PC1	PC2	10
9	PE1	PD7	PE5	PA1	PA5	PA7	PB1	PB6	PC0	PC3	9
8	PE0	PD6	PD5	PE7	PA4	PB0	PB4	PC6	PC4	PC5	8
7	PD4	PD3	PD2	PE6	PA6	PB2	PB5	PC7	P50	P51	7
6	PD0	PD1	P47	P46	PA2	PB3	P52	P54	VCC_ USB	PH1/ USB0_ DP	6
5	P43	P44	P42	P45	P41	P12	P53	P55	VSS_ USB	PH2/ USB0_ DM	5
4	VREFL0	P40	VREFH0	VBATT	P34	P32	P27	P15	P13	P14	4
3	P07	AVCC0	PJ3	MD/ FINED	RES#	P35	P30	P16	P17	P20	3
2	AVCC1	AVSS0	AVSS1	XCOUT	VSS	VCC	P31	P25	P21	P22	2
1	P05	EMLE	VCL	XCIN	P37/ XTAL	P36/ EXTAL	P33	P26	P24	P23	1
INDEX	A	B	C	D	E	F	G	H	J	K	

Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.8, List of Pin and Pin Functions (100-Pin TFLGA).

Figure 1.6 Pin Assignment (100-Pin TFLGA)

1.5.5 100-Pin LFQFP



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

Figure 1.7 Pin Assignment (100-Pin LFQFP)

1.5.6 64-Pin TFBGA

RX671 Group
PTBG0064KB-A (64-pin TFBGA)
(Upper Perspective View)

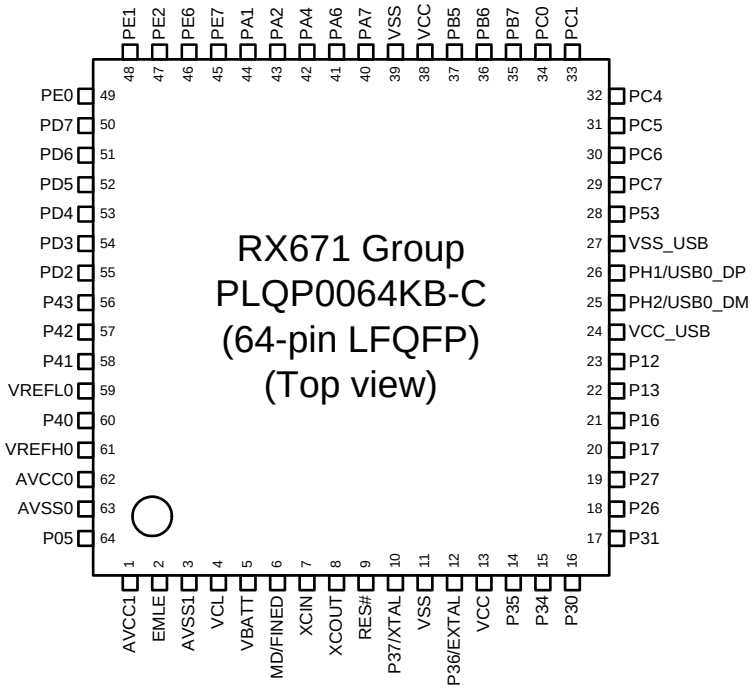
	A	B	C	D	E	F	G	H	
8	PE2	PE6	PE7	PA4	VSS	PB5	PC0	PC1	8
7	PE0	PE1	PA1	PA2	VCC	PB6	PC5	PC4	7
6	PD7	PD6	PD5	PA6	PA7	PB7	PC7	PC6	6
5	PD2	PD3	PD4	P43	BSCANP	P53	VSS_USB	PH1/ USB0_DP	5
4	VREFL0	P42	P41	P40	P13	P12	VCC_USB	PH2/ USB0_DM	4
3	VREFH0	AVCC0	MD/FINED	RES#	P34	P35	P30	P16	3
2	AVSS0	AVSS1	VBATT	XCOUT	VSS	VCC	P31	P17	2
1	AVCC1	EMLE	VCL	XCIN	P37/ XTAL	P36/ EXTAL	P27	P26	1
	A	B	C	D	E	F	G	H	

INDEX

Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.10, List of Pin and Pin Functions (64-Pin TFBGA).

Figure 1.8 Pin Assignment (64-Pin TFBGA)

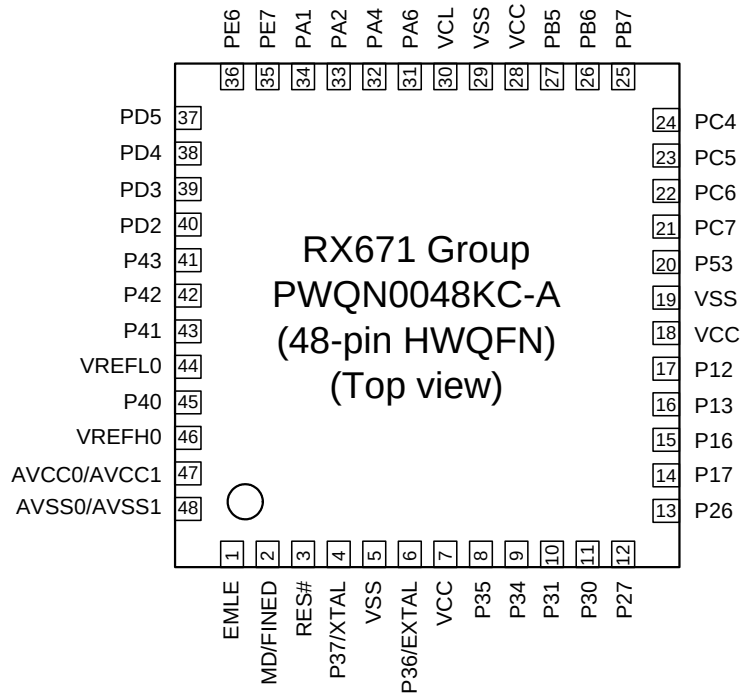
1.5.7 64-Pin LFQFP



Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.11, List of Pin and Pin Functions (64-Pin LFQFP).

Figure 1.9 Pin Assignment (64-Pin LFQFP)

1.5.8 48-Pin HWQFN



Note: It is recommended to connect the exposed die pad to VSS.

Note: This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.12, List of Pin and Pin Functions (48-Pin HWQFN).

Figure 1.10 Pin Assignment (48-Pin HWQFN)

1.6 List of Pin and Pin Functions

1.6.1 145-Pin TFLGA (0.65-mm Pitch)

Table 1.5 List of Pin and Pin Functions (0.65-mm Pitch, 145-Pin TFLGA) (1/8)

Pin No.				Timer	Communication				
145-Pin TFLGA (0.65-mm pitch)	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
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		IRQ0	AN108	
A7		P92	A18	POE4#	RXD7/SMISO7/ SSCL7		IRQ10		
A8		PD2	D2[A2/D2]	MTIOC4D/ TIC2	CRX0/MISOC-A	SDHI_D2-B/ QIO2-B	IRQ2	AN105	
A9		PD6	D6[A6/D6]	MTIC5V/ MTIOC8A/ POE4#	SSLC2-A	SDHI_D0-B/ QIO0-B	IRQ6	AN101	
A10	VSS								
A11		P62	CS2#/ RAS#/ D1[A1/D1]				IRQ2		
A12		PE1	D9[A9/D9]/ D1[A1/D1]	MTIOC4C/ MTIOC3B/ PO18	TXD12/SMOSI12/ SSDA12/ TXDX12/SIOX12/ SSLB2-B		IRQ9	ANEX1	
A13		PE3	D11[A11/ D11]/ D3[A3/D3]	MTIOC4B/ PO26/POE8#/ TOC3	CTS12#/RTS12#/ SS12#		IRQ11		
B1	AVCC1								
B2	AVCC0								
B3		P05					IRQ13		
B4	VREFL0								
B5		P43					IRQ11-DS	AN003	
B6		P47					IRQ15-DS	AN007	
B7		P91	A17		SCK7		IRQ9		
B8		PD0	D0[A0/D0]	POE4#			IRQ0	AN107	
B9		PD4	D4[A4/D4]	MTIOC8B/ POE11#	SSLC0-A	SDHI_CMD-B/ QSSL-B	IRQ4	AN103	
B10	VCC								
B11		P61	CS1#/ SDCS#/ D0[A0/D0]				IRQ1		
B12		PE2	D10[A10/ D10]/ D2[A2/D2]	MTIOC4A/ PO23/TIC3	RXD12/ SMISO12/ SSCL12/ RXDX12/ SSLB3-B		IRQ7-DS		

Table 1.5 List of Pin and Pin Functions (0.65-mm Pitch, 145-Pin TFLGA) (2/8)

Pin No.				Timer	Communication				
145-Pin TFLGA (0.65-mm pitch)	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
B13		PE4	D12[A12/D12]/D4[A4/D4]	MTIOC4D/MTIOC1A/PO28	SSLB0-B		IRQ12		
C1	AVSS1								
C2		P02		TMCI1	SCK6		IRQ10	AN109	
C3	VREFH0								
C4		P41					IRQ9-DS	AN001	
C5		P46					IRQ14-DS	AN006	
C6	VSS								
C7		PD1	D1[A1/D1]	MTIOC4B/POE0#	CTX0/MOSIC-A		IRQ1	AN106	
C8		PD3	D3[A3/D3]	MTIOC8D/POE8#/TOC2	RSPCKC-A	SDHI_D3-B/QIO3-B	IRQ3	AN104	
C9		PD7	D7[A7/D7]	MTIC5U/POE0#	SSLC3-A	SDHI_D1-B/QIO1-B	IRQ7	AN100	
C10		P63	CS3#/CAS#/D2[A2/D2]				IRQ3		
C11		PE0	D8[A8/D8]/D0[A0/D0]	MTIOC3D	SCK12/SSLB1-B		IRQ8	ANEX0	
C12		P70	SDCLK				IRQ0		
C13	VSS								
D1		P00		TMRI0	TXD6/SMOSI6/SSDA6		IRQ8	AN111	
D2		PF5					IRQ4		
D3		P03					IRQ11		
D4		P01		TMCI0	RXD6/SMISO6/SSCL6		IRQ9	AN110	
D5	VCC								
D6		P93	A19	POE0#	CTS7#/RTS7#/SS7#		IRQ11		
D7		PD5	D5[A5/D5]	MTIC5W/MTIOC8C/POE10#	SSLC1-A	SDHI_CLK-B/QSPCLK-B	IRQ5	AN102	
D8		P60	CS0#				IRQ0		
D9		P64	CS4#/WE#/D3[A3/D3]				IRQ4		
D10		PE7	D15[A15/D15]/D7[A7/D7]	MTIOC6A/TOC1	MISOB-B	SDHI_WP/SDHI_D1-B/QIO1-B	IRQ7		
D11	VCC								
D12		PE5	D13[A13/D13]/D5[A5/D5]	MTIOC4C/MTIOC2B	RSPCKB-B		IRQ5		
D13		PE6	D14[A14/D14]/D6[A6/D6]	MTIOC6C/TIC1	MOSIB-B	SDHI_CD/SDHI_D0-B/QIO0-B	IRQ6		
E1	VSS								
E2	VCL								

Table 1.5 List of Pin and Pin Functions (0.65-mm Pitch, 145-Pin TFLGA) (3/8)

Pin No.				Timer	Communication				
145-Pin TFLGA (0.65-mm pitch)	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
E3		PJ5		POE8#	CTS2#/RTS2#/SS2#		IRQ13		
E4	EMLE								
E5		P44					IRQ12-DS	AN004	
E10		PA0	A0/BC0#	MTIOC4A/MTIOC6D/TIOCA0/CACREF/PO16	SSLA1-B/SSL01-B		IRQ0		
E11		P66	CS6#/DQM0	MTIOC7D			IRQ14		
E12		P65	CS5#/CKE				IRQ13		
E13		P67	CS7#/DQM1	MTIOC7C			IRQ15		
F1	XCIN								
F2	XCOUT								
F3	EXCIN	PJ3	EDACK1	MTIOC3C	CTS6#/RTS6#/CTS0#/RTS0#/SS6#/SS0#		IRQ11		
F4	VBATT								
F10		PA3	A3	MTIOC0D/MTCLKD/TIOCD0/TCLKB/PO19	RXD5/SMISO5/SSCL5		IRQ6-DS		
F11		PA4	A4	MTIC5U/MTCLKA/TIOCA1/TMRI0/PO20	TXD5/SMOSI5/SSDA5/SSLA0-B/SSL00-B/TXD12/SMOSI12/SSDA12/TXDX12/SIOX12		IRQ5-DS		
F12		PA1	A1	MTIOC0B/MTCLKC/MTIOC7B/TIOCB0/PO17	SCK5/SSLA2-B/SSL02-B/SCK12	SDHI_CD	IRQ11		
F13		PA2	A2	MTIOC7A/PO18	RXD5/SMISO5/SSCL5/SSLA3-B/SSL03-B/RXD12/SMISO12/SSCL12/RXDX12	SDHI_WP	IRQ10		
G1	XTAL	P37							
G2	RES#								
G3	MD/FINED								
G4	BSCANP								
G10		PA5	A5	MTIOC6B/TIOCB1/PO21	RSPCKA-B/RSPCK0-B		IRQ5		
G11		PA6	A6	MTIC5V/MTCLKB/TIOCA2/TMCI3/PO22/POE10#	CTS5#/RTS5#/SS5#/MOSIA-B/MOSI0-B/CTS12#/RTS12#/SS12#		IRQ14		
G12	VSS_USB								
G13					USB1_DP				

Table 1.5 List of Pin and Pin Functions (0.65-mm Pitch, 145-Pin TFLGA) (4/8)

Pin No.				Timer	Communication				
145-Pin TFLGA (0.65-mm pitch)	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
H1	EXTAL	P36							
H2	VCC								
H3	VSS								
H4	UPSEL	P35					NMI		
H10		PB0	A8	MTIC5W/ TIOCA3/PO24	RXD4/RXD6/ SMISO4/ SMISO6/SSCL4/ SSCL6		IRQ12		
H11		PA7	A7	TIOCB2/PO23	MISOA-B/ MISO0-B		IRQ7		
H12	VCC_USB								
H13					USB1_DM				
J1	TRST#	P34		MTIOC0A/ TMCI3/PO12/ POE10#	SCK6/SCK0		IRQ4		TS0
J2		P33	EDREQ1	MTIOC0D/ TIOCD0/ TMRI3/PO11/ POE4#/ POE11#	RXD6/RXD0/ SMISO6/ SMISO0/SSCL6/ SSCL0/CRX0		IRQ3-DS		TS1
J3		P32		MTIOC0C/ TIOCC0/ TMO3/PO10/ RTCOUT/ RTCIC2/ POE0#/ POE10#	TXD6/TXD0/ SMOSI6/ SMOSI0/SSDA6/ SSDA0/CTX0/ USB0_VBUSEN		IRQ2-DS		TAMPI2
J4	TDI	P30		MTIOC4B/ TMRI3/PO8/ RTCIC0/ POE8#	RXD1/SMISO1/ SSCL1/MISOB-A		IRQ0-DS		TAMPI0
J10		PB3	A11	MTIOC0A/ MTIOC4A/ TIOCD3/ TCLKD/ TMO0/PO27/ POE11#	SCK4/SCK6/ PMC0-DS		IRQ3		
J11		PB4	A12	TIOCA4/PO28	CTS9#/RTS9#/ SS9#/SS11#/ CTS11#/RTS11#/ SS011#/ CTS011#/ RTS011#/DE011		IRQ4		
J12		PB2	A10	TIOCC3/ TCLKC/PO26	CTS4#/RTS4#/ CTS6#/RTS6#/ SS4#/SS6#		IRQ2		
J13		PB1	A9	MTIOC0C/ MTIOC4C/ TIOCB3/ TMCI0/PO25	TXD4/TXD6/ SMOSI4/ SMOSI6/SSDA4/ SSDA6		IRQ4-DS		
K1	TCK	P27	CS7#	MTIOC2B/ TMCI3/PO7	SCK1/RSPCKB-A		IRQ7		TS2

Table 1.5 List of Pin and Pin Functions (0.65-mm Pitch, 145-Pin TFLGA) (5/8)

Pin No.				Timer	Communication				
145-Pin TFLGA (0.65-mm pitch)	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
K2	TDO	P26	CS6#	MTIOC2A/ TMO1/PO6	TXD1/CTS3#/ RTS3#/SMOSI1/ SS3#/SSDA1/ MOSIB-A		IRQ6		TS3
K3	TMS	P31		MTIOC4D/ TMCI2/PO9/ RTCIC1	CTS1#/RTS1#/ SS1#/SSLB0-A		IRQ1-DS		TAMPI1
K4		P15		MTIOC0B/ MTCLKB/ TIOCB2/ TCLKB/ TMCI2/PO13	RXD1/SCK3/ SMISO1/SSCL1/ CRX1-DS		IRQ5		TS10
K5	TRDATA2	P54	ALE/ D1[A1/D1]/ EDACK0	MTIOC4B/ TMCI1	CTS2#/RTS2#/ SS2#/CTX1/ MOSIC-B		IRQ4		
K6		P53*1	BCLK		SSIRXD0/ PMC0-DS		IRQ3		TS12
K7		P51	WR1#/ BC1#/ WAIT#		SCK2/SSLB2-A		IRQ1		
K8	VCC								
K9	TRDATA0	P80	EDREQ0	MTIOC3B/ PO26	SCK10/RTS10#/ SCK010/ RTS010#/DE010/ USB1_EXICEN	SDHI_WP/ QIO2-A	IRQ8		
K10	TRDATA6	P76	CS6#	PO22	SMISO11/ SSCL11/RXD11/ SMISO011/ SSCL011/ RXD011	SDHI_CMD-A/ QSSL-A	IRQ14		
K11		PB7	A15	MTIOC3B/ TIOCB5/PO31	TXD9/SMOSI9/ SSDA9/ SMOSI11/ SSDA11/TXD11/ SMOSI011/ SSDA011/ TXD011		IRQ15		
K12		PB6	A14	MTIOC3D/ TIOCA5/PO30	RXD9/SMISO9/ SSCL9/SMISO11/ SSCL11/RXD11/ SMISO011/ SSCL011/ RXD011		IRQ6		
K13		PB5	A13	MTIOC2A/ MTIOC1B/ TIOCB4/ TMRI1/PO29/ POE4#	SCK9/SCK11/ SCK011		IRQ13		
L1		P25	CS5#/ EDACK1	MTIOC4C/ MTCLKB/ TIOCA4/PO5	RXD3/SMISO3/ SSCL3	SDHI_CD	IRQ5	ADTRG0#	TS4/ CLKOUT
L2		P23	EDACK0	MTIOC3D/ MTCLKD/ TIOCD3/PO3	TXD3/CTS0#/ RTS0#/SMOSI3/ SS0#/SSDA3/ SSIBCK0	SDHI_D1-C	IRQ3		TS6

Table 1.5 List of Pin and Pin Functions (0.65-mm Pitch, 145-Pin TFLGA) (6/8)

Pin No.				Timer	Communication				
145-Pin TFLGA (0.65-mm pitch)	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
L3		P16		MTIOC3C/ MTIOC3D/ TIOCB1/ TCLKC/ TMO2/PO14/ RTCOUT	TXD1/RXD3/ SMOSI1/ SMISO3/SSDA1/ SSCL3/SCL2-DS/ USB0_VBUS/ USB0_VBUSEN/ USB0_OVRCURB		IRQ6	ADTRG0#	
L4		P24	CS4#/ EDREQ1	MTIOC4A/ MTCLKA/ TIOCB4/ TMRI1/PO4	SCK3/ USB0_VBUSEN	SDHI_WP	IRQ12		TS5
L5		P13		MTIOC0B/ TIOCA5/ TMO3/PO13	TXD2/SMOSI2/ SSDA2/ SDA0[FM+]/ SDAHS0[FM+/ HS]		IRQ3	ADTRG1#	
L6		P56	EDACK1	MTIOC3C/ TIOCA1	SCK7/ RSPCKC-B		IRQ6		
L7		P52	RD#		RXD2/SMISO2/ SSCL2/SSLB3-A		IRQ2		
L8	TRCLK	P83	EDACK1	MTIOC4C	SS10#/CTS10#/ SCK10/SS010#/ CTS010#/ SCK010		IRQ3		
L9		PC5	D3[A3/D3]/ A21/CS2#/ WAIT#	MTIOC3B/ MTCLKD/ TMRI2/PO29	SCK8/SCK10/ RSPCKA-A/ SSIBCK0/ SCK010/ RSPCK0-A		IRQ5		TS14
L10		PC4	A20/CS3#	MTIOC3D/ MTCLKC/ TMCI1/PO25/ POE0#	SCK5/CTS8#/ RTS8#/SS8#/ SS10#/CTS10#/ RTS10#/ SSLA0-A/ AUDIO_CLK/ SS010#/ CTS010#/ RTS010#/DE010/ SSL00-A	SDHI_D1-A/ QIO1-A	IRQ12		TSCAP
L11		PC2	A18	MTIOC4B/ TCLKA/PO21	RXD5/SMISO5/ SSCL5/SSLA3-A/ TXDB011/ SSL03-A	SDHI_D3-A	IRQ10		
L12	TRDATA4	P73	CS3#	PO16	USB1_VBUS/ USB1_VBUSEN/ USB1_OVRCURB		IRQ8		
L13	VSS								
M1		P22	EDREQ0	MTIOC3B/ MTCLKC/ TIOCC3/ TMO0/PO2	SCK0/ USB0_OVRCURB/ AUDIO_CLK	SDHI_D0-C	IRQ15		TS7

Table 1.5 List of Pin and Pin Functions (0.65-mm Pitch, 145-Pin TFLGA) (7/8)

Pin No.				Timer	Communication				
145-Pin TFLGA (0.65-mm pitch)	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
M2		P17		MTIOC3A/ MTIOC3B/ MTIOC4B/ TIOCB0/ TCLKD/ TMO1/PO15/ POE8#	SCK1/TXD3/ SMOSI3/SSDA3/ SDA2-DS/ SSITXD0	SDHI_D3-C	IRQ7	ADTRG1#	
M3		P86		MTIOC4D/ TIOCA0	SMISO10/ SSCL10/RXD10/ SMISO010/ SSCL010/ RXD010		IRQ14		
M4		P12		MTIC5U/ TMCI1	RXD2/SMISO2/ SSCL2/ SCL0[FM+]/ SCLHS0[FM+/ HS]		IRQ2		
M5	VCC_USB								
M6	VSS_USB								
M7		P50	WR0#/WR#		TXD2/SMOSI2/ SSDA2/SSLB1-A		IRQ0		
M8		PC6	D2[A2/D2]/ A22/CS1#	MTIOC3C/ MTCLKA/ TMCI2/TIC0/ PO30	RXD8/SMISO8/ SSCL8/ SMISO10/ SSCL10/RXD10/ MOSIA-A/ SSILRCK0/ SMISO010/ SSCL010/ RXD010/ MOSI0-A		IRQ13		TS13
M9	TRDATA1	P81	EDACK0	MTIOC3D/ PO27	SMISO10/ SSCL10/RXD10/ SMISO010/ SSCL010/ RXD010/ USB1_OVRCUR B	SDHI_CD/ QIO3-A	IRQ9		
M10	TRDATA7	P77	CS7#	PO23	SMOSI11/ SSDA11/TXD11/ SMOSI011/ SSDA011/ TXD011/ USB1_ID	SDHI_CLK-A/ QSPCLK-A	IRQ7		
M11		PC0	A16	MTIOC3C/ TCLKC/PO17	CTS5#/RTS5#/ SS5#/SSLA1-A/ RXD011/ SMISO011/ SSCL011/ SSL01-A		IRQ14		TS16
M12		PC1	A17	MTIOC3A/ TCLKD/PO18	SCK5/SSLA2-A/ TXD011/ SMOSI011/ SSDA011/ TXDA011/ SSL02-A		IRQ12		TS15
M13	VCC								

Table 1.5 List of Pin and Pin Functions (0.65-mm Pitch, 145-Pin TFLGA) (8/8)

Pin No.				Timer	Communication				
145-Pin TFLGA (0.65-mm pitch)	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
N1		P21		MTIOC1B/ MTIOC4A/ TIOCA3/ TMCI0/PO1	RXD0/SMISO0/ SSCL0/SCL1/ USB0_EXICEN/ SSILRCK0	SDHI_CLK-C	IRQ9		TS8
N2		P20		MTIOC1A/ TIOCB3/ TMRI0/PO0	TXD0/SMOSI0/ SSDA0/SDA1/ USB0_ID/ SSIRXD0	SDHI_CMD-C	IRQ8		TS9
N3		P87		MTIOC4C/ TIOCA2	SMOSI10/ SSDA10/TXD10/ SMOSI010/ SSDA010/ TXD010	SDHI_D2-C	IRQ15		
N4		P14		MTIOC3A/ MTCLKA/ TIOCB5/ TCLKA/ TMRI2/PO15	CTS1#/RTS1#/ SS1#/CTX1/ USB0_OVRCURA		IRQ4		TS11
N5		PH2		TMRI0	USB0_DM		IRQ1		
N6		PH1		TMO0	USB0_DP		IRQ0		
N7	TRDATA3	P55	D0[A0/D0]/ WAIT#/ EDREQ0	MTIOC4D/ TMO3	TXD7/SMOSI7/ SSDA7/CRX1/ MISOC-B		IRQ10		
N8	VSS								
N9	UB	PC7	A23/CS0#	MTIOC3A/ MTCLKB/ TMO2/TOC0/ PO31/ CACREF	TXD8/SMOSI8/ SSDA8/ SMOSI10/ SSDA10/TXD10/ MISOA-A/ SSITXD0/ SMOSI010/ SSDA010/ TXD010/ MISO0-A		IRQ14		
N10	TRSYNC	P82	EDREQ1	MTIOC4A/ PO28	SMOSI10/ SSDA10/TXD10/ SMOSI010/ SSDA010/ TXD010/ USB1_VBUSEN		IRQ2		
N11		PC3	A19	MTIOC4D/ TCLKB/PO24	TXD5/SMOSI5/ SSDA5/ PMC0-DS	SDHI_D0-A/ QIO0-A	IRQ11		
N12	TRSYNC1	P75	CS5#	PO20	SCK11/RTS11#/ SCK011/ RTS011#/DE011/ USB1_OVRCURA	SDHI_D2-A	IRQ13		
N13	TRDATA5	P74	A20/CS4#	PO19	SS11#/CTS11#/ SS011#/ CTS011#/ USB1_VBUSEN		IRQ12		

Note 1. P53 is multiplexed with the BCLK pin function, so cannot be used as an I/O port pin when the external bus is enabled.

1.6.2 145-Pin TFLGA (0.50-mm Pitch)

Table 1.6 List of Pin and Pin Functions (0.50-mm Pitch, 145-Pin TFLGA) (1/8)

Pin No.				Timer	Communication				
145-Pin TFLGA (0.50-mm pitch)	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
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		IRQ0	AN108	
A7		P92	A18	POE4#	RXD7/SMISO7/ SSCL7		IRQ10		
A8		PD2	D2[A2/D2]	MTIOC4D/ TIC2	CRX0/MISOC-A	SDHI_D2-B/ QIO2-B	IRQ2	AN105	
A9		PD6	D6[A6/D6]	MTIC5V/ MTIOC8A/ POE4#	SSLC2-A	SDHI_D0-B/ QIO0-B	IRQ6	AN101	
A10	VSS								
A11		P62	CS2#/ RAS#/ D1[A1/D1]				IRQ2		
A12		PE1	D9[A9/D9]/ D1[A1/D1]	MTIOC4C/ MTIOC3B/ PO18	TXD12/SMOSI12/ SSDA12/ TXDX12/SIOX12/ SSLB2-B		IRQ9	ANEX1	
A13		PE3	D11[A11/ D11]/ D3[A3/D3]	MTIOC4B/ PO26/POE8#/ TOC3	CTS12#/RTS12#/ SS12#		IRQ11		
B1	AVCC1								
B2	AVCC0								
B3		P05					IRQ13		
B4	VREFL0								
B5		P43					IRQ11-DS	AN003	
B6		P47					IRQ15-DS	AN007	
B7		P91	A17		SCK7		IRQ9		
B8		PD0	D0[A0/D0]	POE4#			IRQ0	AN107	
B9		PD4	D4[A4/D4]	MTIOC8B/ POE11#	SSLC0-A	SDHI_CMD-B/ QSSL-B	IRQ4	AN103	
B10	VCC								
B11		P61	CS1#/ SDCS#/ D0[A0/D0]				IRQ1		
B12		PE2	D10[A10/ D10]/ D2[A2/D2]	MTIOC4A/ PO23/TIC3	RXD12/ SMISO12/ SSCL12/ RXDX12/ SSLB3-B		IRQ7-DS		

Table 1.6 List of Pin and Pin Functions (0.50-mm Pitch, 145-Pin TFLGA) (2/8)

Pin No.				Timer	Communication				
145-Pin TFLGA (0.50-mm pitch)	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
B13		PE4	D12[A12/D12]/D4[A4/D4]	MTIOC4D/MTIOC1A/PO28	SSLB0-B		IRQ12		
C1	AVSS1								
C2		P02		TMCI1	SCK6		IRQ10	AN109	
C3	VREFH0								
C4		P41					IRQ9-DS	AN001	
C5		P46					IRQ14-DS	AN006	
C6	VSS								
C7		PD1	D1[A1/D1]	MTIOC4B/POE0#	CTX0/MOSIC-A		IRQ1	AN106	
C8		PD3	D3[A3/D3]	MTIOC8D/POE8#/TOC2	RSPCKC-A	SDHI_D3-B/QIO3-B	IRQ3	AN104	
C9		PD7	D7[A7/D7]	MTIC5U/POE0#	SSLC3-A	SDHI_D1-B/QIO1-B	IRQ7	AN100	
C10		P63	CS3#/CAS#/D2[A2/D2]				IRQ3		
C11		PE0	D8[A8/D8]/D0[A0/D0]	MTIOC3D	SCK12/SSLB1-B		IRQ8	ANEX0	
C12		P70	SDCLK				IRQ0		
C13	VSS								
D1		P00		TMRI0	TXD6/SMOSI6/SSDA6		IRQ8	AN111	
D2		PF5					IRQ4		
D3		P03					IRQ11		
D4		P01		TMCI0	RXD6/SMISO6/SSCL6		IRQ9	AN110	
D5	VCC								
D6		P93	A19	POE0#	CTS7#/RTS7#/SS7#		IRQ11		
D7		PD5	D5[A5/D5]	MTIC5W/MTIOC8C/POE10#	SSLC1-A	SDHI_CLK-B/QSPCLK-B	IRQ5	AN102	
D8		P60	CS0#				IRQ0		
D9		P64	CS4#/WE#/D3[A3/D3]				IRQ4		
D10		PE7	D15[A15/D15]/D7[A7/D7]	MTIOC6A/TOC1	MISOB-B	SDHI_WP/SDHI_D1-B/QIO1-B	IRQ7		
D11	VCC								
D12		PE5	D13[A13/D13]/D5[A5/D5]	MTIOC4C/MTIOC2B	RSPCKB-B		IRQ5		
D13		PE6	D14[A14/D14]/D6[A6/D6]	MTIOC6C/TIC1	MOSIB-B	SDHI_CD/SDHI_D0-B/QIO0-B	IRQ6		
E1	VSS								
E2	VCL								

Table 1.6 List of Pin and Pin Functions (0.50-mm Pitch, 145-Pin TFLGA) (3/8)

Pin No.				Timer	Communication				
145-Pin TFLGA (0.50-mm pitch)	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
E3		PJ5		POE8#	CTS2#/RTS2#/SS2#		IRQ13		
E4	EMLE								
E5		P44					IRQ12-DS	AN004	
E10		PA0	A0/BC0#	MTIOC4A/MTIOC6D/TIOCA0/CACREF/PO16	SSLA1-B/SSL01-B		IRQ0		
E11		P66	CS6#/DQM0	MTIOC7D			IRQ14		
E12		P65	CS5#/CKE				IRQ13		
E13		P67	CS7#/DQM1	MTIOC7C			IRQ15		
F1	XCIN								
F2	XCOUT								
F3	EXCIN	PJ3	EDACK1	MTIOC3C	CTS6#/RTS6#/CTS0#/RTS0#/SS6#/SS0#		IRQ11		
F4	VBATT								
F10		PA3	A3	MTIOC0D/MTCLKD/TIOCD0/TCLKB/PO19	RXD5/SMISO5/SSCL5		IRQ6-DS		
F11	VSS								
F12		PA1	A1	MTIOC0B/MTCLKC/MTIOC7B/TIOCB0/PO17	SCK5/SSLA2-B/SSL02-B/SCK12	SDHI_CD	IRQ11		
F13		PA2	A2	MTIOC7A/PO18	RXD5/SMISO5/SSCL5/SSLA3-B/SSL03-B/RXD12/SMISO12/SSCL12/RXDX12	SDHI_WP	IRQ10		
G1	XTAL	P37							
G2	RES#								
G3	MD/FINED								
G4	BSCANP								
G10		PA5	A5	MTIOC6B/TIOCB1/PO21	RSPCKA-B/RSPCK0-B		IRQ5		
G11		PA6	A6	MTIC5V/MTCLKB/TIOCA2/TMCI3/PO22/POE10#	CTS5#/RTS5#/SS5#/MOSIA-B/MOSI0-B/CTS12#/RTS12#/SS12#		IRQ14		
G12	VCC								
G13		PA4	A4	MTIC5U/MTCLKA/TIOCA1/TMRI0/PO20	TXD5/SMOSI5/SSDA5/SSLA0-B/SSL00-B/TXD12/SMOSI12/SSDA12/TXDX12/SIOX12		IRQ5-DS		

Table 1.6 List of Pin and Pin Functions (0.50-mm Pitch, 145-Pin TFLGA) (4/8)

Pin No.				Timer	Communication				
145-Pin TFLGA (0.50-mm pitch)	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
H1	EXTAL	P36							
H2	VCC								
H3	VSS								
H4	UPSEL	P35					NMI		
H10		P72	A19/CS2#				IRQ10		
H11		P71	A18/CS1#				IRQ1		
H12		PB0	A8	MTIC5W/ TIOCA3/PO24	RXD4/RXD6/ SMISO4/ SMISO6/SSCL4/ SSCL6		IRQ12		
H13		PA7	A7	TIOCB2/PO23	MISOA-B/ MISO0-B		IRQ7		
J1	TRST#	P34		MTIOC0A/ TMCI3/PO12/ POE10#	SCK6/SCK0		IRQ4		TS0
J2		P33	EDREQ1	MTIOC0D/ TIOCD0/ TMRI3/PO11/ POE4#/ POE11#	RXD6/RXD0/ SMISO6/ SMISO0/SSCL6/ SSCL0/CRX0		IRQ3-DS		TS1
J3		P32		MTIOC0C/ TIOCC0/ TMO3/PO10/ RTCOUT/ RTCIC2/ POE0#/ POE10#	TXD6/TXD0/ SMOSI6/ SMOSI0/SSDA6/ SSDA0/CTX0/ USB0_VBUSEN		IRQ2-DS		TAMPI2
J4	TDI	P30		MTIOC4B/ TMRI3/PO8/ RTCIC0/ POE8#	RXD1/SMISO1/ SSCL1/MISOB-A		IRQ0-DS		TAMPI0
J10		PB3	A11	MTIOC0A/ MTIOC4A/ TIOCD3/ TCLKD/ TMO0/PO27/ POE11#	SCK4/SCK6/ PMC0-DS		IRQ3		
J11		PB4	A12	TIOCA4/PO28	CTS9#/RTS9#/ SS9#/SS11#/ CTS11#/RTS11#/ SS011#/ CTS011#/ RTS011#/DE011		IRQ4		
J12		PB2	A10	TIOCC3/ TCLKC/PO26	CTS4#/RTS4#/ CTS6#/RTS6#/ SS4#/SS6#		IRQ2		
J13		PB1	A9	MTIOC0C/ MTIOC4C/ TIOCB3/ TMCI0/PO25	TXD4/TXD6/ SMOSI4/ SMOSI6/SSDA4/ SSDA6		IRQ4-DS		
K1	TCK	P27	CS7#	MTIOC2B/ TMCI3/PO7	SCK1/RSPCKB-A		IRQ7		TS2

Table 1.6 List of Pin and Pin Functions (0.50-mm Pitch, 145-Pin TFLGA) (5/8)

Pin No.				Timer	Communication				
145-Pin TFLGA (0.50-mm pitch)	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
K2	TDO	P26	CS6#	MTIOC2A/ TMO1/PO6	TXD1/CTS3#/ RTS3#/SMOSI1/ SS3#/SSDA1/ MOSIB-A		IRQ6		TS3
K3	TMS	P31		MTIOC4D/ TMCI2/PO9/ RTCIC1	CTS1#/RTS1#/ SS1#/SSLB0-A		IRQ1-DS		TAMPI1
K4		P15		MTIOC0B/ MTCLKB/ TIOCB2/ TCLKB/ TMCI2/PO13	RXD1/SCK3/ SMISO1/SSCL1/ CRX1-DS		IRQ5		TS10
K5	TRDATA2	P54	ALE/ D1[A1/D1]/ EDACK0	MTIOC4B/ TMCI1	CTS2#/RTS2#/ SS2#/CTX1/ MOSIC-B		IRQ4		
K6		P53*1	BCLK		SSIRXD0/ PMC0-DS		IRQ3		TS12
K7		P51	WR1#/ BC1#/ WAIT#		SCK2/SSLB2-A		IRQ1		
K8	VCC								
K9	TRDATA0	P80	EDREQ0	MTIOC3B/ PO26	SCK10/RTS10#/ SCK010/ RTS010#/DE010	SDHI_WP/ QIO2-A	IRQ8		
K10	TRDATA6	P76	CS6#	PO22	SMISO11/ SSCL11/RXD11/ SMISO011/ SSCL011/ RXD011	SDHI_CMD-A/ QSSL-A	IRQ14		
K11		PB7	A15	MTIOC3B/ TIOCB5/PO31	TXD9/SMOSI9/ SSDA9/ SMOSI11/ SSDA11/TXD11/ SMOSI011/ SSDA011/ TXD011		IRQ15		
K12		PB6	A14	MTIOC3D/ TIOCA5/PO30	RXD9/SMISO9/ SSCL9/SMISO11/ SSCL11/RXD11/ SMISO011/ SSCL011/ RXD011		IRQ6		
K13		PB5	A13	MTIOC2A/ MTIOC1B/ TIOCB4/ TMRI1/PO29/ POE4#	SCK9/SCK11/ SCK011		IRQ13		
L1		P25	CS5#/ EDACK1	MTIOC4C/ MTCLKB/ TIOCA4/PO5	RXD3/SMISO3/ SSCL3	SDHI_CD	IRQ5	ADTRG0#	TS4/ CLKOUT
L2		P23	EDACK0	MTIOC3D/ MTCLKD/ TIOCD3/PO3	TXD3/CTS0#/ RTS0#/SMOSI3/ SS0#/SSDA3/ SSIBCK0	SDHI_D1-C	IRQ3		TS6

Table 1.6 List of Pin and Pin Functions (0.50-mm Pitch, 145-Pin TFLGA) (6/8)

Pin No.				Timer	Communication				
145-Pin TFLGA (0.50-mm pitch)	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
L3		P16		MTIOC3C/ MTIOC3D/ TIOCB1/ TCLKC/ TMO2/PO14/ RTCOUT	TXD1/RXD3/ SMOSI1/ SMISO3/SSDA1/ SSCL3/SCL2-DS/ USB0_VBUS/ USB0_VBUSEN/ USB0_OVRCURB		IRQ6	ADTRG0#	
L4		P24	CS4#/ EDREQ1	MTIOC4A/ MTCLKA/ TIOCB4/ TMRI1/PO4	SCK3/ USB0_VBUSEN	SDHI_WP	IRQ12		TS5
L5		P13		MTIOC0B/ TIOCA5/ TMO3/PO13	TXD2/SMOSI2/ SSDA2/ SDA0[FM+]/ SDAHS0[FM+/ HS]		IRQ3	ADTRG1#	
L6		P56	EDACK1	MTIOC3C/ TIOCA1	SCK7/ RSPCKC-B		IRQ6		
L7		P52	RD#		RXD2/SMISO2/ SSCL2/SSLB3-A		IRQ2		
L8	TRCLK	P83	EDACK1	MTIOC4C	SS10#/CTS10#/ SCK10/SS010#/ CTS010#/ SCK010		IRQ3		
L9		PC5	D3[A3/D3]/ A21/CS2#/ WAIT#	MTIOC3B/ MTCLKD/ TMRI2/PO29	SCK8/SCK10/ RSPCKA-A/ SSIBCK0/ SCK010/ RSPCK0-A		IRQ5		TS14
L10		PC4	A20/CS3#	MTIOC3D/ MTCLKC/ TMCI1/PO25/ POE0#	SCK5/CTS8#/ RTS8#/SS8#/ SS10#/CTS10#/ RTS10#/ SSLA0-A/ AUDIO_CLK/ SS010#/ CTS010#/ RTS010#/DE010/ SSL00-A	SDHI_D1-A/ QIO1-A	IRQ12		TSCAP
L11		PC2	A18	MTIOC4B/ TCLKA/PO21	RXD5/SMISO5/ SSCL5/SSLA3-A/ TXDB011/ SSL03-A	SDHI_D3-A	IRQ10		
L12	TRDATA4	P73	CS3#	PO16			IRQ8		
L13	VSS								
M1		P22	EDREQ0	MTIOC3B/ MTCLKC/ TIOCC3/ TMO0/PO2	SCK0/ USB0_OVRCURB/ AUDIO_CLK	SDHI_D0-C	IRQ15		TS7
M2		P17		MTIOC3A/ MTIOC3B/ MTIOC4B/ TIOCB0/ TCLKD/ TMO1/PO15/ POE8#	SCK1/TXD3/ SMOSI3/SSDA3/ SDA2-DS/ SSITXD0	SDHI_D3-C	IRQ7	ADTRG1#	

Table 1.6 List of Pin and Pin Functions (0.50-mm Pitch, 145-Pin TFLGA) (7/8)

Pin No.				Timer	Communication				
145-Pin TFLGA (0.50-mm pitch)	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
M3		P86		MTIOC4D/ TIOCA0	SMISO10/ SSCL10/RXD10/ SMISO010/ SSCL010/ RXD010		IRQ14		
M4		P12		MTIC5U/ TMCI1	RXD2/SMISO2/ SSCL2/ SCL0[FM+]/ SCLHS0[FM+/ HS]		IRQ2		
M5	VCC_USB								
M6	VSS_USB								
M7		P50	WR0#/WR#		TXD2/SMOSI2/ SSDA2/SSLB1-A		IRQ0		
M8		PC6	D2[A2/D2]/ A22/CS1#	MTIOC3C/ MTCLKA/ TMCI2/TIC0/ PO30	RXD8/SMISO8/ SSCL8/ SMISO10/ SSCL10/RXD10/ MOSIA-A/ SSILRCK0/ SMISO010/ SSCL010/ RXD010/ MOSI0-A		IRQ13		TS13
M9	TRDATA1	P81	EDACK0	MTIOC3D/ PO27	SMISO10/ SSCL10/RXD10/ SMISO010/ SSCL010/ RXD010	SDHI_CD/ QIO3-A	IRQ9		
M10	TRDATA7	P77	CS7#	PO23	SMOSI11/ SSDA11/TXD11/ SMOSI011/ SSDA011/ TXD011	SDHI_CLK-A/ QSPCLK-A	IRQ7		
M11		PC0	A16	MTIOC3C/ TCLKC/PO17	CTS5#/RTS5#/ SS5#/SSLA1-A/ RXD011/ SMISO011/ SSCL011/ SSL01-A		IRQ14		TS16
M12		PC1	A17	MTIOC3A/ TCLKD/PO18	SCK5/SSLA2-A/ TXD011/ SMOSI011/ SSDA011/ TXDA011/ SSL02-A		IRQ12		TS15
M13	VCC								
N1		P21		MTIOC1B/ MTIOC4A/ TIOCA3/ TMCI0/PO1	RXD0/SMISO0/ SSCL0/SCL1/ USB0_EXICEN/ SSILRCK0	SDHI_CLK-C	IRQ9		TS8
N2		P20		MTIOC1A/ TIOCB3/ TMRI0/PO0	TXD0/SMOSI0/ SSDA0/SDA1/ USB0_ID/ SSIRXD0	SDHI_CMD-C	IRQ8		TS9

Table 1.6 List of Pin and Pin Functions (0.50-mm Pitch, 145-Pin TFLGA) (8/8)

Pin No.				Timer	Communication				
145-Pin TFLGA (0.50-mm pitch)	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
N3		P87		MTIOC4C/ TIOCA2	SMOSI10/ SSDA10/TXD10/ SMOSI010/ SSDA010/ TXD010	SDHI_D2-C	IRQ15		
N4		P14		MTIOC3A/ MTCLKA/ TIOCB5/ TCLKA/ TMRI2/PO15	CTS1#/RTS1#/ SS1#/CTX1/ USB0_OVRCURA		IRQ4		TS11
N5		PH2		TMRI0	USB0_DM		IRQ1		
N6		PH1		TMO0	USB0_DP		IRQ0		
N7	TRDATA3	P55	D0[A0/D0]/ WAIT#/ EDREQ0	MTIOC4D/ TMO3	TXD7/SMOSI7/ SSDA7/CRX1/ MISOC-B		IRQ10		
N8	VSS								
N9	UB	PC7	A23/CS0#	MTIOC3A/ MTCLKB/ TMO2/TOC0/ PO31/ CACREF	TXD8/SMOSI8/ SSDA8/ SMOSI10/ SSDA10/TXD10/ MISOA-A/ SSITXD0/ SMOSI010/ SSDA010/ TXD010/ MISO0-A		IRQ14		
N10	TRSYNC	P82	EDREQ1	MTIOC4A/ PO28	SMOSI10/ SSDA10/TXD10/ SMOSI010/ SSDA010/ TXD010		IRQ2		
N11		PC3	A19	MTIOC4D/ TCLKB/PO24	TXD5/SMOSI5/ SSDA5/ PMC0-DS	SDHI_D0-A/ QIO0-A	IRQ11		
N12	TRSYNC1	P75	CS5#	PO20	SCK11/RTS11#/ SCK011/ RTS011#/DE011	SDHI_D2-A	IRQ13		
N13	TRDATA5	P74	A20/CS4#	PO19	SS11#/CTS11#/ SS011#/CTS011#		IRQ12		

Note 1. P53 is multiplexed with the BCLK pin function, so cannot be used as an I/O port pin when the external bus is enabled.

1.6.3 144-Pin LFQFP

Table 1.7 List of Pin and Pin Functions (144-Pin LFQFP) (1/8)

Pin No.				Timer	Communication				
144-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
1	AVSS0								
2		P05					IRQ13		
3	AVCC1								
4		P03					IRQ11		
5	AVSS1								
6		P02		TMCI1	SCK6		IRQ10	AN109	
7		P01		TMCI0	RXD6/SMISO6/ SSCL6		IRQ9	AN110	
8		P00		TMRI0	TXD6/SMOSI6/ SSDA6		IRQ8	AN111	
9		PF5					IRQ4		
10	EMLE								
11		PJ5		POE8#	CTS2#/RTS2#/ SS2#		IRQ13		
12	VSS								
13	EXCIN	PJ3	EDACK1	MTIOC3C	CTS6#/RTS6#/ CTS0#/RTS0#/ SS6#/SS0#		IRQ11		
14	VCL								
15	VBATT								
16	MD/FINED								
17	XCIN								
18	XCOUT								
19	RES#								
20	XTAL	P37							
21	VSS								
22	EXTAL	P36							
23	VCC								
24	UPSEL	P35					NMI		
25	TRST#	P34		MTIOC0A/ TMCI3/PO12/ POE10#	SCK6/SCK0		IRQ4		TS0
26		P33	EDREQ1	MTIOC0D/ TIOCD0/ TMRI3/PO11/ POE4#/ POE11#	RXD6/RXD0/ SMISO6/ SMISO0/SSCL6/ SSCL0/CRX0		IRQ3-DS		TS1
27		P32		MTIOC0C/ TIOCC0/ TMO3/PO10/ RTCOUT/ RTCIC2/ POE0#/ POE10#	TXD6/TXD0/ SMOSI6/ SMOSI0/SSDA6/ SSDA0/CTX0/ USB0_VBUSEN		IRQ2-DS		TAMPI2

Table 1.7 List of Pin and Pin Functions (144-Pin LFQFP) (2/8)

Pin No.				Timer	Communication				
144-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
28	TMS	P31		MTIOC4D/ TMCI2/PO9/ RTCIC1	CTS1#/RTS1#/ SS1#/SSLB0-A		IRQ1-DS		TAMPI1
29	TDI	P30		MTIOC4B/ TMRI3/PO8/ RTCIC0/ POE8#	RXD1/SMISO1/ SSCL1/MISOB-A		IRQ0-DS		TAMPI0
30	TCK	P27	CS7#	MTIOC2B/ TMCI3/PO7	SCK1/RSPCKB-A		IRQ7		TS2
31	TDO	P26	CS6#	MTIOC2A/ TMO1/PO6	TXD1/CTS3#/ RTS3#/SMOSI1/ SS3#/SSDA1/ MOSIB-A		IRQ6		TS3
32		P25	CS5#/ EDACK1	MTIOC4C/ MTCLKB/ TIOCA4/PO5	RXD3/SMISO3/ SSCL3	SDHI_CD	IRQ5	ADTRG0#	TS4/ CLKOUT
33		P24	CS4#/ EDREQ1	MTIOC4A/ MTCLKA/ TIOCB4/ TMRI1/PO4	SCK3/ USB0_VBUSEN	SDHI_WP	IRQ12		TS5
34		P23	EDACK0	MTIOC3D/ MTCLKD/ TIOCD3/PO3	TXD3/CTS0#/ RTS0#/SMOSI3/ SS0#/SSDA3/ SSIBCK0	SDHI_D1-C	IRQ3		TS6
35		P22	EDREQ0	MTIOC3B/ MTCLKC/ TIOCC3/ TMO0/PO2	SCK0/ USB0_OVRCURB/ AUDIO_CLK	SDHI_D0-C	IRQ15		TS7
36		P21		MTIOC1B/ MTIOC4A/ TIOCA3/ TMCI0/PO1	RXD0/SMISO0/ SSCL0/SCL1/ USB0_EXICEN/ SSILRCK0	SDHI_CLK-C	IRQ9		TS8
37		P20		MTIOC1A/ TIOCB3/ TMRI0/PO0	TXD0/SMOSI0/ SSDA0/SDA1/ USB0_ID/ SSIRXD0	SDHI_CMD-C	IRQ8		TS9
38		P17		MTIOC3A/ MTIOC3B/ MTIOC4B/ TIOCB0/ TCLKD/ TMO1/PO15/ POE8#	SCK1/TXD3/ SMOSI3/SSDA3/ SDA2-DS/ SSITXD0	SDHI_D3-C	IRQ7	ADTRG1#	
39		P87		MTIOC4C/ TIOCA2	SMOSI10/ SSDA10/TXD10/ SMOSI010/ SSDA010/ TXD010	SDHI_D2-C	IRQ15		
40		P16		MTIOC3C/ MTIOC3D/ TIOCB1/ TCLKC/ TMO2/PO14/ RTCOUT	TXD1/RXD3/ SMOSI1/ SMISO3/SSDA1/ SSCL3/SCL2-DS/ USB0_VBUS/ USB0_VBUSEN/ USB0_OVRCURB		IRQ6	ADTRG0#	

Table 1.7 List of Pin and Pin Functions (144-Pin LFQFP) (3/8)

Pin No.				Timer	Communication				
144-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
41		P86		MTIOC4D/ TIOCA0	SMISO10/ SSCL10/RXD10/ SMISO010/ SSCL010/ RXD010		IRQ14		
42		P15		MTIOC0B/ MTCLKB/ TIOCB2/ TCLKB/ TMCI2/PO13	RXD1/SCK3/ SMISO1/SSCL1/ CRX1-DS		IRQ5		TS10
43		P14		MTIOC3A/ MTCLKA/ TIOCB5/ TCLKA/ TMRI2/PO15	CTS1#/RTS1#/ SS1#/CTX1/ USB0_OVRCURA		IRQ4		TS11
44		P13		MTIOC0B/ TIOCA5/ TMO3/PO13	TXD2/SMOSI2/ SSDA2/ SDA0[FM+]/ SDAHS0[FM+/ HS]		IRQ3	ADTRG1#	
45		P12		MTIC5U/ TMCI1	RXD2/SMISO2/ SSCL2/ SCL0[FM+]/ SCLHS0[FM+/ HS]		IRQ2		
46	VCC_USB								
47		PH2		TMRI0	USB0_DM		IRQ1		
48		PH1		TMO0	USB0_DP		IRQ0		
49	VSS_USB								
50		P56	EDACK1	MTIOC3C/ TIOCA1	SCK7/ RSPCKC-B		IRQ6		
51	TRDATA3	P55	D0[A0/D0]/ WAIT#/ EDREQ0	MTIOC4D/ TMO3	TXD7/SMOSI7/ SSDA7/CRX1/ MISOC-B		IRQ10		
52	TRDATA2	P54	ALE/ D1[A1/D1]/ EDACK0	MTIOC4B/ TMCI1	CTS2#/RTS2#/ SS2#/CTX1/ MOSIC-B		IRQ4		
53		P53*1	BCLK		SSIRXD0/ PMC0-DS		IRQ3		TS12
54		P52	RD#		RXD2/SMISO2/ SSCL2/SSLB3-A		IRQ2		
55		P51	WR1#/ BC1#/ WAIT#		SCK2/SSLB2-A		IRQ1		
56		P50	WR0#/WR#		TXD2/SMOSI2/ SSDA2/SSLB1-A		IRQ0		
57	VSS								
58	TRCLK	P83	EDACK1	MTIOC4C	SS10#/CTS10#/ SCK10/SS010#/ CTS010#/ SCK010		IRQ3		
59	VCC								

Table 1.7 List of Pin and Pin Functions (144-Pin LFQFP) (4/8)

Pin No.				Timer	Communication				
144-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
60	UB	PC7	A23/CS0#	MTIOC3A/ MTCLKB/ TMO2/TOC0/ PO31/ CACREF	TXD8/SMOSI8/ SSDA8/ SMOSI10/ SSDA10/TXD10/ MISOA-A/ SSITXD0/ SMOSI010/ SSDA010/ TXD010/ MISO0-A		IRQ14		
61		PC6	D2[A2/D2]/ A22/CS1#	MTIOC3C/ MTCLKA/ TMCI2/TIC0/ PO30	RXD8/SMISO8/ SSCL8/ SMISO10/ SSCL10/RXD10/ MOSIA-A/ SSILRCK0/ SMISO010/ SSCL010/ RXD010/ MOSI0-A		IRQ13		TS13
62		PC5	D3[A3/D3]/ A21/CS2#/ WAIT#	MTIOC3B/ MTCLKD/ TMRI2/PO29	SCK8/SCK10/ RSPCKA-A/ SSIBCK0/ SCK010/ RSPCK0-A		IRQ5		TS14
63	TRSYNC	P82	EDREQ1	MTIOC4A/ PO28	SMOSI10/ SSDA10/TXD10/ SMOSI010/ SSDA010/ TXD010		IRQ2		
64	TRDATA1	P81	EDACK0	MTIOC3D/ PO27	SMISO10/ SSCL10/RXD10/ SMISO010/ SSCL010/ RXD010	SDHI_CD/ QIO3-A	IRQ9		
65	TRDATA0	P80	EDREQ0	MTIOC3B/ PO26	SCK10/RTS10#/ SCK010/ RTS010#/DE010	SDHI_WP/ QIO2-A	IRQ8		
66		PC4	A20/CS3#	MTIOC3D/ MTCLKC/ TMCI1/PO25/ POE0#	SCK5/CTS8#/ RTS8#/SS8#/ SS10#/CTS10#/ RTS10#/ SSLA0-A/ AUDIO_CLK/ SS010#/ CTS010#/ RTS010#/DE010/ SSL00-A	SDHI_D1-A/ QIO1-A	IRQ12		TSCAP
67		PC3	A19	MTIOC4D/ TCLKB/PO24	TXD5/SMOSI5/ SSDA5/ PMC0-DS	SDHI_D0-A/ QIO0-A	IRQ11		
68	TRDATA7	P77	CS7#	PO23	SMOSI11/ SSDA11/TXD11/ SMOSI011/ SSDA011/ TXD011	SDHI_CLK-A/ QSPCLK-A	IRQ7		

Table 1.7 List of Pin and Pin Functions (144-Pin LFQFP) (5/8)

Pin No.				Timer	Communication				
144-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
69	TRDATA6	P76	CS6#	PO22	SMISO11/ SSCL11/RXD11/ SMISO011/ SSCL011/ RXD011	SDHI_CMD-A/ QSSL-A	IRQ14		
70		PC2	A18	MTIOC4B/ TCLKA/PO21	RXD5/SMISO5/ SSCL5/SSLA3-A/ TXDB011/ SSL03-A	SDHI_D3-A	IRQ10		
71	TRSYNC1	P75	CS5#	PO20	SCK11/RTS11#/ SCK011/ RTS011#/DE011	SDHI_D2-A	IRQ13		
72	TRDATA5	P74	A20/CS4#	PO19	SS11#/CTS11#/ SS011#/CTS011#		IRQ12		
73		PC1	A17	MTIOC3A/ TCLKD/PO18	SCK5/SSLA2-A/ TXD011/ SMOSI011/ SSDA011/ TXDA011/ SSL02-A		IRQ12		TS15
74	VCC								
75		PC0	A16	MTIOC3C/ TCLKC/PO17	CTS5#/RTS5#/ SS5#/SSLA1-A/ RXD011/ SMISO011/ SSCL011/ SSL01-A		IRQ14		TS16
76	VSS								
77	TRDATA4	P73	CS3#	PO16			IRQ8		
78		PB7	A15	MTIOC3B/ TIOCB5/PO31	TXD9/SMOSI9/ SSDA9/ SMOSI11/ SSDA11/TXD11/ SMOSI011/ SSDA011/ TXD011		IRQ15		
79		PB6	A14	MTIOC3D/ TIOCA5/PO30	RXD9/SMISO9/ SSCL9/SMISO11/ SSCL11/RXD11/ SMISO011/ SSCL011/ RXD011		IRQ6		
80		PB5	A13	MTIOC2A/ MTIOC1B/ TIOCB4/ TMRI1/PO29/ POE4#	SCK9/SCK11/ SCK011		IRQ13		
81		PB4	A12	TIOCA4/PO28	CTS9#/RTS9#/ SS9#/SS11#/ CTS11#/RTS11#/ SS011#/ CTS011#/ RTS011#/DE011		IRQ4		

Table 1.7 List of Pin and Pin Functions (144-Pin LFQFP) (6/8)

Pin No.				Timer	Communication				
144-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
82		PB3	A11	MTIOC0A/ MTIOC4A/ TIOCD3/ TCLKD/ TMO0/PO27/ POE11#	SCK4/SCK6/ PMC0-DS		IRQ3		
83		PB2	A10	TIOCC3/ TCLKC/PO26	CTS4#/RTS4#/ CTS6#/RTS6#/ SS4#/SS6#		IRQ2		
84		PB1	A9	MTIOC0C/ MTIOC4C/ TIOCB3/ TMCI0/PO25	TXD4/TXD6/ SMOSI4/ SMOSI6/SSDA4/ SSDA6		IRQ4-DS		
85		P72	A19/CS2#				IRQ10		
86		P71	A18/CS1#				IRQ1		
87		PB0	A8	MTIC5W/ TIOCA3/PO24	RXD4/RXD6/ SMISO4/ SMISO6/SSCL4/ SSCL6		IRQ12		
88		PA7	A7	TIOCB2/PO23	MISOA-B/ MISO0-B		IRQ7		
89		PA6	A6	MTIC5V/ MTCLKB/ TIOCA2/ TMCI3/PO22/ POE10#	CTS5#/RTS5#/ SS5#/MOSIA-B/ MOSI0-B/ CTS12#/RTS12#/ SS12#		IRQ14		
90		PA5	A5	MTIOC6B/ TIOCB1/PO21	RSPCKA-B/ RSPCK0-B		IRQ5		
91	VCC								
92		PA4	A4	MTIC5U/ MTCLKA/ TIOCA1/ TMRI0/PO20	TXD5/SMOSI5/ SSDA5/SSLA0-B/ SSL00-B/TXD12/ SMOSI12/ SSDA12/ TXDX12/SIOX12		IRQ5-DS		
93	VSS								
94		PA3	A3	MTIOC0D/ MTCLKD/ TIOCD0/ TCLKB/PO19	RXD5/SMISO5/ SSCL5		IRQ6-DS		
95		PA2	A2	MTIOC7A/ PO18	RXD5/SMISO5/ SSCL5/SSLA3-B/ SSL03-B/RXD12/ SMISO12/ SSCL12/RXDX12	SDHI_WP	IRQ10		
96		PA1	A1	MTIOC0B/ MTCLKC/ MTIOC7B/ TIOCB0/PO17	SCK5/SSLA2-B/ SSL02-B/SCK12	SDHI_CD	IRQ11		
97		PA0	A0/BC0#	MTIOC4A/ MTIOC6D/ TIOCA0/ CACREF/ PO16	SSLA1-B/ SSL01-B		IRQ0		

Table 1.7 List of Pin and Pin Functions (144-Pin LFQFP) (7/8)

Pin No.				Timer	Communication				
144-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
98		P67	CS7#/ DQM1	MTIOC7C			IRQ15		
99		P66	CS6#/ DQM0	MTIOC7D			IRQ14		
100		P65	CS5#/CKE				IRQ13		
101		PE7	D15[A15/ D15]/ D7[A7/D7]	MTIOC6A/ TOC1	MISOB-B	SDHI_WP/ SDHI_D1-B/ QIO1-B	IRQ7		
102		PE6	D14[A14/ D14]/ D6[A6/D6]	MTIOC6C/ TIC1	MOSIB-B	SDHI_CD/ SDHI_D0-B/ QIO0-B	IRQ6		
103	VCC								
104		P70	SDCLK				IRQ0		
105	VSS								
106		PE5	D13[A13/ D13]/ D5[A5/D5]	MTIOC4C/ MTIOC2B	RSPCKB-B		IRQ5		
107		PE4	D12[A12/ D12]/ D4[A4/D4]	MTIOC4D/ MTIOC1A/ PO28	SSLB0-B		IRQ12		
108		PE3	D11[A11/ D11]/ D3[A3/D3]	MTIOC4B/ PO26/POE8#/ TOC3	CTS12#/RTS12#/ SS12#		IRQ11		
109		PE2	D10[A10/ D10]/ D2[A2/D2]	MTIOC4A/ PO23/TIC3	RXD12/ SMISO12/ SSCL12/ RXDX12/ SSLB3-B		IRQ7-DS		
110		PE1	D9[A9/D9]/ D1[A1/D1]	MTIOC4C/ MTIOC3B/ PO18	TXD12/SMOSI12/ SSDA12/ TXDX12/SIOX12/ SSLB2-B		IRQ9	ANEX1	
111		PE0	D8[A8/D8]/ D0[A0/D0]	MTIOC3D	SCK12/SSLB1-B		IRQ8	ANEX0	
112		P64	CS4#/WE#/ D3[A3/D3]				IRQ4		
113		P63	CS3#/ CAS#/ D2[A2/D2]				IRQ3		
114		P62	CS2#/ RAS#/ D1[A1/D1]				IRQ2		
115		P61	CS1#/ SDCS#/ D0[A0/D0]				IRQ1		
116	VSS								
117		P60	CS0#				IRQ0		
118	VCC								
119		PD7	D7[A7/D7]	MTIC5U/ POE0#	SSLC3-A	SDHI_D1-B/ QIO1-B	IRQ7	AN100	
120		PD6	D6[A6/D6]	MTIC5V/ MTIOC8A/ POE4#	SSLC2-A	SDHI_D0-B/ QIO0-B	IRQ6	AN101	

Table 1.7 List of Pin and Pin Functions (144-Pin LFQFP) (8/8)

Pin No.				Timer	Communication				
144-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC SDRAMC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
121		PD5	D5[A5/D5]	MTIC5W/ MTIOC8C/ POE10#	SSLC1-A	SDHI_CLK-B/ QSPCLK-B	IRQ5	AN102	
122		PD4	D4[A4/D4]	MTIOC8B/ POE11#	SSLC0-A	SDHI_CMD-B/ QSSL-B	IRQ4	AN103	
123		PD3	D3[A3/D3]	MTIOC8D/ POE8#/TOC2	RSPCKC-A	SDHI_D3-B/ QIO3-B	IRQ3	AN104	
124		PD2	D2[A2/D2]	MTIOC4D/ TIC2	CRX0/MISOC-A	SDHI_D2-B/ QIO2-B	IRQ2	AN105	
125		PD1	D1[A1/D1]	MTIOC4B/ POE0#	CTX0/MOSIC-A		IRQ1	AN106	
126		PD0	D0[A0/D0]	POE4#			IRQ0	AN107	
127		P93	A19	POE0#	CTS7#/RTS7#/ SS7#		IRQ11		
128		P92	A18	POE4#	RXD7/SMISO7/ SSCL7		IRQ10		
129		P91	A17		SCK7		IRQ9		
130	VSS								
131		P90	A16		TXD7/SMOSI7/ SSDA7		IRQ0	AN108	
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. P53 is multiplexed with the BCLK pin function, so cannot be used as an I/O port pin when the external bus is enabled.

1.6.4 100-Pin TFLGA

Table 1.8 List of Pin and Pin Functions (100-Pin TFLGA) (1/6)

Pin No.				Timer	Communication				
100-Pin TFLGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
A1		P05					IRQ13		
A2	AVCC1								
A3		P07					IRQ15	ADTRG0#	
A4	VREFL0								
A5		P43					IRQ11-DS	AN003	
A6		PD0	D0[A0/D0]	POE4#			IRQ0	AN107	
A7		PD4	D4[A4/D4]	MTIOC8B/ POE11#	SSLC0-A	SDHI_CMD-B/ QSSL-B	IRQ4	AN103	
A8		PE0	D8[A8/D8]/ D0[A0/D0]	MTIOC3D	SCK12/SSLB1-B		IRQ8	ANEX0	
A9		PE1	D9[A9/D9]/ D1[A1/D1]	MTIOC4C/ MTIOC3B/ PO18	TXD12/SMOSI12/ SSDA12/TXDX12/ SIOX12/SSLB2-B		IRQ9	ANEX1	
A10		PE2	D10[A10/ D10]/ D2[A2/D2]	MTIOC4A/ PO23/TIC3	RXD12/SMISO12/ SSCL12/RXDX12/ SSLB3-B		IRQ7-DS		
B1	EMLE								
B2	AVSS0								
B3	AVCC0								
B4		P40					IRQ8-DS	AN000	
B5		P44					IRQ12-DS	AN004	
B6		PD1	D1[A1/D1]	MTIOC4B/ POE0#	CTX0/MOSIC-A		IRQ1	AN106	
B7		PD3	D3[A3/D3]	MTIOC8D/ POE8#/TOC2	RSPCKC-A	SDHI_D3-B/ QIO3-B	IRQ3	AN104	
B8		PD6	D6[A6/D6]	MTIC5V/ MTIOC8A/ POE4#	SSLC2-A	SDHI_D0-B/ QIO0-B	IRQ6	AN101	
B9		PD7	D7[A7/D7]	MTIC5U/ POE0#	SSLC3-A	SDHI_D1-B/ QIO1-B	IRQ7	AN100	
B10		PE3	D11[A11/ D11]/ D3[A3/D3]	MTIOC4B/ PO26/POE8#/ TOC3	CTS12#/RTS12#/ SS12#		IRQ11		
C1	VCL								
C2	AVSS1								
C3	EXCIN	PJ3	EDACK1	MTIOC3C	CTS6#/RTS6#/ CTS0#/RTS0#/ SS6#/SS0#		IRQ11		
C4	VREFH0								
C5		P42					IRQ10-DS	AN002	
C6		P47					IRQ15-DS	AN007	
C7		PD2	D2[A2/D2]	MTIOC4D/ TIC2	CRX0/MISOC-A	SDHI_D2-B/ QIO2-B	IRQ2	AN105	

Table 1.8 List of Pin and Pin Functions (100-Pin TFLGA) (2/6)

Pin No.				Timer	Communication				
100-Pin TFLGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
C8		PD5	D5[A5/D5]	MTIC5W/ MTIOC8C/ POE10#	SSLC1-A	SDHI_CLK-B/ QSPCLK-B	IRQ5	AN102	
C9		PE5	D13[A13/ D13]/ D5[A5/D5]	MTIOC4C/ MTIOC2B	RSPCKB-B		IRQ5		
C10		PE4	D12[A12/ D12]/ D4[A4/D4]	MTIOC4D/ MTIOC1A/ PO28	SSLB0-B		IRQ12		
D1	XCIN								
D2	XCOUT								
D3	MD/FINED								
D4	VBATT								
D5		P45					IRQ13-DS	AN005	
D6		P46					IRQ14-DS	AN006	
D7		PE6	D14[A14/ D14]/ D6[A6/D6]	MTIOC6C/ TIC1	MOSIB-B	SDHI_CD/ SDHI_D0-B/ QIO0-B	IRQ6		
D8		PE7	D15[A15/ D15]/ D7[A7/D7]	MTIOC6A/ TOC1	MISOB-B	SDHI_WP/ SDHI_D1-B/ QIO1-B	IRQ7		
D9		PA1	A1	MTIOC0B/ MTCLKC/ MTIOC7B/ TIOCB0/PO17	SCK5/SSLA2-B/ SSL02-B/SCK12	SDHI_CD	IRQ11		
D10		PA0	A0/BC0#	MTIOC4A/ MTIOC6D/ TIOCA0/ CACREF/ PO16	SSLA1-B/ SSL01-B		IRQ0		
E1	XTAL	P37							
E2	VSS								
E3	RES#								
E4	TRST#	P34		MTIOC0A/ TMCI3/PO12/ POE10#	SCK6/SCK0		IRQ4		TS0
E5		P41					IRQ9-DS	AN001	
E6		PA2	A2	MTIOC7A/ PO18	RXD5/SMISO5/ SSCL5/SSLA3-B/ SSL03-B/RXD12/ SMISO12/ SSCL12/RXDX12	SDHI_WP	IRQ10		
E7		PA6	A6	MTIC5V/ MTCLKB/ TIOCA2/ TMCI3/PO22/ POE10#	CTS5#/RTS5#/ SS5#/MOSIA-B/ MOSI0-B/CTS12#/ RTS12#/SS12#		IRQ14		
E8		PA4	A4	MTIC5U/ MTCLKA/ TIOCA1/ TMRI0/PO20	TXD5/SMOSI5/ SSDA5/SSLA0-B/ SSL00-B/TXD12/ SMOSI12/ SSDA12/TXDX12/ SIOX12		IRQ5-DS		

Table 1.8 List of Pin and Pin Functions (100-Pin TFLGA) (3/6)

Pin No.				Timer	Communication				
100-Pin TFLGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
E9		PA5	A5	MTIOC6B/ TIOCB1/PO21	RSPCKA-B/ RSPCK0-B		IRQ5		
E10		PA3	A3	MTIOC0D/ MTCLKD/ TIOCD0/ TCLKB/PO19	RXD5/SMISO5/ SSCL5		IRQ6-DS		
F1	EXTAL	P36							
F2	VCC								
F3	UPSEL	P35					NMI		
F4		P32		MTIOC0C/ TIOCC0/ TMO3/PO10/ RTCOUT/ RTCIC2/ POE0#/ POE10#	TXD6/TXD0/ SMOSI6/SMOSI0/ SSDA6/SSDA0/ CTX0/ USB0_VBUSEN		IRQ2-DS		TAMPI2
F5		P12		MTIC5U/ TMCI1	RXD2/SMISO2/ SSCL2/ SCL0[FM+]/ SCLHS0[FM+/HS]		IRQ2		
F6		PB3	A11	MTIOC0A/ MTIOC4A/ TIOCD3/ TCLKD/ TMO0/PO27/ POE11#	SCK6/PMC0-DS		IRQ3		
F7		PB2	A10	TIOCC3/ TCLKC/PO26	CTS6#/RTS6#/ SS6#		IRQ2		
F8		PB0	A8	MTIC5W/ TIOCA3/PO24	RXD6/SMISO6/ SSCL6		IRQ12		
F9		PA7	A7	TIOCB2/PO23	MISOA-B/ MISO0-B		IRQ7		
F10	VSS								
G1		P33	EDREQ1	MTIOC0D/ TIOCD0/ TMRI3/PO11/ POE4#/ POE11#	RXD6/RXD0/ SMISO6/SMISO0/ SSCL6/SSCL0/ CRX0		IRQ3-DS		TS1
G2	TMS	P31		MTIOC4D/ TMCI2/PO9/ RTCIC1	CTS1#/RTS1#/ SS1#/SSLB0-A		IRQ1-DS		TAMPI1
G3	TDI	P30		MTIOC4B/ TMRI3/PO8/ RTCIC0/ POE8#	RXD1/SMISO1/ SSCL1/MISOB-A		IRQ0-DS		TAMPI0
G4	TCK	P27	CS7#	MTIOC2B/ TMCI3/PO7	SCK1/RSPCKB-A		IRQ7		TS2
G5		P53*1	BCLK		SSIRXD0/ PMC0-DS		IRQ3		TS12
G6		P52	RD#		RXD2/SMISO2/ SSCL2/SSLB3-A		IRQ2		

Table 1.8 List of Pin and Pin Functions (100-Pin TFLGA) (4/6)

Pin No.				Timer	Communication				
100-Pin TFLGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
G7		PB5	A13	MTIOC2A/ MTIOC1B/ TIOCB4/ TMRI1/PO29/ POE4#	SCK9/SCK11/ SCK011		IRQ13		
G8		PB4	A12	TIOCA4/PO28	CTS9#/RTS9#/ SS9#/SS11#/ CTS11#/RTS11#/ SS011#/CTS011#/ RTS011#/DE011		IRQ4		
G9		PB1	A9	MTIOC0C/ MTIOC4C/ TIOCB3/ TMCI0/PO25	TXD6/SMOSI6/ SSDA6		IRQ4-DS		
G10	VCC								
H1	TDO	P26	CS6#	MTIOC2A/ TMO1/PO6	TXD1/CTS3#/ RTS3#/SMOSI1/ SS3#/SSDA1/ MOSIB-A		IRQ6		TS3
H2		P25	CS5#/ EDACK1	MTIOC4C/ MTCLKB/ TIOCA4/PO5	RXD3/SMISO3/ SSCL3	SDHI_CD	IRQ5	ADTRG0#	TS4/ CLKOUT
H3		P16		MTIOC3C/ MTIOC3D/ TIOCB1/ TCLKC/ TMO2/PO14/ RTCOUT	TXD1/RXD3/ SMOSI1/SMISO3/ SSDA1/SSCL3/ SCL2-DS/ USB0_VBUS/ USB0_VBUSEN/ USB0_OVRCURB		IRQ6	ADTRG0#	
H4		P15		MTIOC0B/ MTCLKB/ TIOCB2/ TCLKB/ TMCI2/PO13	RXD1/SCK3/ SMISO1/SSCL1/ CRX1-DS		IRQ5		TS10
H5		P55	D0[A0/D0]/ WAIT#/ EDREQ0	MTIOC4D/ TMO3	CRX1/MISOC-B		IRQ10		
H6		P54	ALE/ D1[A1/D1]/ EDACK0	MTIOC4B/ TMCI1	CTS2#/RTS2#/ SS2#/CTX1/ MOSIC-B		IRQ4		
H7	UB	PC7	A23/CS0#	MTIOC3A/ MTCLKB/ TMO2/TOC0/ PO31/ CACREF	TXD8/SMOSI8/ SSDA8/SMOSI10/ SSDA10/TXD10/ MISOA-A/ SSITXD0/ SMOSI010/ SSDA010/ TXD010/MISO0-A		IRQ14		
H8		PC6	D2[A2/D2]/ A22/CS1#	MTIOC3C/ MTCLKA/ TMCI2/TIC0/ PO30	RXD8/SMISO8/ SSCL8/SMISO10/ SSCL10/RXD10/ MOSIA-A/ SSILRCK0/ SMISO010/ SSCL010/ RXD010/MOSI0-A		IRQ13		TS13

Table 1.8 List of Pin and Pin Functions (100-Pin TFLGA) (5/6)

Pin No.				Timer	Communication				
100-Pin TFLGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
H9		PB6	A14	MTIOC3D/ TIOCA5/PO30	RXD9/SMISO9/ SSCL9/SMISO11/ SSCL11/RXD11/ SMISO011/ SSCL011/RXD011		IRQ6		
H10		PB7	A15	MTIOC3B/ TIOCB5/PO31	TXD9/SMOSI9/ SSDA9/SMOSI11/ SSDA11/TXD11/ SMOSI011/ SSDA011/TXD011		IRQ15		
J1		P24	CS4#/ EDREQ1	MTIOC4A/ MTCLKA/ TIOCB4/ TMRI1/PO4	SCK3/ USB0_VBUSEN	SDHI_WP	IRQ12		TS5
J2		P21		MTIOC1B/ MTIOC4A/ TIOCA3/ TMCI0/PO1	RXD0/SMISO0/ SSCL0/SCL1/ USB0_EXICEN/ SSILRCK0	SDHI_CLK-C	IRQ9		TS8
J3		P17		MTIOC3A/ MTIOC3B/ MTIOC4B/ TIOCB0/ TCLKD/ TMO1/PO15/ POE8#	SCK1/TXD3/ SMOSI3/SSDA3/ SDA2-DS/ SSITXD0	SDHI_D3-C	IRQ7	ADTRG1#	
J4		P13		MTIOC0B/ TIOCA5/ TMO3/PO13	TXD2/SMOSI2/ SSDA2/ SDA0[FM+]/ SDAHS0[FM+/HS]		IRQ3	ADTRG1#	
J5	VSS_USB								
J6	VCC_USB								
J7		P50	WR0#/WR#		TXD2/SMOSI2/ SSDA2/SSLB1-A		IRQ0		
J8		PC4	A20/CS3#	MTIOC3D/ MTCLKC/ TMCI1/PO25/ POE0#	SCK5/CTS8#/ RTS8#/SS8#/ SS10#/CTS10#/ RTS10#/SSLA0-A/ AUDIO_CLK/ SS010#/CTS010#/ RTS010#/DE010/ SSL00-A	SDHI_D1-A/ QIO1-A	IRQ12		TSCAP
J9		PC0	A16	MTIOC3C/ TCLKC/PO17	CTS5#/RTS5#/ SS5#/SSLA1-A/ RXD011/ SMISO011/ SSCL011/ SSL01-A		IRQ14		TS16
J10		PC1	A17	MTIOC3A/ TCLKD/PO18	SCK5/SSLA2-A/ TXD011/ SMOSI011/ SSDA011/ TXDA011/ SSL02-A		IRQ12		TS15
K1		P23	EDACK0	MTIOC3D/ MTCLKD/ TIOCD3/PO3	TXD3/CTS0#/ RTS0#/SMOSI3/ SS0#/SSDA3/ SSIBCK0	SDHI_D1-C	IRQ3		TS6

Table 1.8 List of Pin and Pin Functions (100-Pin TFLGA) (6/6)

Pin No.				Timer	Communication				
100-Pin TFLGA	Power Supply Clock System Control	I/O Port	Bus EXDMAC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
K2		P22	EDREQ0	MTIOC3B/ MTCLKC/ TIOCC3/ TMO0/PO2	SCK0/ USB0_OVRCURB/ AUDIO_CLK	SDHI_D0-C	IRQ15		TS7
K3		P20		MTIOC1A/ TIOCB3/ TMRI0/PO0	TXD0/SMOSI0/ SSDA0/SDA1/ USB0_ID/ SSIRXD0	SDHI_CMD-C	IRQ8		TS9
K4		P14		MTIOC3A/ MTCLKA/ TIOCB5/ TCLKA/ TMRI2/PO15	CTS1#/RTS1#/ SS1#/CTX1/ USB0_OVRCURA		IRQ4		TS11
K5		PH2		TMRI0	USB0_DM		IRQ1		
K6		PH1		TMO0	USB0_DP		IRQ0		
K7		P51	WR1#/ BC1#/ WAIT#		SCK2/SSLB2-A		IRQ1		
K8		PC5	D3[A3/D3]/ A21/CS2#/ WAIT#	MTIOC3B/ MTCLKD/ TMRI2/PO29	SCK8/SCK10/ RSPCKA-A/ SSIBCK0/ SCK010/ RSPCK0-A		IRQ5		TS14
K9		PC3	A19	MTIOC4D/ TCLKB/PO24	TXD5/SMOSI5/ SSDA5/PMC0-DS	SDHI_D0-A/ QIO0-A	IRQ11		
K10		PC2	A18	MTIOC4B/ TCLKA/PO21	RXD5/SMISO5/ SSCL5/SSLA3-A/ TXDB011/ SSL03-A	SDHI_D3-A	IRQ10		

Note 1. P53 is multiplexed with the BCLK pin function, so cannot be used as an I/O port pin when the external bus is enabled.

1.6.5 100-Pin LFQFP

Table 1.9 List of Pin and Pin Functions (100-Pin LFQFP) (1/6)

Pin No.				Timer	Communication				
100-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
1	AVCC1								
2	EMLE								
3	AVSS1								
4	EXCIN	PJ3	EDACK1	MTIOC3C	CTS6#/RTS6#/ CTS0#/RTS0#/ SS6#/SS0#		IRQ11		
5	VCL								
6	VBATT								
7	MD/FINED								
8	XCIN								
9	XCOUT								
10	RES#								
11	XTAL	P37							
12	VSS								
13	EXTAL	P36							
14	VCC								
15	UPSEL	P35					NMI		
16	TRST#	P34		MTIOC0A/ TMCI3/PO12/ POE10#	SCK6/SCK0		IRQ4		TS0
17		P33	EDREQ1	MTIOC0D/ TIOCD0/ TMRI3/PO11/ POE4#/ POE11#	RXD6/RXD0/ SMISO6/ SMISO0/SSCL6/ SSCL0/CRX0		IRQ3-DS		TS1
18		P32		MTIOC0C/ TIOCC0/ TMO3/PO10/ RTCOUT/ RTCIC2/ POE0#/ POE10#	TXD6/TXD0/ SMOSI6/ SMOSI0/SSDA6/ SSDA0/CTX0/ USB0_VBUSEN		IRQ2-DS		TAMPI2
19	TMS	P31		MTIOC4D/ TMCI2/PO9/ RTCIC1	CTS1#/RTS1#/ SS1#/SSLB0-A		IRQ1-DS		TAMPI1
20	TDI	P30		MTIOC4B/ TMRI3/PO8/ RTCIC0/ POE8#	RXD1/SMISO1/ SSCL1/MISOB-A		IRQ0-DS		TAMPI0
21	TCK	P27	CS7#	MTIOC2B/ TMCI3/PO7	SCK1/RSPCKB-A		IRQ7		TS2
22	TDO	P26	CS6#	MTIOC2A/ TMO1/PO6	TXD1/CTS3#/ RTS3#/SMOSI1/ SS3#/SSDA1/ MOSIB-A		IRQ6		TS3
23		P25	CS5#/ EDACK1	MTIOC4C/ MTCLKB/ TIOCA4/PO5	RXD3/SMISO3/ SSCL3	SDHI_CD	IRQ5	ADTRG0#	TS4/ CLKOUT

Table 1.9 List of Pin and Pin Functions (100-Pin LFQFP) (2/6)

Pin No.				Timer	Communication				
100-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
24		P24	CS4#/ EDREQ1	MTIOC4A/ MTCLKA/ TIOCB4/ TMRI1/PO4	SCK3/ USB0_VBUSEN	SDHI_WP	IRQ12		TS5
25		P23	EDACK0	MTIOC3D/ MTCLKD/ TIOCD3/PO3	TXD3/CTS0#/ RTS0#/SMOSI3/ SS0#/SSDA3/ SSIBCK0	SDHI_D1-C	IRQ3		TS6
26		P22	EDREQ0	MTIOC3B/ MTCLKC/ TIOCC3/ TMO0/PO2	SCK0/ USB0_OVRCURB/ AUDIO_CLK	SDHI_D0-C	IRQ15		TS7
27		P21		MTIOC1B/ MTIOC4A/ TIOCA3/ TMCI0/PO1	RXD0/SMISO0/ SSCL0/SCL1/ USB0_EXICEN/ SSILRCK0	SDHI_CLK-C	IRQ9		TS8
28		P20		MTIOC1A/ TIOCB3/ TMRI0/PO0	TXD0/SMOSI0/ SSDA0/SDA1/ USB0_ID/ SSIRXD0	SDHI_CMD-C	IRQ8		TS9
29		P17		MTIOC3A/ MTIOC3B/ MTIOC4B/ TIOCB0/ TCLKD/ TMO1/PO15/ POE8#	SCK1/TXD3/ SMOSI3/SSDA3/ SDA2-DS/ SSITXD0	SDHI_D3-C	IRQ7	ADTRG1#	
30		P16		MTIOC3C/ MTIOC3D/ TIOCB1/ TCLKC/ TMO2/PO14/ RTCOUT	TXD1/RXD3/ SMOSI1/ SMISO3/SSDA1/ SSCL3/SCL2-DS/ USB0_VBUS/ USB0_VBUSEN/ USB0_OVRCURB		IRQ6	ADTRG0#	
31		P15		MTIOC0B/ MTCLKB/ TIOCB2/ TCLKB/ TMCI2/PO13	RXD1/SCK3/ SMISO1/SSCL1/ CRX1-DS		IRQ5		TS10
32		P14		MTIOC3A/ MTCLKA/ TIOCB5/ TCLKA/ TMRI2/PO15	CTS1#/RTS1#/ SS1#/CTX1/ USB0_OVRCURA		IRQ4		TS11
33		P13		MTIOC0B/ TIOCA5/ TMO3/PO13	TXD2/SMOSI2/ SSDA2/ SDA0[FM+]/ SDAHS0[FM+/ HS]		IRQ3	ADTRG1#	
34		P12		MTIC5U/ TMCI1	RXD2/SMISO2/ SSCL2/ SCL0[FM+]/ SCLHS0[FM+/ HS]		IRQ2		
35	VCC_USB								
36		PH2		TMRI0	USB0_DM		IRQ1		

Table 1.9 List of Pin and Pin Functions (100-Pin LFQFP) (3/6)

Pin No.				Timer	Communication				
100-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
37		PH1		TMO0	USB0_DP		IRQ0		
38	VSS_USB								
39		P55	D0[A0/D0]/ WAIT#/ EDREQ0	MTIOC4D/ TMO3	CRX1/MISOC-B		IRQ10		
40		P54	ALE/ D1[A1/D1]/ EDACK0	MTIOC4B/ TMCI1	CTS2#/RTS2#/ SS2#/CTX1/ MOSIC-B		IRQ4		
41		P53*1	BCLK		SSIRXD0/ PMC0-DS		IRQ3		TS12
42		P52	RD#		RXD2/SMISO2/ SSCL2/SSLB3-A		IRQ2		
43		P51	WR1#/ BC1#/ WAIT#		SCK2/SSLB2-A		IRQ1		
44		P50	WR0#/WR#		TXD2/SMOSI2/ SSDA2/SSLB1-A		IRQ0		
45	UB	PC7	A23/CS0#	MTIOC3A/ MTCLKB/ TMO2/TOC0/ PO31/ CACREF	TXD8/SMOSI8/ SSDA8/ SMOSI10/ SSDA10/TXD10/ MISOA-A/ SSITXD0/ SMOSI010/ SSDA010/ TXD010/ MISO0-A		IRQ14		
46		PC6	D2[A2/D2]/ A22/CS1#	MTIOC3C/ MTCLKA/ TMCI2/TIC0/ PO30	RXD8/SMISO8/ SSCL8/ SMISO10/ SSCL10/RXD10/ MOSIA-A/ SSILRCK0/ SMISO010/ SSCL010/ RXD010/ MOSI0-A		IRQ13		TS13
47		PC5	D3[A3/D3]/ A21/CS2#/ WAIT#	MTIOC3B/ MTCLKD/ TMRI2/PO29	SCK8/SCK10/ RSPCKA-A/ SSIBCK0/ SCK010/ RSPCK0-A		IRQ5		TS14
48		PC4	A20/CS3#	MTIOC3D/ MTCLKC/ TMCI1/PO25/ POE0#	SCK5/CTS8#/ RTS8#/SS8#/ SS10#/CTS10#/ RTS10#/ SSLA0-A/ AUDIO_CLK/ SS010#/ CTS010#/ RTS010#/DE010/ SSL00-A	SDHI_D1-A/ QIO1-A	IRQ12		TSCAP
49		PC3	A19	MTIOC4D/ TCLKB/PO24	TXD5/SMOSI5/ SSDA5/ PMC0-DS	SDHI_D0-A/ QIO0-A	IRQ11		

Table 1.9 List of Pin and Pin Functions (100-Pin LFQFP) (4/6)

Pin No.				Timer	Communication				
100-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
50		PC2	A18	MTIOC4B/ TCLKA/PO21	RXD5/SMISO5/ SSCL5/SSLA3-A/ TXDB011/ SSL03-A	SDHI_D3-A	IRQ10		
51		PC1	A17	MTIOC3A/ TCLKD/PO18	SCK5/SSLA2-A/ TXD011/ SMOSI011/ SSDA011/ TXDA011/ SSL02-A		IRQ12		TS15
52		PC0	A16	MTIOC3C/ TCLKC/PO17	CTS5#/RTS5#/ SS5#/SSLA1-A/ RXD011/ SMISO011/ SSCL011/ SSL01-A		IRQ14		TS16
53		PB7	A15	MTIOC3B/ TIOCB5/PO31	TXD9/SMOSI9/ SSDA9/ SMOSI11/ SSDA11/TXD11/ SMOSI011/ SSDA011/ TXD011		IRQ15		
54		PB6	A14	MTIOC3D/ TIOCA5/PO30	RXD9/SMISO9/ SSCL9/SMISO11/ SSCL11/RXD11/ SMISO011/ SSCL011/ RXD011		IRQ6		
55		PB5	A13	MTIOC2A/ MTIOC1B/ TIOCB4/ TMRI1/PO29/ POE4#	SCK9/SCK11/ SCK011		IRQ13		
56		PB4	A12	TIOCA4/PO28	CTS9#/RTS9#/ SS9#/SS11#/ CTS11#/RTS11#/ SS011#/ CTS011#/ RTS011#/DE011		IRQ4		
57		PB3	A11	MTIOC0A/ MTIOC4A/ TIOCD3/ TCLKD/ TMO0/PO27/ POE11#	SCK6/PMC0-DS		IRQ3		
58		PB2	A10	TIOCC3/ TCLKC/PO26	CTS6#/RTS6#/ SS6#		IRQ2		
59		PB1	A9	MTIOC0C/ MTIOC4C/ TIOCB3/ TMCI0/PO25	TXD6/SMOSI6/ SSDA6		IRQ4-DS		
60	VCC								
61		PB0	A8	MTIC5W/ TIOCA3/PO24	RXD6/SMISO6/ SSCL6		IRQ12		
62	VSS								

Table 1.9 List of Pin and Pin Functions (100-Pin LFQFP) (5/6)

Pin No.	Power Supply Clock System Control	I/O Port	Bus EXDMAC	Timer	Communication		Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
100-Pin LFQFP				(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)			
63		PA7	A7	TIOCB2/PO23	MISOA-B/ MISO0-B		IRQ7		
64		PA6	A6	MTIC5V/ MTCLKB/ TIOCA2/ TMCI3/PO22/ POE10#	CTS5#/RTS5#/ SS5#/MOSIA-B/ MOSI0-B/ CTS12#/RTS12#/ SS12#		IRQ14		
65		PA5	A5	MTIOC6B/ TIOCB1/PO21	RSPCKA-B/ RSPCK0-B		IRQ5		
66		PA4	A4	MTIC5U/ MTCLKA/ TIOCA1/ TMRI0/PO20	TXD5/SMOSI5/ SSDA5/SSLA0-B/ SSL00-B/TXD12/ SMOSI12/ SSDA12/ TXDX12/SIOX12		IRQ5-DS		
67		PA3	A3	MTIOC0D/ MTCLKD/ TIOCD0/ TCLKB/PO19	RXD5/SMISO5/ SSCL5		IRQ6-DS		
68		PA2	A2	MTIOC7A/ PO18	RXD5/SMISO5/ SSCL5/SSLA3-B/ SSL03-B/RXD12/ SMISO12/ SSCL12/RXDX12	SDHI_WP	IRQ10		
69		PA1	A1	MTIOC0B/ MTCLKC/ MTIOC7B/ TIOCB0/PO17	SCK5/SSLA2-B/ SSL02-B/SCK12	SDHI_CD	IRQ11		
70		PA0	A0/BC0#	MTIOC4A/ MTIOC6D/ TIOCA0/ CACREF/ PO16	SSLA1-B/ SSL01-B		IRQ0		
71		PE7	D15[A15/ D15]/ D7[A7/D7]	MTIOC6A/ TOC1	MISOB-B	SDHI_WP/ SDHI_D1-B/ QIO1-B	IRQ7		
72		PE6	D14[A14/ D14]/ D6[A6/D6]	MTIOC6C/ TIC1	MOSIB-B	SDHI_CD/ SDHI_D0-B/ QIO0-B	IRQ6		
73		PE5	D13[A13/ D13]/ D5[A5/D5]	MTIOC4C/ MTIOC2B	RSPCKB-B		IRQ5		
74		PE4	D12[A12/ D12]/ D4[A4/D4]	MTIOC4D/ MTIOC1A/ PO28	SSLB0-B		IRQ12		
75		PE3	D11[A11/ D11]/ D3[A3/D3]	MTIOC4B/ PO26/POE8#/ TOC3	CTS12#/RTS12#/ SS12#		IRQ11		
76		PE2	D10[A10/ D10]/ D2[A2/D2]	MTIOC4A/ PO23/TIC3	RXD12/ SMISO12/ SSCL12/ RXDX12/ SSLB3-B		IRQ7-DS		
77		PE1	D9[A9/D9]/ D1[A1/D1]	MTIOC4C/ MTIOC3B/ PO18	TXD12/SMOSI12/ SSDA12/ TXDX12/SIOX12/ SSLB2-B		IRQ9	ANEX1	

Table 1.9 List of Pin and Pin Functions (100-Pin LFQFP) (6/6)

Pin No.				Timer	Communication				
100-Pin LFQFP	Power Supply Clock System Control	I/O Port	Bus EXDMAC	(MTU, TPU, TMR, PPG, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, CAN, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, CLKOUT, Tamper detection)
78		PE0	D8[A8/D8]/ D0[A0/D0]	MTIOC3D	SCK12/SSLB1-B		IRQ8	ANEX0	
79		PD7	D7[A7/D7]	MTIC5U/ POE0#	SSLC3-A	SDHI_D1-B/ QIO1-B	IRQ7	AN100	
80		PD6	D6[A6/D6]	MTIC5V/ MTIOC8A/ POE4#	SSLC2-A	SDHI_D0-B/ QIO0-B	IRQ6	AN101	
81		PD5	D5[A5/D5]	MTIC5W/ MTIOC8C/ POE10#	SSLC1-A	SDHI_CLK-B/ QSPCLK-B	IRQ5	AN102	
82		PD4	D4[A4/D4]	MTIOC8B/ POE11#	SSLC0-A	SDHI_CMD-B/ QSSL-B	IRQ4	AN103	
83		PD3	D3[A3/D3]	MTIOC8D/ POE8#/TOC2	RSPCKC-A	SDHI_D3-B/ QIO3-B	IRQ3	AN104	
84		PD2	D2[A2/D2]	MTIOC4D/ TIC2	CRX0/MISOC-A	SDHI_D2-B/ QIO2-B	IRQ2	AN105	
85		PD1	D1[A1/D1]	MTIOC4B/ POE0#	CTX0/MOSIC-A		IRQ1	AN106	
86		PD0	D0[A0/D0]	POE4#			IRQ0	AN107	
87		P47					IRQ15-DS	AN007	
88		P46					IRQ14-DS	AN006	
89		P45					IRQ13-DS	AN005	
90		P44					IRQ12-DS	AN004	
91		P43					IRQ11-DS	AN003	
92		P42					IRQ10-DS	AN002	
93		P41					IRQ9-DS	AN001	
94	VREFL0								
95		P40					IRQ8-DS	AN000	
96	VREFH0								
97	AVCC0								
98		P07					IRQ15	ADTRG0#	
99	AVSS0								
100		P05					IRQ13		

Note 1. P53 is multiplexed with the BCLK pin function, so cannot be used as an I/O port pin when the external bus is enabled.

1.6.6 64-Pin TFBGA

Table 1.10 List of Pin and Pin Functions (64-Pin TFBGA) (1/3)

Pin No.			Timer	Communication				
64-Pin TFBGA	Power Supply Clock System Control	I/O Port	(MTU, TPU, TMR, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, Tamper detection)
A1	AVCC1							
A2	AVSS0							
A3	VREFH0							
A4	VREFL0							
A5		PD2	MTIOC4D/TIC2		SDHI_D2-B/ QIO2-B	IRQ2	AN105	
A6		PD7	MTIC5U/POE0#		SDHI_D1-B/ QIO1-B	IRQ7	AN100	
A7		PE0	MTIOC3D	SCK12/SSLB1-B		IRQ8	ANEX0	
A8		PE2	MTIOC4A/TIC3	RXD12/SMISO12/ SSCL12/RXDX12/ SSLB3-B		IRQ7-DS		
B1	EMLE							
B2	AVSS1							
B3	AVCC0							
B4		P42				IRQ10-DS	AN002	
B5		PD3	MTIOC8D/POE8#/ TOC2		SDHI_D3-B/ QIO3-B	IRQ3	AN104	
B6		PD6	MTIC5V/MTIOC8A/ POE4#		SDHI_D0-B/ QIO0-B	IRQ6	AN101	
B7		PE1	MTIOC4C/ MTIOC3B	TXD12/SMOSI12/ SSDA12/TXDX12/ SIOX12/SSLB2-B		IRQ9	ANEX1	
B8		PE6	MTIOC6C/TIC1	MOSIB-B	SDHI_CD/ SDHI_D0-B/ QIO0-B	IRQ6		
C1	VCL							
C2	VBATT							
C3	MD/FINED							
C4		P41				IRQ9-DS	AN001	
C5		PD4	MTIOC8B/POE11#		SDHI_CMD-B/ QSSL-B	IRQ4	AN103	
C6		PD5	MTIC5W/ MTIOC8C/POE10#		SDHI_CLK-B/ QSPCLK-B	IRQ5	AN102	
C7		PA1	MTIOC0B/ MTCLKC/ MTIOC7B/TIOCB0	SCK5/SSLA2-B/ SSL02-B/SCK12	SDHI_CD	IRQ11		
C8		PE7	MTIOC6A/TOC1	MISOB-B	SDHI_WP/ SDHI_D1-B/ QIO1-B	IRQ7		
D1	XCIN							
D2	XCOUT							
D3	RES#							
D4		P40				IRQ8-DS	AN000	
D5		P43				IRQ11-DS	AN003	
D6		PA6	MTIC5V/MTCLKB/ TIOCA2/TMCI3/ POE10#	CTS5#/RTS5#/ SS5#/MOSIA-B/ MOSI0-B/CTS12#/ RTS12#/SS12#		IRQ14		

Table 1.10 List of Pin and Pin Functions (64-Pin TFBGA) (2/3)

Pin No.			Timer	Communication				
64-Pin TFBGA	Power Supply Clock System Control	I/O Port	(MTU, TPU, TMR, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, Tamper detection)
D7		PA2	MTIOC7A	RXD5/SMISO5/ SSCL5/SSLA3-B/ SSL03-B/RXD12/ SMISO12/SSCL12/ RXDX12	SDHI_WP	IRQ10		
D8		PA4	MTIC5U/MTCLKA/ TIOCA1/TMRI0	TXD5/SMOSI5/ SSDA5/SSLA0-B/ SSL00-B/TXD12/ SMOSI12/SSDA12/ TXDX12/SIOX12		IRQ5-DS		
E1	XTAL	P37						
E2	VSS							
E3	TRST#	P34	MTIOC0A/TMCI3/ POE10#			IRQ4		TS0
E4		P13	MTIOC0B/TIOCA5/ TMO3	TXD2/SMOSI2/ SSDA2/SDA0[FM+]/ SDAHS0[FM+/HS]		IRQ3	ADTRG1#	
E5	BSCANP							
E6		PA7	TIOCB2	MISOA-B/MISO0-B		IRQ7		
E7	VCC							
E8	VSS							
F1	EXTAL	P36						
F2	VCC							
F3	UPSEL	P35				NMI		
F4		P12	TMCI1/MTIC5U	RXD2/SMISO2/ SSCL2/SCL0[FM+]/ SCLHS0[FM+/HS]		IRQ2		
F5		P53		SSIRXD0/PMC0-DS		IRQ3		TS12
F6		PB7	MTIOC3B/TIOCB5	TXD9/SMOSI9/ SSDA9/SMOSI11/ SSDA11/TXD11/ SMOSI011/ SSDA011/TXD011		IRQ15		
F7		PB6	MTIOC3D/TIOCA5	RXD9/SMISO9/ SSCL9/SMISO11/ SSCL11/RXD11/ SMISO011/ SSCL011/RXD011		IRQ6		
F8		PB5	MTIOC2A/ MTIOC1B/TIOCB4/ TMRI1/POE4#	SCK9/SCK11/ SCK011		IRQ13		
G1	TCK	P27	MTIOC2B/TMCI3	SCK1/RSPCKB-A		IRQ7		TS2
G2	TMS	P31	MTIOC4D/TMCI2/ RTCIC1	CTS1#/RTS1#/ SS1#/SSLB0-A		IRQ1-DS		TAMPI1
G3	TDI	P30	MTIOC4B/TMRI3/ RTCIC0/POE8#	RXD1/SMISO1/ SSCL1/MISOB-A		IRQ0-DS		TAMPI0
G4	VCC_USB							
G5	VSS_USB							
G6	UB	PC7	MTIOC3A/ MTCLKB/TMO2/ TOC0/CACREF	TXD8/SMOSI8/ SSDA8/SMOSI10/ SSDA10/TXD10/ MISOA-A/SSITXD0/ SMOSI010/ SSDA010/TXD010/ MISO0-A		IRQ14		

Table 1.10 List of Pin and Pin Functions (64-Pin TFBGA) (3/3)

Pin No.			Timer	Communication				
64-Pin TFBGA	Power Supply Clock System Control	I/O Port	(MTU, TPU, TMR, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, Tamper detection)
G7		PC5	MTIOC3B/ MTCLKD/TMRI2	SCK8/SCK10/ RSPCKA-A/ SSIBCK0/SCK010/ RSPCK0-A		IRQ5		TS14
G8		PC0	MTIOC3C/TCLKC	CTS5#/RTS5#/ SS5#/SSLA1-A/ RXD011/SMISO011/ SSCL011/SSL01-A		IRQ14		TS16
H1	TDO	P26	MTIOC2A/TMO1	TXD1/CTS3#/ RTS3#/SMOSI1/ SS3#/SSDA1/ MOSIB-A		IRQ6		TS3
H2		P17	MTIOC3A/ MTIOC3B/ MTIOC4B/TIOCB0/ TCLKD/TMO1/ POE8#	SCK1/TXD3/ SMOSI3/SSDA3/ SDA2-DS/SSITXD0	SDHI_D3-C	IRQ7	ADTRG1#	
H3		P16	MTIOC3C/ MTIOC3D/TIOCB1/ TCLKC/TMO2/ RTCOUT	TXD1/RXD3/ SMOSI1/SMISO3/ SSDA1/SSCL3/ SCL2-DS/ USB0_VBUS		IRQ6	ADTRG0#	
H4		PH2	TMRI0	USB0_DM		IRQ1		
H5		PH1	TMO0	USB0_DP		IRQ0		
H6		PC6	MTIOC3C/ MTCLKA/TMCI2/ TIC0	RXD8/SMISO8/ SSCL8/SMISO10/ SSCL10/RXD10/ MOSIA-A/ SMISO010/ SSCL010/RXD010/ MOSI0-A/SSILRCK0		IRQ13		TS13
H7		PC4	MTIOC3D/ MTCLKC/TMCI1/ POE0#	SCK5/CTS8#/ RTS8#/SS8#/SS10#/ CTS10#/RTS10#/ SSLA0-A/ AUDIO_CLK/ SS010#/CTS010#/ RTS010#/DE010/ SSL00-A	SDHI_D1-A/ QIO1-A	IRQ12		TSCAP
H8		PC1	MTIOC3A/TCLKD	SCK5/SSLA2-A/ TXD011/SMOSI011/ SSDA011/TXDA011/ SSL02-A		IRQ12		TS15

1.6.7 64-Pin LFQFP

Table 1.11 List of Pin and Pin Functions (64-Pin LFQFP) (1/3)

Pin No.			Timer	Communication				
64-Pin LFQFP	Power Supply Clock System Control	I/O Port	(MTU, TPU, TMR, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, Tamper detection)
1	AVCC1							
2	EMLE							
3	AVSS1							
4	VCL							
5	VBATT							
6	MD/FINED							
7	XCIN							
8	XCOUT							
9	RES#							
10	XTAL	P37						
11	VSS							
12	EXTAL	P36						
13	VCC							
14	UPSEL	P35				NMI		
15	TRST#	P34	MTIOC0A/TMCI3/ POE10#			IRQ4		TS0
16	TDI	P30	MTIOC4B/TMRI3/ RTCIC0/POE8#	RXD1/SMISO1/ SSCL1/MISOB-A		IRQ0-DS		TAMPI0
17	TMS	P31	MTIOC4D/TMCI2/ RTCIC1	CTS1#/RTS1#/ SS1#/SSLB0-A		IRQ1-DS		TAMPI1
18	TDO	P26	MTIOC2A/TMO1	TXD1/CTS3#/ RTS3#/SMOSI1/ SS3#/SSDA1/ MOSIB-A		IRQ6		TS3
19	TCK	P27	MTIOC2B/TMCI3	SCK1/RSPCKB-A		IRQ7		TS2
20		P17	MTIOC3A/ MTIOC3B/ MTIOC4B/TIOCB0/ TCLKD/TMO1/ POE8#	SCK1/TXD3/ SMOSI3/SSDA3/ SDA2-DS/SSITXD0	SDHI_D3-C	IRQ7	ADTRG1#	
21		P16	MTIOC3C/ MTIOC3D/TIOCB1/ TCLKC/TMO2/ RTCOUT	TXD1/RXD3/ SMOSI1/SMISO3/ SSDA1/SSCL3/ SCL2-DS/ USB0_VBUS		IRQ6	ADTRG0#	
22		P13	MTIOC0B/TIOCA5/ TMO3	TXD2/SMOSI2/ SSDA2/SDA0[FM+]/ SDAHS0[FM+/HS]		IRQ3	ADTRG1#	
23		P12	TMCI1/MTIC5U	RXD2/SMISO2/ SSCL2/SCL0[FM+]/ SCLHS0[FM+/HS]		IRQ2		
24	VCC_USB							
25		PH2	TMRI0	USB0_DM		IRQ1		
26		PH1	TMO0	USB0_DP		IRQ0		
27	VSS_USB							
28		P53		SSIRXD0/PMC0-DS		IRQ3		TS12

Table 1.11 List of Pin and Pin Functions (64-Pin LFQFP) (2/3)

Pin No.			Timer	Communication				
64-Pin LFQFP	Power Supply Clock System Control	I/O Port	(MTU, TPU, TMR, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, Tamper detection)
29	UB	PC7	MTIOC3A/ MTCLKB/TMO2/ TOC0/CACREF	TXD8/SMOSI8/ SSDA8/SMOSI10/ SSDA10/TXD10/ MISOA-A/SSITXD0/ SMOSI010/ SSDA010/TXD010/ MISO0-A		IRQ14		
30		PC6	MTIOC3C/ MTCLKA/TMCI2/ TIC0	RXD8/SMISO8/ SSCL8/SMISO10/ SSCL10/RXD10/ MOSIA-A/ SMISO010/ SSCL010/RXD010/ MOSI0-A/SSILRCK0		IRQ13		TS13
31		PC5	MTIOC3B/ MTCLKD/TMRI2	SCK8/SCK10/ RSPCKA-A/ SSIBCK0/SCK010/ RSPCK0-A		IRQ5		TS14
32		PC4	MTIOC3D/ MTCLKC/TMCI1/ POE0#	SCK5/CTS8#/ RTS8#/SS8#/SS10#/ CTS10#/RTS10#/ SSLA0-A/ AUDIO_CLK/ SS010#/CTS010#/ RTS010#/DE010/ SSL00-A	SDHI_D1-A/ QIO1-A	IRQ12		TSCAP
33		PC1	MTIOC3A/TCLKD	SCK5/SSLA2-A/ TXD011/SMOSI011/ SSDA011/TXDA011/ SSL02-A		IRQ12		TS15
34		PC0	MTIOC3C/TCLKC	CTS5#/RTS5#/ SS5#/SSLA1-A/ RXD011/SMISO011/ SSCL011/SSL01-A		IRQ14		TS16
35		PB7	MTIOC3B/TIOCB5	TXD9/SMOSI9/ SSDA9/SMOSI11/ SSDA11/TXD11/ SMOSI011/ SSDA011/TXD011		IRQ15		
36		PB6	MTIOC3D/TIOCA5	RXD9/SMISO9/ SSCL9/SMISO11/ SSCL11/RXD11/ SMISO011/ SSCL011/RXD011		IRQ6		
37		PB5	MTIOC2A/ MTIOC1B/TIOCB4/ TMRI1/POE4#	SCK9/SCK11/ SCK011		IRQ13		
38	VCC							
39	VSS							
40		PA7	TIOCB2	MISOA-B/MISO0-B		IRQ7		
41		PA6	MTIC5V/MTCLKB/ TIOCA2/TMCI3/ POE10#	CTS5#/RTS5#/ SS5#/MOSIA-B/ MOSI0-B/CTS12#/ RTS12#/SS12#		IRQ14		

Table 1.11 List of Pin and Pin Functions (64-Pin LFQFP) (3/3)

Pin No.			Timer	Communication				
64-Pin LFQFP	Power Supply Clock System Control	I/O Port	(MTU, TPU, TMR, RTC, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, USB, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU, Tamper detection)
42		PA4	MTIC5U/MTCLKA/ TIOCA1/TMRI0	TXD5/SMOSI5/ SSDA5/SSLA0-B/ SSL00-B/TXD12/ SMOSI12/SSDA12/ TXDX12/SIOX12		IRQ5-DS		
43		PA2	MTIOC7A	RXD5/SMISO5/ SSCL5/SSLA3-B/ SSL03-B/RXD12/ SMISO12/SSCL12/ RXDX12	SDHI_WP	IRQ10		
44		PA1	MTIOC0B/ MTCLKC/ MTIOC7B/TIOCB0	SCK5/SSLA2-B/ SSL02-B/SCK12	SDHI_CD	IRQ11		
45		PE7	MTIOC6A/TOC1	MISOB-B	SDHI_WP/ SDHI_D1-B/ QIO1-B	IRQ7		
46		PE6	MTIOC6C/TIC1	MOSIB-B	SDHI_CD/ SDHI_D0-B/ QIO0-B	IRQ6		
47		PE2	MTIOC4A/TIC3	RXD12/SMISO12/ SSCL12/RXDX12/ SSLB3-B		IRQ7-DS		
48		PE1	MTIOC4C/ MTIOC3B	TXD12/SMOSI12/ SSDA12/TXDX12/ SIOX12/SSLB2-B		IRQ9	ANEX1	
49		PE0	MTIOC3D	SCK12/SSLB1-B		IRQ8	ANEX0	
50		PD7	MTIC5U/POE0#		SDHI_D1-B/ QIO1-B	IRQ7	AN100	
51		PD6	MTIC5V/MTIOC8A/ POE4#		SDHI_D0-B/ QIO0-B	IRQ6	AN101	
52		PD5	MTIC5W/ MTIOC8C/POE10#		SDHI_CLK-B/ QSPCLK-B	IRQ5	AN102	
53		PD4	MTIOC8B/POE11#		SDHI_CMD-B/ QSSL-B	IRQ4	AN103	
54		PD3	MTIOC8D/POE8#/ TOC2		SDHI_D3-B/ QIO3-B	IRQ3	AN104	
55		PD2	MTIOC4D/TIC2		SDHI_D2-B/ QIO2-B	IRQ2	AN105	
56		P43				IRQ11-DS	AN003	
57		P42				IRQ10-DS	AN002	
58		P41				IRQ9-DS	AN001	
59	VREFL0							
60		P40				IRQ8-DS	AN000	
61	VREFH0							
62	AVCC0							
63	AVSS0							
64		P05				IRQ13		

1.6.8 48-Pin HWQFN

Table 1.12 List of Pin and Pin Functions (48-Pin HWQFN) (1/3)

Pin No.			Timer	Communication				
48-Pin HWQFN	Power Supply Clock System Control	I/O Port	(MTU, TPU, TMR, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU)
1	EMLE							
2	MD/FINED							
3	RES#							
4	XTAL	P37						
5	VSS							
6	EXTAL	P36						
7	VCC							
8	UPSEL	P35				NMI		
9	TRST#	P34	MTIOC0A/TMCI3/ POE10#			IRQ4		TS0
10	TMS	P31	MTIOC4D/TMCI2	CTS1#/RTS1#/ SS1#/SSLB0-A		IRQ1-DS		
11	TDI	P30	MTIOC4B/TMRI3/ POE8#	RXD1/SMISO1/ SSCL1/MISOB-A		IRQ0-DS		
12	TCK	P27	MTIOC2B/TMCI3	SCK1/RSPCKB-A		IRQ7		TS2
13	TDO	P26	MTIOC2A/TMO1	TXD1/CTS3#/ RTS3#/SMOSI1/ SS3#/SSDA1/ MOSIB-A		IRQ6		TS3
14		P17	MTIOC3A/ MTIOC3B/ MTIOC4B/TIOCB0/ TCLKD/TMO1/ POE8#	SCK1/TXD3/ SMOSI3/SSDA3/ SDA2-DS/SSITXD0	SDHI_D3-C	IRQ7	ADTRG1#	
15		P16	MTIOC3C/ MTIOC3D/TIOCB1/ TCLKC/TMO2	TXD1/RXD3/ SMOSI1/SMISO3/ SSDA1/SSCL3/ SCL2-DS		IRQ6	ADTRG0#	
16		P13	MTIOC0B/TIOCA5/ TMO3	TXD2/SMOSI2/ SSDA2/SDA0[FM+]/ SDAHS0[FM+/HS]		IRQ3	ADTRG1#	
17		P12	TMCI1/MTIC5U	RXD2/SMISO2/ SSCL2/SCL0[FM+]/ SCLHS0[FM+/HS]		IRQ2		
18	VCC							
19	VSS							
20		P53		SSIRXD0/PMC0-DS		IRQ3		TS12
21	UB	PC7	MTIOC3A/ MTCLKB/TMO2/ TOC0/CACREF	TXD8/SMOSI8/ SSDA8/SMOSI10/ SSDA10/TXD10/ MISOA-A/SSITXD0/ SMOSI010/ SSDA010/TXD010/ MISO0-A		IRQ14		
22		PC6	MTIOC3C/ MTCLKA/TMCI2/ TIC0	RXD8/SMISO8/ SSCL8/SMISO10/ SSCL10/RXD10/ MOSIA-A/ SSILRCK0/ SMISO010/ SSCL010/RXD010/ MOSI0-A		IRQ13		TS13

Table 1.12 List of Pin and Pin Functions (48-Pin HWQFN) (2/3)

Pin No.	Power Supply Clock System Control	I/O Port	Timer	Communication		Interrupt	A/D	Others
48-Pin HWQFN			(MTU, TPU, TMR, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, SSIE, REMC)	(QSPIX, SDHI)			(CTSU)
23		PC5	MTIOC3B/ MTCLKD/TMRI2	SCK8/SCK10/ RSPCKA-A/ SSIBCK0/SCK010/ RSPCK0-A		IRQ5		TS14
24		PC4	MTIOC3D/ MTCLKC/TMCI1/ POE0#	SCK5/CTS8#/ RTS8#/SS8#/SS10#/ CTS10#/RTS10#/ SSLA0-A/ AUDIO_CLK/ SS010#/CTS010#/ RTS010#/DE010/ SSL00-A	SDHI_D1-A/ QIO1-A	IRQ12		TSCAP
25		PB7	MTIOC3B/TIOCB5	TXD9/SMOSI9/ SSDA9/SMOSI11/ SSDA11/TXD11/ SMOSI011/ SSDA011/TXD011		IRQ15		
26		PB6	MTIOC3D/TIOCA5	RXD9/SMISO9/ SSCL9/SMISO11/ SSCL11/RXD11/ SMISO011/ SSCL011/RXD011		IRQ6		
27		PB5	MTIOC2A/ MTIOC1B/TIOCB4/ TMRI1/POE4#	SCK9/SCK11/ SCK011		IRQ13		
28	VCC							
29	VSS							
30	VCL							
31		PA6	MTIC5V/MTCLKB/ TIOCA2/TMCI3/ POE10#	CTS5#/RTS5#/ SS5#/MOSIA-B/ MOSI0-B/CTS12#/ RTS12#/SS12#		IRQ14		
32		PA4	MTIC5U/MTCLKA/ TIOCA1/TMRI0	TXD5/SMOSI5/ SSDA5/SSLA0-B/ SSL00-B/TXD12/ SMOSI12/SSDA12/ TXDX12/SIOX12		IRQ5-DS		
33		PA2	MTIOC7A	RXD5/SMISO5/ SSCL5/SSLA3-B/ SSL03-B/RXD12/ SMISO12/SSCL12/ RXDX12	SDHI_WP	IRQ10		
34		PA1	MTIOC0B/ MTCLKC/ MTIOC7B/TIOCB0	SCK5/SSLA2-B/ SSL02-B/SCK12	SDHI_CD	IRQ11		
35		PE7	MTIOC6A/TOC1	MISOB-B	SDHI_WP/ SDHI_D1-B/ QIO1-B	IRQ7		
36		PE6	MTIOC6C/TIC1	MOSIB-B	SDHI_CD/ SDHI_D0-B/ QIO0-B	IRQ6		
37		PD5	MTIC5W/ MTIOC8C/POE10#		SDHI_CLK-B/ QSPCLK-B	IRQ5	AN102	
38		PD4	MTIOC8B/POE11#		SDHI_CMD-B/ QSSL-B	IRQ4	AN103	
39		PD3	MTIOC8D/POE8#/ TOC2		SDHI_D3-B/ QIO3-B	IRQ3	AN104	

Table 1.12 List of Pin and Pin Functions (48-Pin HWQFN) (3/3)

Pin No.			Timer	Communication				
48-Pin HWQFN	Power Supply Clock System Control	I/O Port	(MTU, TPU, TMR, CMTW, POE, CAC)	(SCI, RSCI, RSPI, RSPIA, RIIC, RIICHS, SSIE, REMC)	(QSPIX, SDHI)	Interrupt	A/D	Others (CTSU)
40		PD2	MTIOC4D/TIC2		SDHI_D2-B/ QIO2-B	IRQ2	AN105	
41		P43				IRQ11-DS	AN003	
42		P42				IRQ10-DS	AN002	
43		P41				IRQ9-DS	AN001	
44	VREFL0							
45		P40				IRQ8-DS	AN000	
46	VREFH0							
47	AVCC0/ AVCC1							
48	AVSS0/AVSS1							

2. CPU

The RXv3 CPU is based on the RXv3 instruction set architecture. Its instruction processing efficiency has been improved relative to that of the RXv2 CPU, so it delivers higher performance.
The RXv3 instruction set architecture (RXv3) provides upward compatibility from the RXv2 instruction set architecture (RXv2) and the RXv1 instruction set architecture (RXv1).

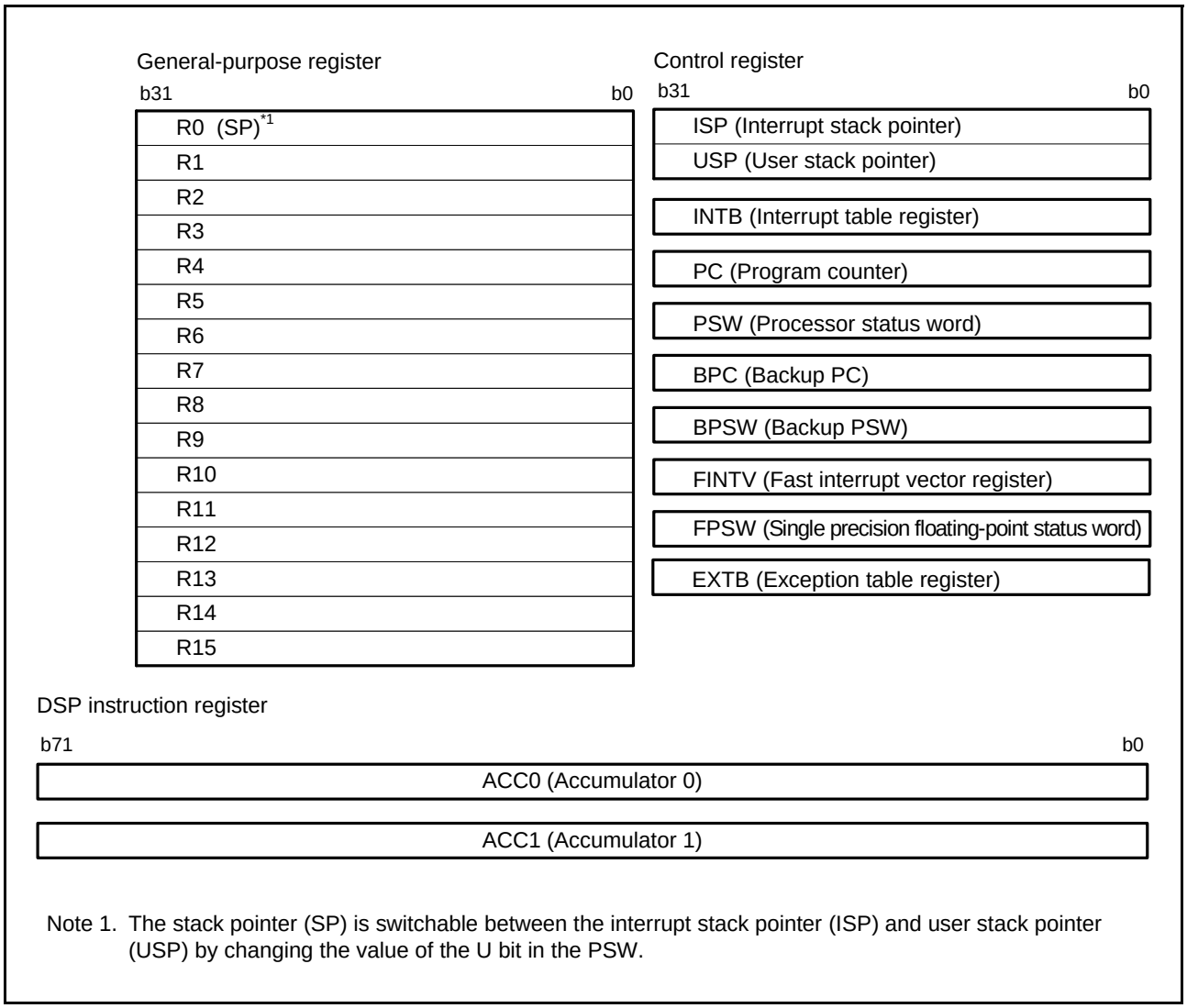
- Adoption of variable-length instruction format
  The CPU has short formats for frequently used instructions, facilitating the development of efficient programs that take up less memory.
- Powerful instruction set
  DSP instructions and floating-point operation instructions realize high-speed arithmetic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

2.1 Features

- Minimum instruction execution rate: One clock cycle
- Address space: 4-Gbyte linear addresses
- Register set of the CPU
  General purpose: Sixteen 32-bit registers
  Control: Ten 32-bit registers
  Accumulator: Two 72-bit registers
- Variable-length instruction format (lengths from one to eight bytes)
- 113 instructions
  Standard provided instructions: 111
    Basic instructions: 77
    Single-precision floating point instructions: 11
    DSP instructions: 23
  Instructions for register bank save function: 2
- Processor modes
  Supervisor mode and user mode
- Vector tables
  Exception vector table and interrupt vector table
- Memory protection unit
- Data arrangement
  Selectable as little endian or big endian
- On-chip double-precision floating point coprocessor
  Double-precision floating-point processing instructions: 21

2.2 Register Set of the CPU

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



Note 1. The stack pointer (SP) is switchable between the interrupt stack pointer (ISP) and user stack pointer (USP) by changing the value of the U bit in the PSW.

Figure 2.1 Register Set of the CPU

2.2.1 General-Purpose Registers (R0 to R15)

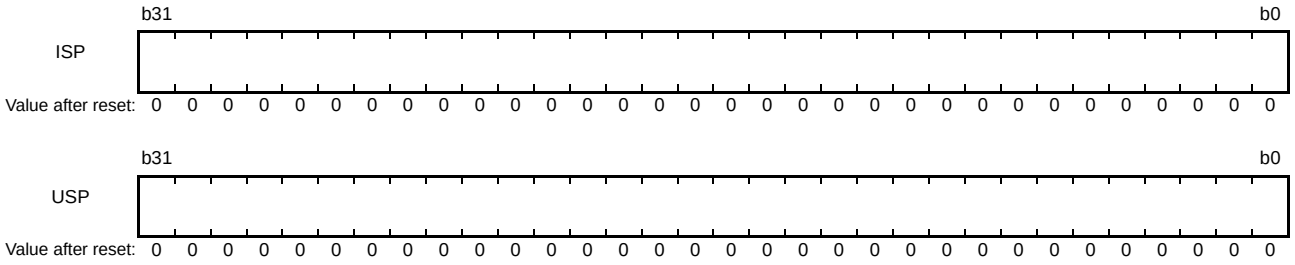
This CPU has sixteen 32-bit general-purpose registers (R0 to R15). R0 to R15 can be used as data registers or address registers.
R0, a general-purpose register, also functions as the stack pointer (SP).
The stack pointer is switched to operate as the interrupt stack pointer (ISP) or user stack pointer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

2.2.2 Control Registers

This CPU has the following ten control registers.

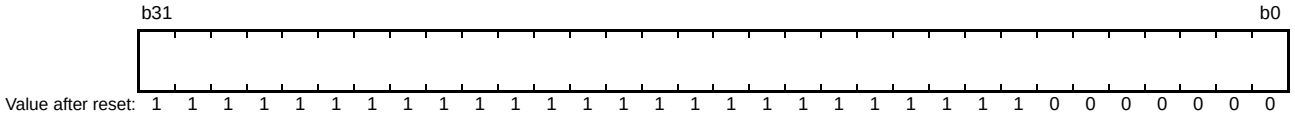
- Interrupt stack pointer (ISP)
- User stack pointer (USP)
- Exception table register (EXTB)
- Interrupt table register (INTB)
- Program counter (PC)
- Processor status word (PSW)
- Backup PC (BPC)
- Backup PSW (BPSW)
- Fast interrupt vector register (FINTV)
- Single-precision floating-point status word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

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



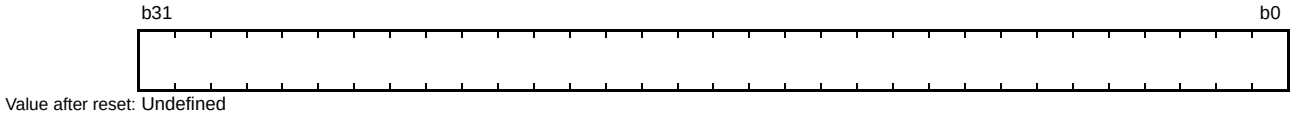
The stack pointer (SP) can be either of two types, the interrupt stack pointer (ISP) or the user stack pointer (USP). Whether the stack pointer operates as the ISP or USP depends on the value of the stack pointer select bit (U) in the processor status word (PSW).

2.2.2.2 Exception Table Register (EXTB)



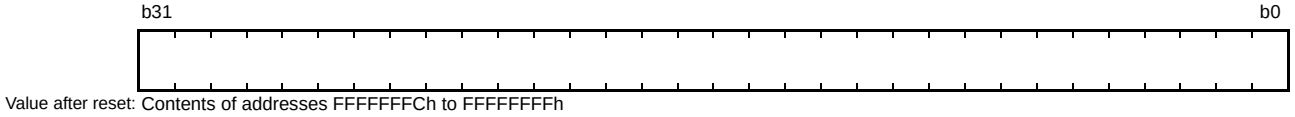
The exception table register (EXTB) specifies the address where the exception vector table starts.

2.2.2.3 Interrupt Table Register (INTB)



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

2.2.2.4 Program Counter (PC)



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

2.2.2.5 Processor Status Word (PSW)

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

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

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

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

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

C Flag (Carry Flag)

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

Z Flag (Zero Flag)

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

S Flag (Sign Flag)

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

O Flag (Overflow Flag)

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

I Bit (Interrupt Enable)

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

U Bit (Stack Pointer Select)

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

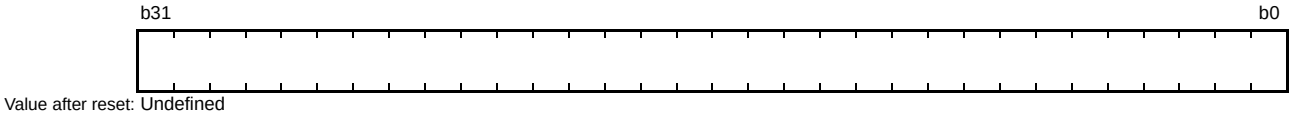
PM Bit (Processor Mode Select)

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

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

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

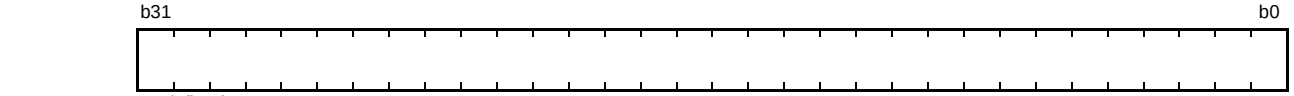
2.2.2.6 Backup PC (BPC)



Value after reset: Undefined

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

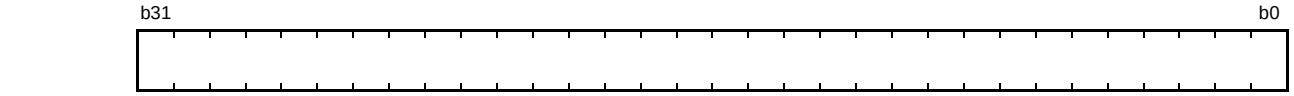
2.2.2.7 Backup PSW (BPSW)



Value after reset: Undefined

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

2.2.2.8 Fast Interrupt Vector Register (FINTV)



Value after reset: Undefined

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

2.2.2.9 Single-Precision 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]	Single-Precision Floating-Point Rounding-Mode Setting	b1 b0 0 0: Rounding towards the nearest value 0 1: Rounding towards 0 1 0: Rounding towards +∞ 1 1: Rounding towards –∞	R/W
b2	CV	Invalid Operation Cause Flag	0: No invalid operation has been encountered. 1: Invalid operation has been encountered.	R/(W) *1
b3	CO	Overflow Cause Flag	0: No overflow has occurred. 1: Overflow has occurred.	R/(W) *1
b4	CZ	Division-by-Zero Cause Flag	0: No division-by-zero has occurred. 1: Division-by-zero has occurred.	R/(W) *1
b5	CU	Underflow Cause Flag	0: No underflow has occurred. 1: Underflow has occurred.	R/(W) *1
b6	CX	Inexact Cause Flag	0: No inexact exception has been generated. 1: Inexact exception has been generated.	R/(W) *1
b7	CE	Unimplemented Processing Cause Flag	0: No unimplemented processing has been encountered. 1: Unimplemented process has been encountered.	R/(W) *1
b8	DN	0 Flush Bit of Denormalized Number	0: A denormalized number is handled as a denormalized number. 1: A denormalized number is handled as 0.*2	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	EV	Invalid Operation Exception Enable	0: Invalid operation exception is masked. 1: Invalid operation exception is enabled.	R/W
b11	EO	Overflow Exception Enable	0: Overflow exception is masked. 1: Overflow exception is enabled.	R/W
b12	EZ	Division-by-Zero Exception Enable	0: Division-by-zero exception is masked. 1: Division-by-zero exception is enabled.	R/W
b13	EU	Underflow Exception Enable	0: Underflow exception is masked. 1: Underflow exception is enabled.	R/W
b14	EX	Inexact Exception Enable	0: Inexact exception is masked. 1: Inexact exception is enabled.	R/W
b25 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26	FV*3	Invalid Operation Flag	0: No invalid operation has been encountered. 1: Invalid operation has been encountered.*8	R/W
b27	FO*4	Overflow Flag	0: No overflow has occurred. 1: Overflow has occurred.*8	R/W
b28	FZ*5	Division-by-Zero Flag	0: No division-by-zero has occurred. 1: Division-by-zero has occurred.*8	R/W
b29	FU*6	Underflow Flag	0: No underflow has occurred. 1: Underflow has occurred.*8	R/W
b30	FX*7	Inexact Flag	0: No inexact exception has been generated. 1: Inexact exception has been generated.*8	R/W
b31	FS	Single-Precision Floating-Point Error Summary Flag	This bit reflects the logical OR of the FU, FZ, FO, and FV flags.	R

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

The single-precision floating-point status word (FPSW) indicates the results of single-precision floating-point arithmetic operations.
When the corresponding exception handling enable bits (Ej) are set to enable processing of the exceptions (Ej = 1), the Cj flags can be used by the exception handling routine to identify the source of that exception. If handling of an exception is masked (Ej = 0), the Fj flag can be used to check for the generation of the exception at the end of a sequence of processing. The Fj flags operate in an accumulative fashion (j = X, U, Z, O, or V).

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

These bits specify the single-precision floating-point rounding-mode.

Explanation of Single-Precision 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 of a hypothetical calculation with infinite precision. 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)

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

- If an exception or processing that is not implemented is not encountered in the execution of a single-precision floating-point arithmetic instruction, the corresponding flags become 0.
- 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 Bit (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 single-precision floating-point exceptions specified in the IEEE754 standard is generated by the single-precision 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 single-precision floating-point exceptions specified in the IEEE754 standard is generated, the corresponding bit is set to 1.

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

FS Flag (Single-Precision Floating-Point Error Summary Flag)

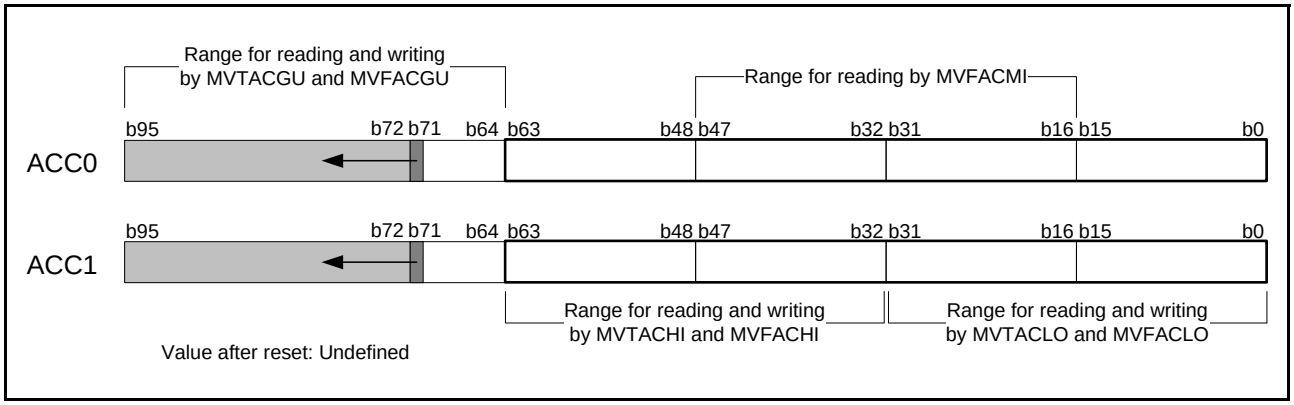
This bit reflects the logical OR of the FU, FZ, FO, and FV flags.

2.2.3 Accumulator

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

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

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



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

2.3 Processor Mode

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

2.3.1 Supervisor Mode

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

2.3.2 User Mode

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

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

2.3.3 Privileged Instruction

Privileged instructions can only be executed in supervisor mode. Executing a privileged instruction in user mode produces a privileged instruction exception. Privileged instructions include the RTFI, MVTIPL, RTE, WAIT, SAVE, and RSTR 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 CPU can handle four types of data: integer, single-precision floating-point number, bit, and string.
For details, refer to RX Family RXv3 Instruction Set Architecture User's Manual: Software.

2.4.1 Integer

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

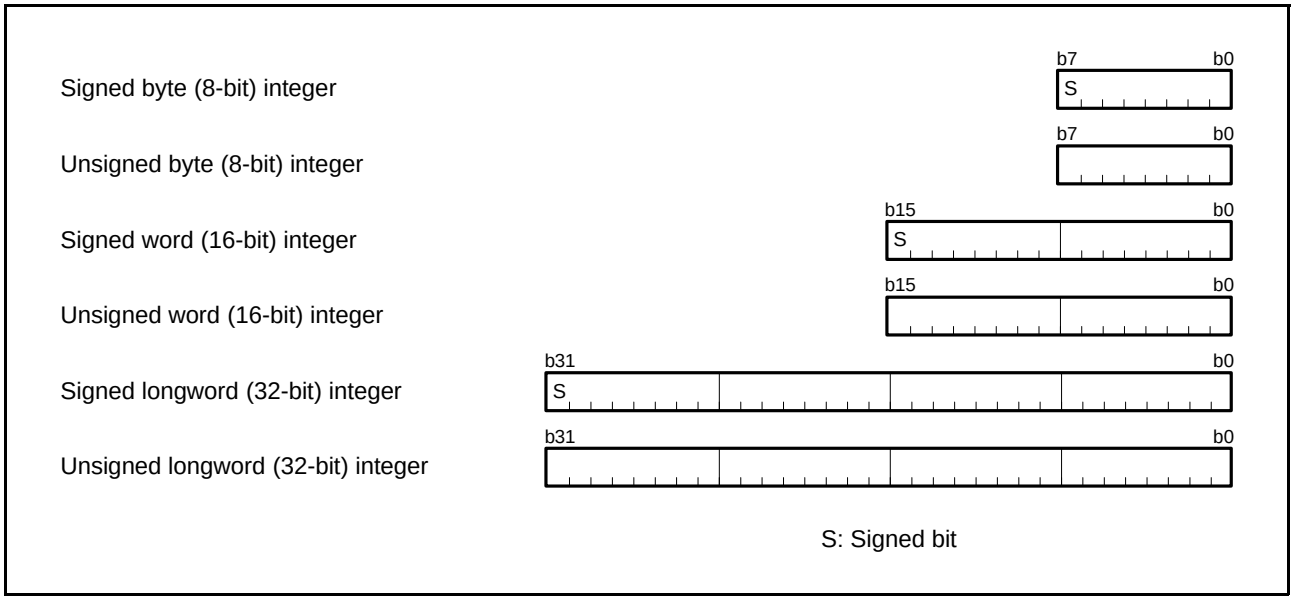


Figure 2.2 Integer

2.4.2 Single-Precision Floating-Point Numbers

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

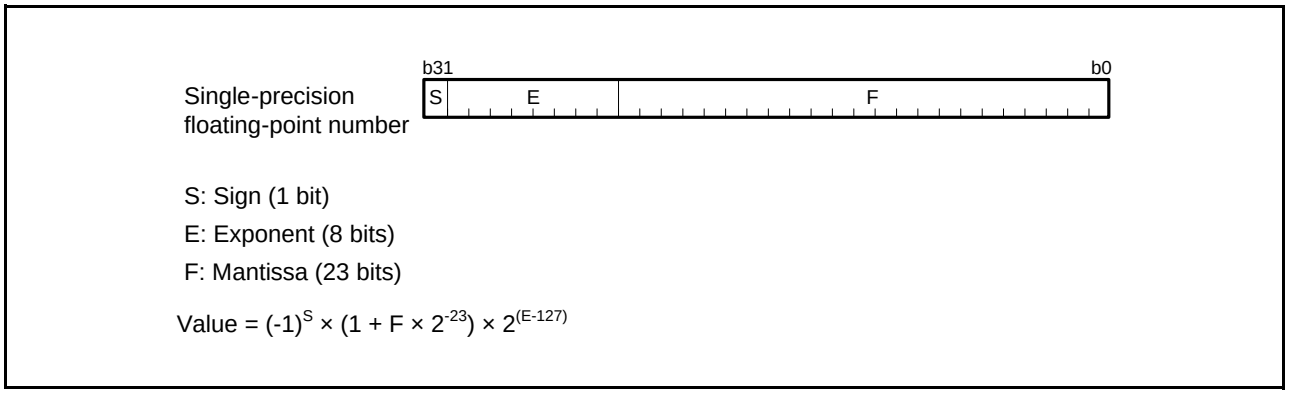


Figure 2.3 Single-Precision Floating-Point Number

The single-precision floating-point number can represent the values listed below.

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

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

2.4.3 Bitwise Operations

Five bit-manipulation instructions are provided for bitwise operations: BCLR, BM*Cnd*, 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.

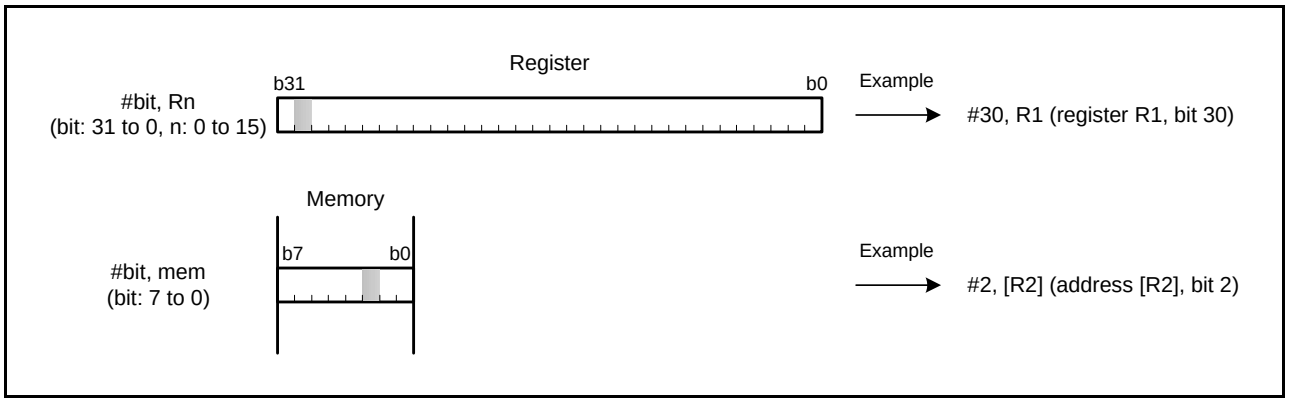


Figure 2.4 Bit

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.

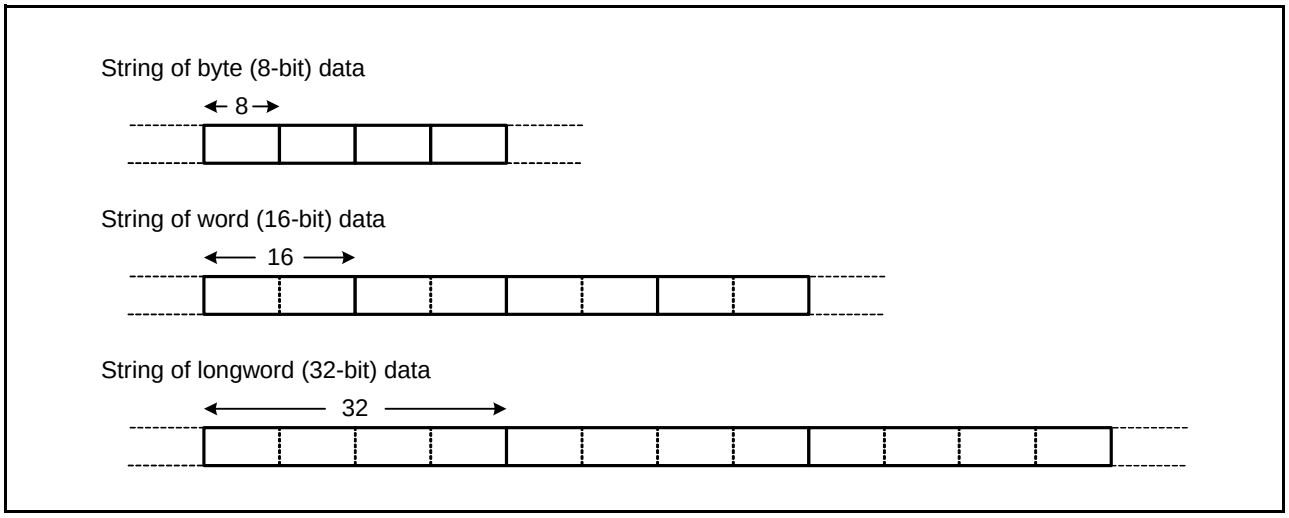


Figure 2.5 String

2.5 Endian

For the 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, this MCU 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

Address of src \ Operation	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

Address of dest \ Operation	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

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

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

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

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

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

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

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

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

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

2.5.2 Access to I/O Registers

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

2.5.3 Notes on Access to I/O Registers

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

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

2.5.4 Data Arrangement

2.5.4.1 Data Arrangement in Registers

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

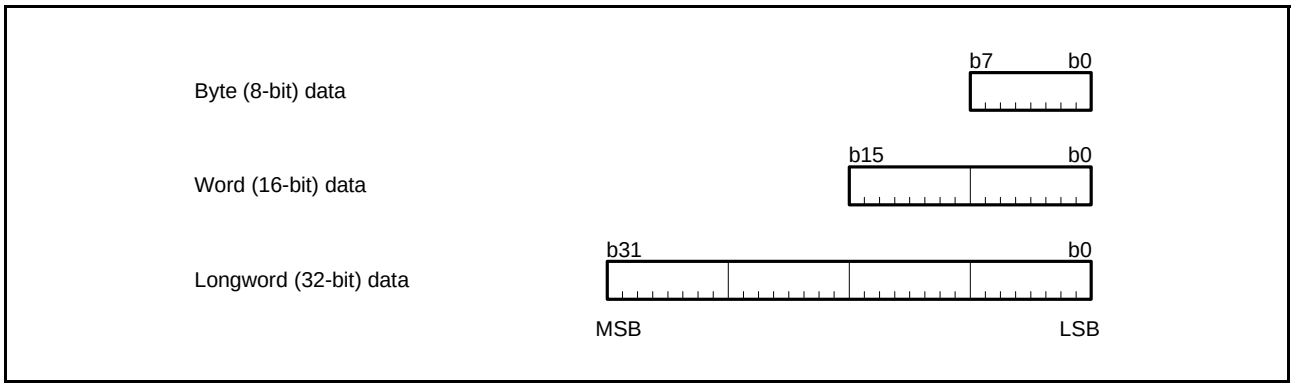


Figure 2.6 Data Arrangement in Registers

2.5.4.2 Data Arrangement in Memory

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

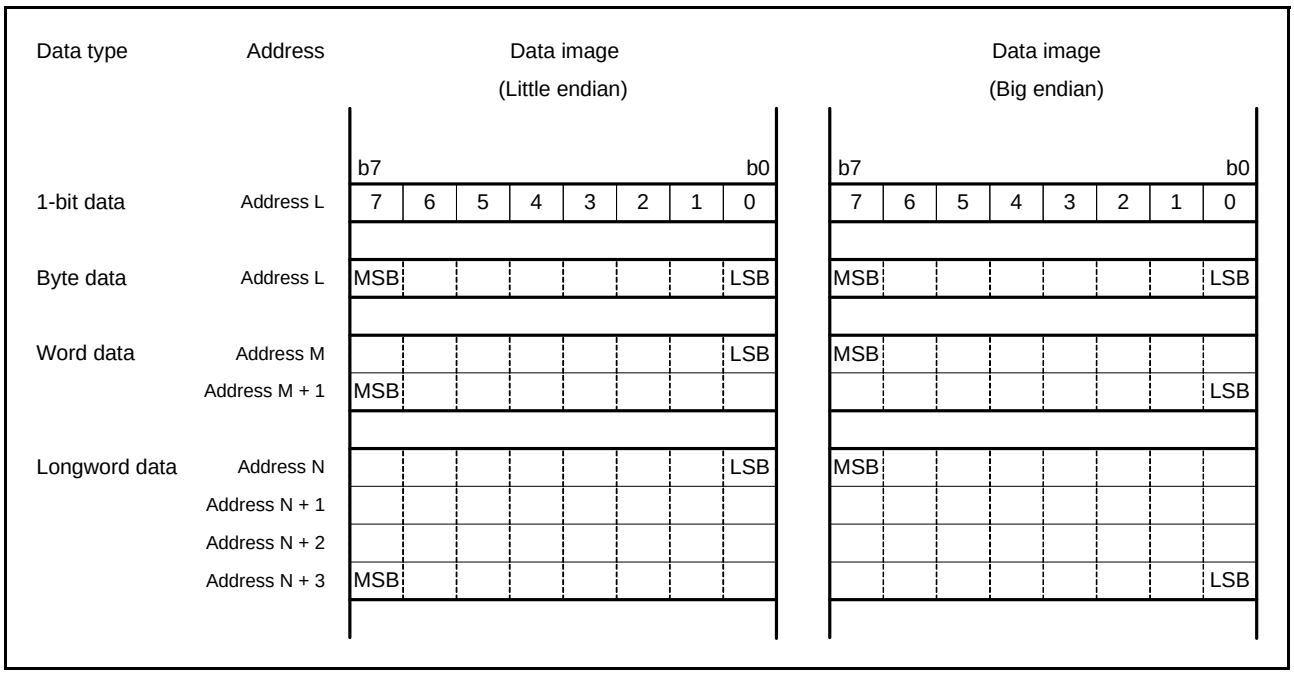


Figure 2.7 Data Arrangement in Memory

2.5.5 Notes on the Allocation of Instruction Codes

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

2.6 Vector Table

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

2.6.1 Exception Vector Table

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

Figure 2.8 shows the exception vector table.

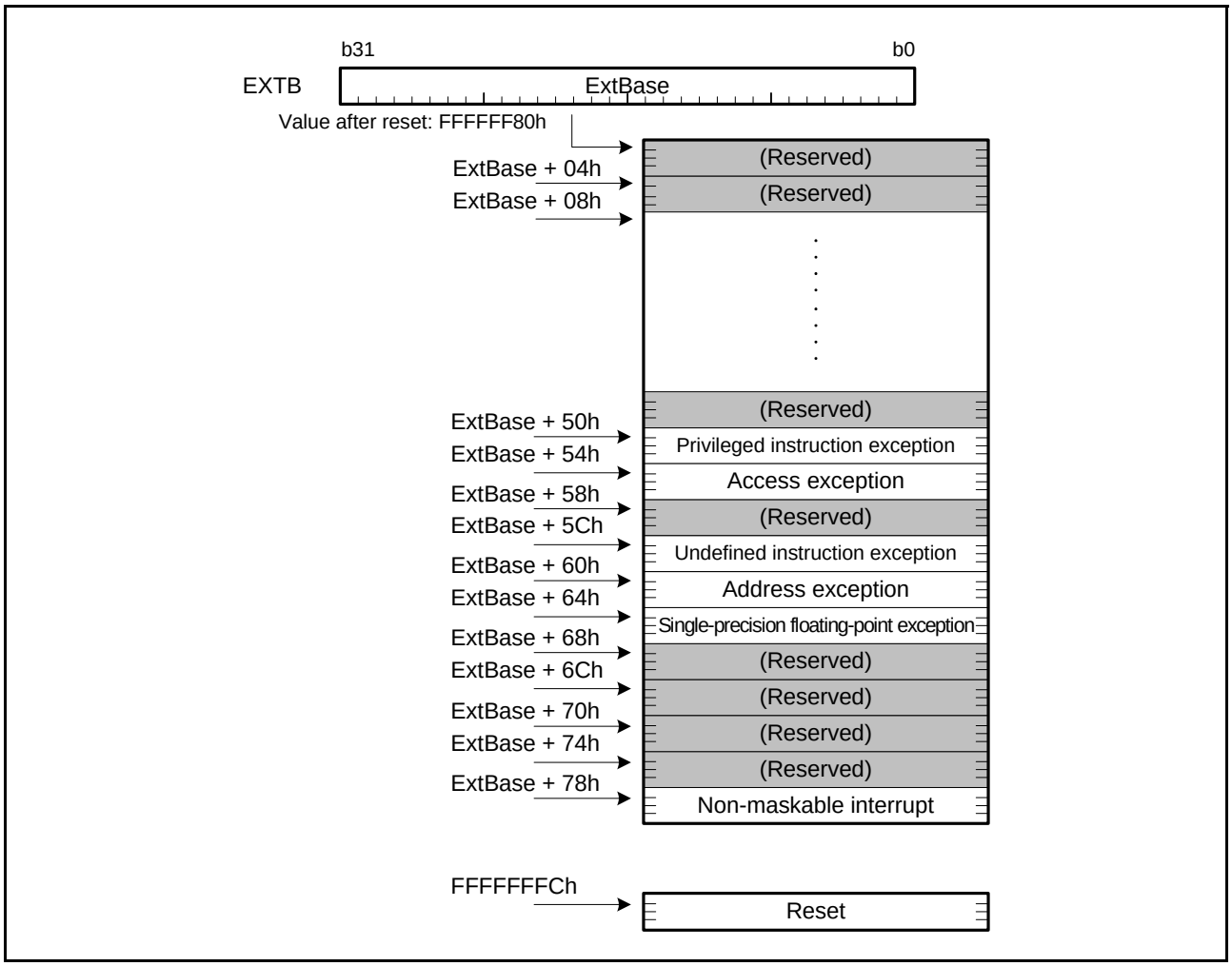


Figure 2.8 Exception Vector Table

2.6.2 Interrupt Vector Table

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

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

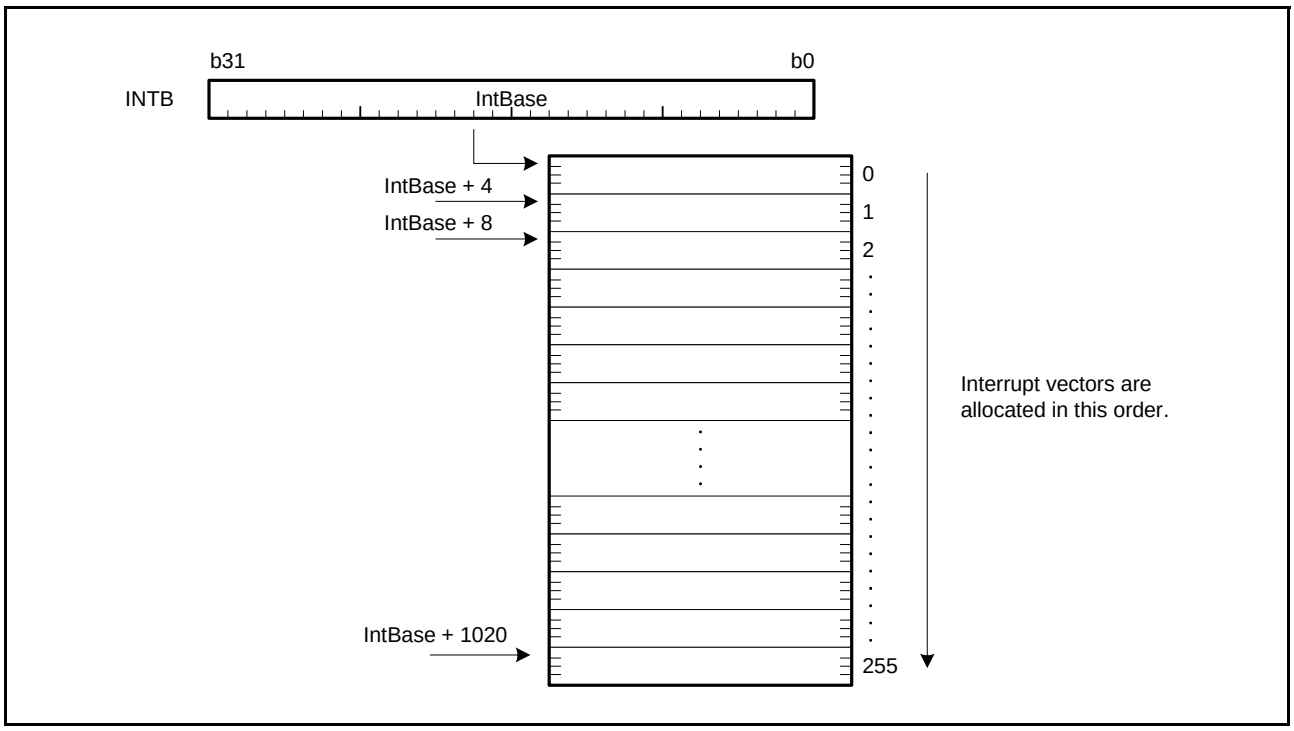


Figure 2.9 Interrupt Vector Table

2.7 Register Bank Save Function

The CPU has dedicated save register banks and functionality for using them for the fast saving and restoring of the values of CPU registers (see Figure 2.10). The save register banks enable the fast collective saving at the start of the exception handling routine and fast collective restoring of register values at the end of the exception handling routine.
The save register banks are only accessible by the SAVE and RSTR instructions, and are independent of the 4-Gbyte address space. Each of the multiple banks is used to save and restore the values of the following CPU registers: all general purpose registers except R0, the USP, FPSW, and accumulators (ACC0, ACC1). Values in the save register banks are undefined after a reset.
A unique number (bank number) is allocated to each save register bank.
The MCU has 16 save register banks, to which the bank numbers 0 to 15 are assigned.

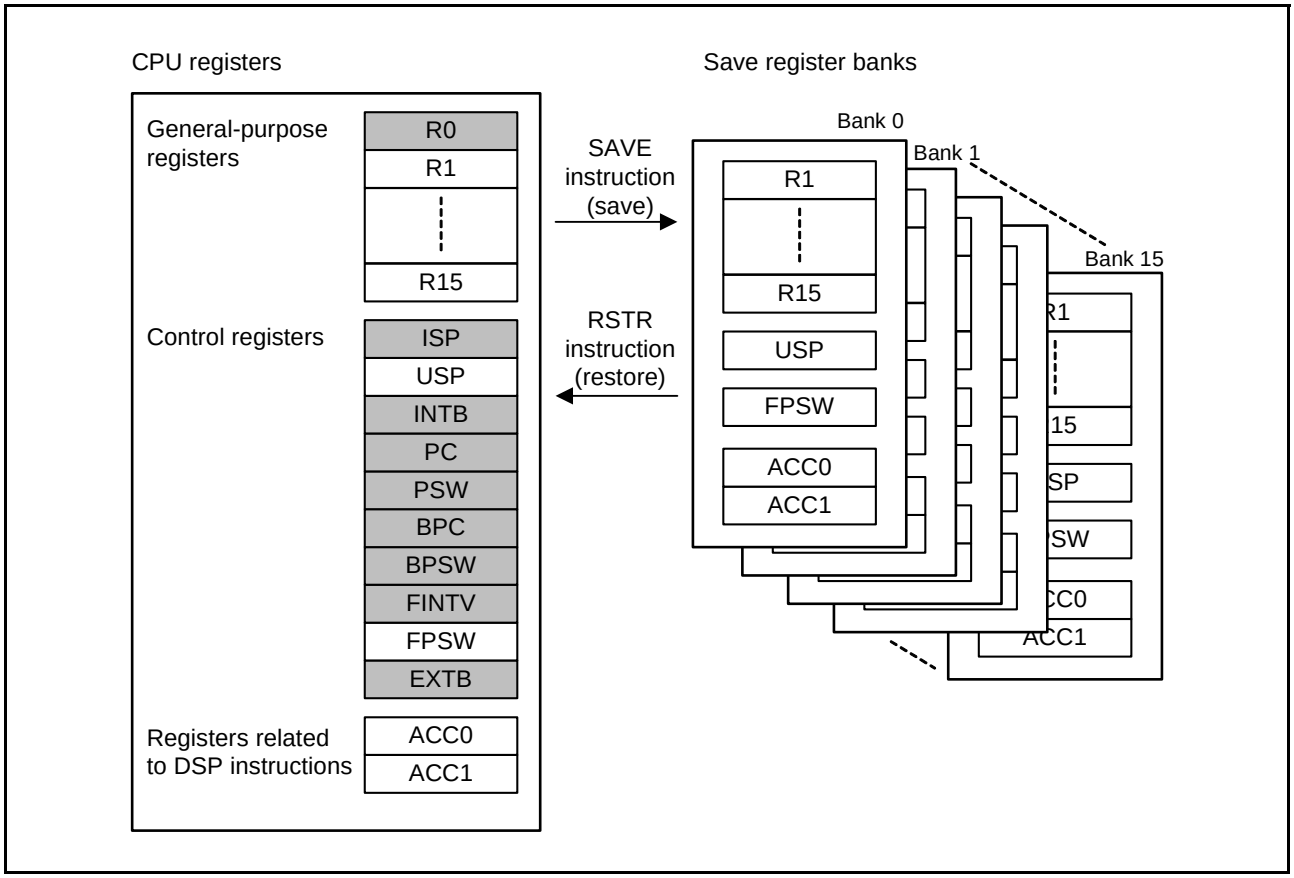


Figure 2.10 Save Register Banks

2.8 Double-Precision Floating-Point Coprocessor

The double-precision floating-point coprocessor operates as a coprocessor of the CPU, and executes double-precision floating-point processing instructions. Using the double-precision floating-point coprocessor considerably accelerates the processing of double-precision floating point arithmetic.

2.8.1 Features

- Double-precision floating-point register set
  Double-precision floating-point data registers: Sixteen 64-bit registers
  Double-precision floating-point control registers: Four 32-bit registers
- Double-precision floating-point processing instructions: 21
- Notifying the interrupt controller of double-precision floating-point exceptions

2.8.2 Double-Precision Floating-Point Register Set

The double-precision floating-point coprocessor consists of 16 double-precision floating-point data registers and 4 double-precision floating-point control registers.

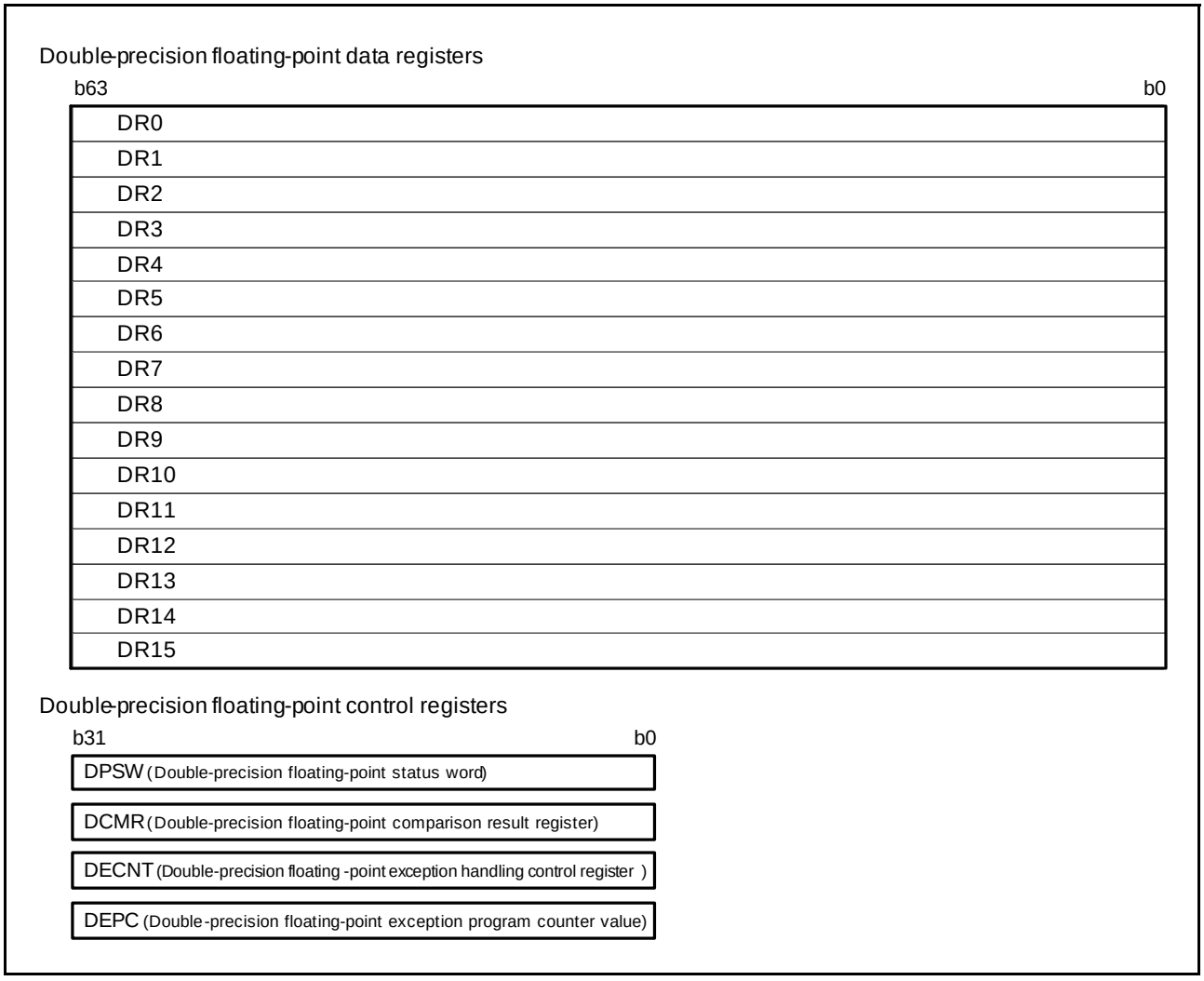


Figure 2.11 Double-Precision Floating-Point Register Set

2.8.2.1 Double-Precision Floating-Point Data Registers (DR0 to DR15)

16 double-precision floating-point data registers with 64-bit width are provided (DR0 to DR15). To specify 32-bit values, use the upper 32 bits (DRH0 to DRH15) or lower 32 bits (DRL0 to DRL15) as separate units.

2.8.2.2 Double-Precision Floating-Point Control Registers

Four double-precision floating-point control registers are provided.

- Double-precision floating-point status word (DPSW)
- Double-precision floating-point comparison result register (DCMR)
- Double-precision floating-point exception handling control register (DECNT)
- Double-precision floating-point exception program counter (DEPC)

(1) Double-Precision Floating-Point Status Word (DPSW)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DFS	DFX	DFU	DFZ	DFO	DFV	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	DEX	DEU	DEZ	DEO	DEV	—	DDN	DCE	DCX	DCU	DCZ	DCO	DCV	DRM[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	DRM[1:0]	Double-Precision Floating-Point Rounding-Mode Setting	b1 b0 0 0: Round to the nearest value 0 1: Round towards 0 1 0: Round towards +∞ 1 1: Round towards –∞	R/W
b2	DCV	Invalid Operation Cause Flag	0: No invalid operation has been encountered. 1: Invalid operation has been encountered.	R/(W)*1
b3	DCO	Overflow Cause Flag	0: No overflow has occurred. 1: Overflow has occurred.	R/(W)*1
b4	DCZ	Division-by-Zero Cause Flag	0: No division-by-zero has occurred. 1: Division-by-zero has occurred.	R/(W)*1
b5	DCU	Underflow Cause Flag	0: No underflow has occurred. 1: Underflow has occurred.	R/(W)*1
b6	DCX	Inexact Cause Flag	0: No inexact exception has been generated. 1: Inexact exception has been generated.	R/(W)*1
b7	DCE	Unimplemented Processing Cause Flag	0:No unimplemented processing has been encountered. 1:Unimplemented processing has been encountered.	R/(W)*1
b8	DDN	0 Flush Bit of Denormalized Number	0:A denormalized number is handled as a denormalized number. 1:A denormalized number is handled as 0.*2	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	DEV	Invalid Operation Exception Enable	0: Invalid operation exception is masked. 1: Invalid operation exception is enabled.	R/W
b11	DEO	Overflow Exception Enable	0: Overflow exception is masked. 1: Overflow exception is enabled.	R/W
b12	DEZ	Division-by-Zero Exception Enable	0: Division-by-zero exception is masked. 1: Division-by-zero exception is enabled.	R/W
b13	DEU	Underflow Exception Enable	0: Underflow exception is masked. 1: Underflow exception is enabled.	R/W
b14	DEX	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	DFV	Invalid Operation Flag*3	0: No invalid operation has been encountered. 1: Invalid operation has been encountered.*8	R/W
b27	DFO	Overflow Flag*4	0: No overflow has occurred. 1: Overflow has occurred.*8	R/W
b28	DFZ	Division-by-Zero Flag*5	0: No division-by-zero has occurred. 1: Division-by-zero has occurred.*8	R/W
b29	DFU	Underflow Flag*6	0: No underflow has occurred. 1: Underflow has occurred.*8	R/W
b30	DFX	Inexact Flag*7	0: No inexact exception has been generated. 1: Inexact exception has been generated.*8	R/W

Bit	Symbol	Bit Name	Description	R/W
b31	DFS	Double-Precision Floating-Point Error Summary Flag	This bit reflects the logical OR of the DFU, DFZ, DFO, and DFV flags.	R

Note 1. When 0 is written to the bit, the setting of the bit will be 0; the bit retains the previous value in response to the writing of 1.
Note 2. Positive denormalized numbers are treated as +0, negative denormalized numbers as –0.
Note 3. When the DEV bit is set to 0, the DFV flag is enabled.
Note 4. When the DEO bit is set to 0, the DFO flag is enabled.
Note 5. When the DEZ bit is set to 0, the DFZ flag is enabled.
Note 6. When the DEU bit is set to 0, the DFU flag is enabled.
Note 7. When the DEX bit is set to 0, the DFX flag is enabled.
Note 8. Once the bit has been set to 1, this value is retained until it is set to 0 by software.

The double-precision floating-point status word (DPSW) indicates the results of double-precision floating-point arithmetic operations.
When the corresponding exception handling enable bits (DEj) are set to enable processing of the exceptions (DEj = 1), the DCj flags can be used by the exception handling routine (the interrupt handling routine with the double-precision floating-point exception as its source) to identify the source of that exception. If handling of an exception is masked (DEj = 0), the DFj flag can be used to check for the generation of the exception at the end of a sequence of processing. The DFj flags operate in an accumulative fashion (j =X, U, Z, O, or V).
The single-precision floating-point status word (FPSW) is neither referred to nor updated in double-precision floating-point arithmetic operations.

DRM[1:0] Bits (Double-Precision Floating-Point Rounding-Mode Setting)

These bits specify the double-precision floating-point rounding-mode.

Explanation of Double-Precision Floating-Point Rounding Modes

- Rounding to the nearest value (the default behavior): An inexact result is rounded to the available value that is closest to the result of a hypothetical calculation with infinite precision. 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.

DCV Flag (Invalid Operation Cause Flag), DCO Flag (Overflow Cause Flag), DCZ Flag (Division-by-Zero Cause Flag), DCU Flag (Underflow Cause Flag), DCX Flag (Inexact Cause Flag), and DCE Flag (Unimplemented Processing Cause Flag)

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

- If an exception or processing that is not implemented is not encountered in the execution of a double-precision floating-point arithmetic instruction other than DABS or DNEG, the corresponding flags become 0.
- When 0 is written to the bit by the MVTDC instruction, the bit is set to 0; the bit retains the previous value when 1 is written by the instruction.

DDN Bit (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.

DEV Bit (Invalid Operation Exception Enable), DEO Bit (Overflow Exception Enable), DEZ Bit (Division-by-Zero Exception Enable), DEU Bit (Underflow Exception Enable), and DEX Bit (Inexact Exception Enable)

When any of the five floating-point exceptions specified in the IEEE754 standard is generated in the execution of a double-precision floating-point operation instructions, these bits determine whether the CPU will start handling the exception (i.e., whether an interrupt request will be sent to the interrupt controller). When the bit corresponding to an exception is 0, interrupt requests are not generated. When the bit corresponding to an exception is 1, interrupt requests are generated.

DFV Flag (Invalid Operation Flag), DFO Flag (Overflow Flag), DFZ Flag (Division-by-Zero Flag), DFU Flag (Underflow Flag), and DFX Flag (Inexact Flag)

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

DFS Flag (Double-Precision Floating-Point Error Summary Flag)

This bit reflects the logical OR of the DFU, DFZ, DFO, and DFV flags.

The value of this register is not updated when the EHM and EHS bits in the DECNT register are 1.

(2) Double-Precision Floating-Point Comparison Result Register (DCMR)

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RES
Value after 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	RES	Double-Precision Floating-Point Compare Instruction Result Flag	0: Condition for comparison was not satisfied. 1: Condition for comparison was satisfied.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

(3) Double-Precision Floating-Point Exception Handling Control Register (DECNT)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EHS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EHM
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	EHM	Double-Precision Floating-Point Exception Information Preservation Mode	0: Mode in which information on the sources of exceptions is not preserved. Information on the generation of double-precision floating-point exceptions is not preserved when they occur. 1: Mode in which information on the generation of exceptions is preserved. When a double-precision floating-point exception is generated and an interrupt request is sent to the interrupt controller, the EHS bit is changed to 1 to preserve the information that an exception has been generated.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	EHS	Double-Precision Floating-Point Exception Information Preservation Status	0: Exception information is not being preserved. 1: Exception information is being preserved. Updating of the following registers stops when the EHM and EHS bits are 1, and interrupts due to double-precision floating-point exceptions do not occur. • Double-precision floating-point status word • Double-precision floating-point exception program counter Writing 0 to this bit releases the exception information at the time the exception was generated from preservation.	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

(4) Double-Precision Floating-Point Exception Program Counter (DEPC)

	b31	b0
Value after reset:	Undefined	

The DEPC register holds the value of the program counter for the instruction that caused the most recent exception when a double-precision floating-point exception is generated and an interrupt request is sent to the interrupt controller. This register is read-only. The value of this register is not updated when the value of the EHS bit in the DECNT register is 1 while that of the EHM bit in the same register is also 1.

2.8.3 Data Types (for the Double-Precision Floating-Point Coprocessor)

The double-precision floating-point coprocessor can handle double-precision floating-point numbers.
In addition, it can handle single-precision floating-point numbers with the DTOF and FTOD instructions, and 32-bit integers with the DROUND, DTOI, DTOU, ITOD and UTOD instructions.
For single-precision floating-point numbers, refer to section 2.4.2, Single-Precision Floating-Point Numbers, and for 32-bit integers, refer to section 2.4.1, Integer.

2.8.3.1 Double-Precision Floating-Point Numbers

The double-precision floating-point number is compliant with that specified in the IEEE754 standard. Operands of this type can be used in fifteen double-precision floating-point arithmetic instructions: DABS, DADD, DCMPcm, DDIV, DMUL, DNEG, DROUND, DSUB, DSQRT, DTOF, DTOI, DTOU, FTOD, ITOD, and UTOD.

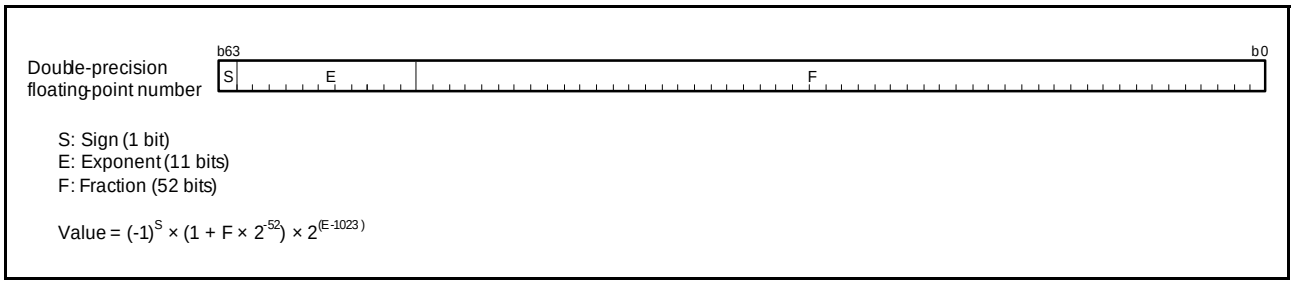


Figure 2.12 Double-Precision Floating-Point Number

The double-precision floating-point number can represent the values listed below.
0 < E < 2047 (normal numbers)
E = 0 and F = 0 (signed zero)
E = 0 and F > 0 (denormalized numbers)*1
E = 2047 and F = 0 (infinity (∞))
E = 2047 and F > 0 (Not-a-Number (NaN))
Note 1. The number is treated as 0 when the DPSW.DDN bit is 1. When the DDN bit is 0, an unimplemented processing exception is generated.

2.8.4 Data Arrangement (for the Double-Precision Floating-Point Coprocessor)

The double-precision floating-point coprocessor can handle 32-bit and 64-bit values.
For 32-bit values, refer to section 2.5.4, Data Arrangement, and section 2.5.1, Switching the Endian.

2.8.4.1 Arrangement of Data in the Double-Precision Floating-Point Registers

Figure 2.13 shows the relationship between the sizes of registers DR0 to DR15 and the bit numbers.

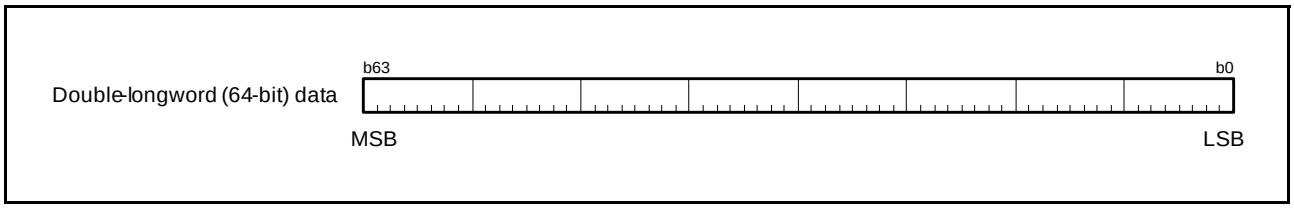


Figure 2.13 Data Arrangement in Registers

2.8.4.2 Arrangement of Data for Double-Precision Floating-Point Numbers in Memory

A double-precision floating-point number is always represented as a double-longword (64 bits) in memory. The data arrangement is selectable as little endian or big endian. Figure 2.14 shows the arrangement of data in memory.

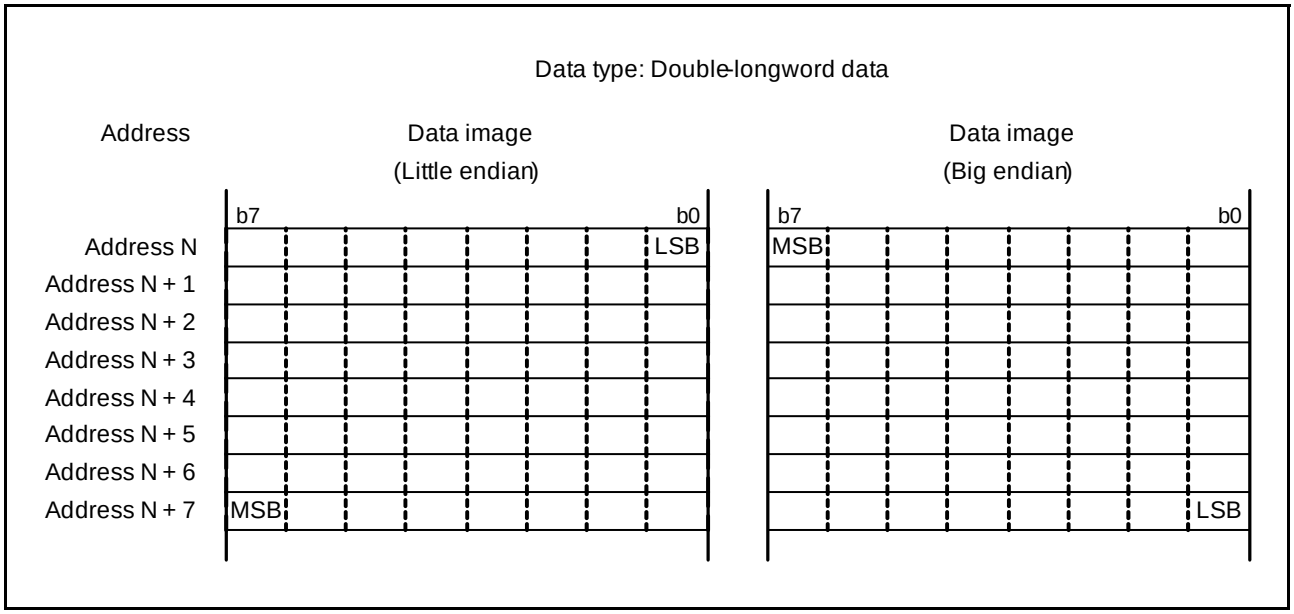


Figure 2.14 Data Arrangement in Memory

2.8.5 Switching the Endian (for the Double-Precision Floating-Point Coprocessor)

The operation in access with 64-bit values by double-precision floating-point processing instructions depends on the endian setting. Table 2.13 to Table 2.16 show operations in access for each endian and in the cases of reading and writing.

Address exceptions are generated in response to 64-bit access to addresses that are not on 32-bit boundaries.

In the tables,

LLL indicates bits D7 to D0 of the double-precision floating-point data registers,
LLH indicates bits D15 to D8 of the double-precision floating-point data registers,
LHL indicates bits D23 to D16 of the double-precision floating-point data registers,
LHH indicates bits D31 to D24 of the double-precision floating-point data registers,
HLL indicates bits D39 to D32 of the double-precision floating-point data registers,
HLH indicates bits D47 to D40 of the double-precision floating-point data registers,
HHL indicates bits D55 to D48 of the double-precision floating-point data registers, and
HHH indicates bits D63 to D56 of the double-precision floating-point data registers.

Double-precision floating-point data register: DRm	**D63 to D56**	**D55 to D48**	**D47 to D40**	**D39 to D32**
	HHH	HHL	HLH	HLL
	D31 to D24	**D23 to D16**	**D15 to D8**	**D7 to D0**
	LHH	LHL	LLH	LLL

Table 2.13 64-Bit Read Operations when Little Endian has been Selected

Operation / Address of src	**Reading a 64-bit unit from address 0**	**Reading a 64-bit unit from address 4**	**Reading a 64-bit unit from address 8**
Address 0	Transfer to LLL	—	—
Address 1	Transfer to LLH	—	—
Address 2	Transfer to LHL	—	—
Address 3	Transfer to LHH	—	—
Address 4	Transfer to HLL	Transfer to LLL	—
Address 5	Transfer to HLH	Transfer to LLH	—
Address 6	Transfer to HHL	Transfer to LHL	—
Address 7	Transfer to HHH	Transfer to LHH	—
Address 8	—	Transfer to HLL	Transfer to LLL
Address 9	—	Transfer to HLH	Transfer to LLH
Address A	—	Transfer to HHL	Transfer to LHL
Address B	—	Transfer to HHH	Transfer to LHH
Address C	—	—	Transfer to HLL
Address D	—	—	Transfer to HLH
Address E	—	—	Transfer to HHL
Address F	—	—	Transfer to HHH

Table 2.14 64-Bit Read Operations when Big Endian has been Selected

Operation / Address of src	Reading a 64-bit unit from address 0	Reading a 64-bit unit from address 4	Reading a 64-bit unit from address 8
Address 0	Transfer to HHH	—	—
Address 1	Transfer to HHL	—	—
Address 2	Transfer to HLH	—	—
Address 3	Transfer to HLL	—	—
Address 4	Transfer to LHH	Transfer to HHH	—
Address 5	Transfer to LHL	Transfer to HHL	—
Address 6	Transfer to LLH	Transfer to HLH	—
Address 7	Transfer to LLL	Transfer to HLL	—
Address 8	—	Transfer to LHH	Transfer to HHH
Address 9	—	Transfer to LHL	Transfer to HHL
Address A	—	Transfer to LLH	Transfer to HLH
Address B	—	Transfer to LLL	Transfer to HLL
Address C	—	—	Transfer to LHH
Address D	—	—	Transfer to LHL
Address E	—	—	Transfer to LLH
Address F	—	—	Transfer to LLL

Table 2.15 64-Bit Write Operations when Little Endian has been Selected

Operation / Address of dest	Writing a 64-bit unit to address 0	Writing a 64-bit unit to address 4	Writing a 64-bit unit to address 8
Address 0	Transfer to LLL	—	—
Address 1	Transfer to LLH	—	—
Address 2	Transfer to LHL	—	—
Address 3	Transfer to LHH	—	—
Address 4	Transfer to HLL	Transfer to LLL	—
Address 5	Transfer to HLH	Transfer to LLH	—
Address 6	Transfer to HHL	Transfer to LHL	—
Address 7	Transfer to HHH	Transfer to LHH	—
Address 8	—	Transfer to HLL	Transfer to LLL
Address 9	—	Transfer to HLH	Transfer to LLH
Address A	—	Transfer to HHL	Transfer to LHL
Address B	—	Transfer to HHH	Transfer to LHH
Address C	—	—	Transfer to HLL
Address D	—	—	Transfer to HLH
Address E	—	—	Transfer to HHL
Address F	—	—	Transfer to HHH

Table 2.16 64-Bit Write Operations when Big Endian has been Selected

Operation / Address of dest	Writing a 64-bit unit to address 0	Writing a 64-bit unit to address 4	Writing a 64-bit unit to address 8
Address 0	Transfer to HHH	—	—
Address 1	Transfer to HHL	—	—
Address 2	Transfer to HLH	—	—
Address 3	Transfer to HLL	—	—
Address 4	Transfer to LHH	Transfer to HHH	—
Address 5	Transfer to LHL	Transfer to HHL	—
Address 6	Transfer to LLH	Transfer to HLH	—
Address 7	Transfer to LLL	Transfer to HLL	—
Address 8	—	Transfer to LHH	Transfer to HHH
Address 9	—	Transfer to LHL	Transfer to HHL
Address A	—	Transfer to LLH	Transfer to HLH
Address B	—	Transfer to LLL	Transfer to HLL
Address C	—	—	Transfer to LHH
Address D	—	—	Transfer to LHL
Address E	—	—	Transfer to LLH
Address F	—	—	Transfer to LLL

2.9 Operation of Instructions

2.9.1 Restrictions on RMPA and String-Manipulation Instructions

2.9.1.1 Transfer Size and Data Prefetching

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

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

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

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

2.9.1.2 Access to the External Space

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

2.9.1.3 Access to I/O Registers

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

2.10 Numbers of Cycles

2.10.1 Instruction and Numbers of Cycles

Table 2.17 to Table 2.25 show the numbers of cycles in operation of each instruction. The listed numbers of cycles for access to memory are the numbers of cycles during no-wait access. The operands in the table below indicate the following meanings.

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

Table 2.17 Numbers of Cycles for Arithmetic/logic Instructions

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

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

Table 2.18 Numbers of Cycles for Transfer Instructions

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

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

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

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

Table 2.19 Numbers of Cycles for Bit Manipulation Instructions

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

Table 2.20 Numbers of Cycles for Branch Instructions

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

?: Conditional operator

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

Table 2.21 Numbers of Cycles for Single-Precision Floating-Point Operation Instructions

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

Table 2.22 Numbers of Cycles for DSP Instructions

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

Table 2.23 Numbers of Cycles for String Manipulation Instructions

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

?: Conditional operator

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

Table 2.24 Numbers of Cycles for System Manipulation Instructions

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

Table 2.25 Numbers of Cycles for Instructions for Register Bank Save Function

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Numbers of Cycles
Instructions for register bank save function	• SAVE “#IMM”/“R”	1
	• RSTR “#IMM”/“R”	3 to 6

2.10.2 Instruction and Numbers of Cycles (for the Double-Precision Floating-Point Coprocessor)

Table 2.26 shows the numbers of cycles for the double-precision floating-point related instructions. The listed numbers of cycles for access to memory are the numbers of cycles during no-wait access. The operands in the table below indicate the following meanings.

#IMM: Immediate
Rs, Rd: General-purpose register
dsp: displacement
DRs, DRs2, DRd, DRd2: Double-precision floating-point data register
DRHs, DRHd: Upper 32 bits in the double-precision floating-point data register
DRLs, DRLd: Lower 32 bits in the double-precision floating-point data register
DCRs, DCRs2, DCRd, DCRd2: Double-precision floating-point control register

Table 2.26 Numbers of Cycles for the Double-Precision Floating-Point Related Instructions

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Numbers of Cycles
Double-precision floating-point processing instructions	• {DADD, DSUB} “DRs, DRs2, DRd”	4*1
	• {DABS, DNEG} “DRs, DRd”	1
	• DCMPcm “DRs, DRs2”	4*1
	• DDIV “DRs, DRs2, DRd”	33*1
	• DMUL “DRs, DRs2, DRd”	5*1
	• DSQRT “DRs, DRd”	33*1
	• {DTOF, DTOI, DTOU, DROUND} “DRs, DRd”	4*1
	• {FTOD, ITOD, UTOD} “Rs, DRd”	4*1
Double-precision floating-point data transfer instructions	• DMOV “Rs, DRHd”/ “Rs, DRLd” / “DRHs, Rd” / “DRLs, Rd” / “DRs, DRd”/“#IMM, DRHd”/ “#IMM, DRLd”	1
	• DMOV “DRs, [Rd]”/ “DRs, dsp[Rd]”	1
	• DMOV “[Rs], DRd”/ “dsp[Rs], DRd”	Throughput: 1 Latency: 2*2
	• DPUSHM “DRs-DRs2” / “DCRs-DCRs2”	n n: Number of registers*4
	• DPOPM “DRd-DRd2” / “DCRd-DCRd2”	Throughput: n Latency: n + 1 n: Number of registers*2, *3
	• MVFDC “DCRs, Rd”	1
	• MVFDR	1
	• MVTDC “Rs, DCRd”	1

Note 1. The numbers of cycles for reference to the results of arithmetic operations by double-precision floating-point data transfer instructions and the DABS and DNEG instructions are the listed numbers of cycles plus 1.
Note 2. When the load data is used by the subsequent instruction, the numbers of cycles described as “latency” is counted as the numbers of cycles for the memory load instruction. For the cycles other than the memory load instruction, the numbers of cycles described as “throughput” is counted.
Note 3. The DPOPM instruction is converted into multiple load operations. The processing is the same as the one for the load operations of the DMOV instruction, where the operation is repeated for the number of specified registers.
Note 4. The DPUSHM instruction is converted into multiple store operations. The processing is the same as the one for the store operations of the DMOV instruction, where the operation is repeated for the number of specified registers.

2.10.3 Numbers of Cycles for Response to Interrupts

Table 2.27 lists numbers of cycles taken by processing for response to interrupts.

Table 2.27 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 Numbers 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 2.27 will be applicable when access to memory from the CPU is processed with no waiting. This MCU has a RAM that allows no-wait access and code flash memory incorporating a ROM Cache. Numbers of cycles for response to interrupts can be minimized by placing program code (and vectors) in code flash memory and the stack in RAM. Furthermore, place the addresses where the exception handling routine start on 8-byte boundaries.

For information on the numbers of cycles from notification to acceptance of the interrupt request, indicated by N in the table above, see Table 2.17 to Table 2.26.

The timing of interrupt acceptance depends on the execution state of the instruction. For more information on this, see section 14.3.1, Acceptance Timing and Saved PC Value.

2.11 Usage Note

2.11.1 Notes on Self-Diagnosis of the RAM for the Save Register Banks

The save register banks in this MCU are configured of RAM. As the save register banks are equipped with a buffer, data may be read from the buffer rather than from the memory cells of the RAM when the same address is to be read by a RSTR instruction after a write operation by the SAVE instruction. When running self-diagnosis of the RAM in the save register banks, confirm that the data have actually been written to the memory by following the procedure below so that data will not be read from the buffer.

(1) Write data to the bank targeted for diagnosis with the SAVE instruction.
(2) Write data to a bank different from the bank in the procedure (1) with the SAVE instruction.
(3) Read the data from the bank in the procedure (1) with the RSTR instruction.

3. Operating Modes

3.1 Operating Mode Types and Selection

There are two types of operating-mode selection: one is be selected by the level of pins when the reset (RES# pin reset, power-on reset, or LVD0 reset) is released, and the other is selected by software after the reset is released.
Table 3.1 shows the relationship between levels on the mode-setting pins (MD and PC7/UB) on release from the reset state and the operating mode selected at that time. For details on each of the operating modes, see section 3.3, Details of Operating Modes. Operation starts with the on-chip ROM (code flash memory and data flash memory) enabled and the external bus disabled, regardless of the mode in which operation started. Set the SYSCR0.EXBE bit to 1 (external bus enabled) to enable the external bus.

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

Mode-Setting Pin			SYSCR0 Initial State	
MD*1	**PC7/UB*2**	**Operating Mode**	**ROME**	**EXBE**
High	—	Single-chip mode	1 (On-chip ROM enabled)	0 (External bus disabled)
Low	Low	Boot mode (SCI interface)		
	High	Boot mode (USB interface)		
Low → High*3	Low	Boot mode (FINE interface)		

Note 1. Do not change the level on the MD pin while the MCU is operating.
Note 2. The PC7 pin, which is multiplexed on the same pin as the UB pin function, may also be used as a general port pin or as an input or output pin for peripheral functions.
Note 3. After release the reset state while the MD pin is at the low level, switch it to the high level within 20 to 100 msec.

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

Table 3.2 Selection of Operating Modes by Register Setting

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

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

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

Table 3.3 Selection of Endian

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

3.2 Register Descriptions

3.2.1 Mode Monitor Register (MDMONR)

Address(es): 0008 0000h

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

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

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

3.2.2 System Control Register 0 (SYSCR0)

Address(es): 0008 0006h

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

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

Note 1. Write data is not retained.

ROME Bit (On-Chip ROM Enable)

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

EXBE Bit (External Bus Enable)

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

3.2.3 System Control Register 1 (SYSCR1)

Address(es): 0008 0008h

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

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

RAME Bit (RAM Enable)

The RAME bit enables or disables the RAM.

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

Even when the RAME bit is set to 0, the RAM retains its value. However, the voltage must be maintained at no less than the specified RAM hold voltage (VRAM), which is stipulated in section 56, Electrical Characteristics.

SBYRAME Bit (Standby RAM Enable)

The SBYRAME bit enables or disables the standby RAM.

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

Even when the SBYRAME bit is set to 0, the standby RAM retains its value. However, the voltage must be maintained at no less than the specified RAM hold voltage (VRAM), which is stipulated in section 56, Electrical Characteristics.

3.3 Details of Operating Modes

3.3.1 Single-Chip Mode

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

3.3.2 On-Chip ROM Enabled Extended Mode

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

3.3.3 On-Chip ROM Disabled Extended Mode

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

3.3.4 Boot Mode (SCI Interface)

In this mode, the flash memory rewriting program (boot program) stored in a dedicated area within the MCU operates. The on-chip ROM (code flash memory and data flash memory) can be programmed or erased from outside the MCU by using a universal asynchronous receiver/transmitter (SCI1). For details, see section 55, Flash Memory (FLASH).
The chip starts up in boot mode (SCI interface) when both the MD and UB pins are set to the low level at the time of release from the reset state.

3.3.5 Boot Mode (USB Interface)

In this mode, the flash memory rewriting program (boot program) stored in a dedicated area within the MCU operates. The on-chip ROM (code flash memory and data flash memory) can be programmed or erased from outside the MCU by using the USB. For details, see section 55, Flash Memory (FLASH).
The chip starts up in boot mode (USB interface) when the MD pin is set to the low level and the UB pin is set to the high

level at the time of release from the reset state.

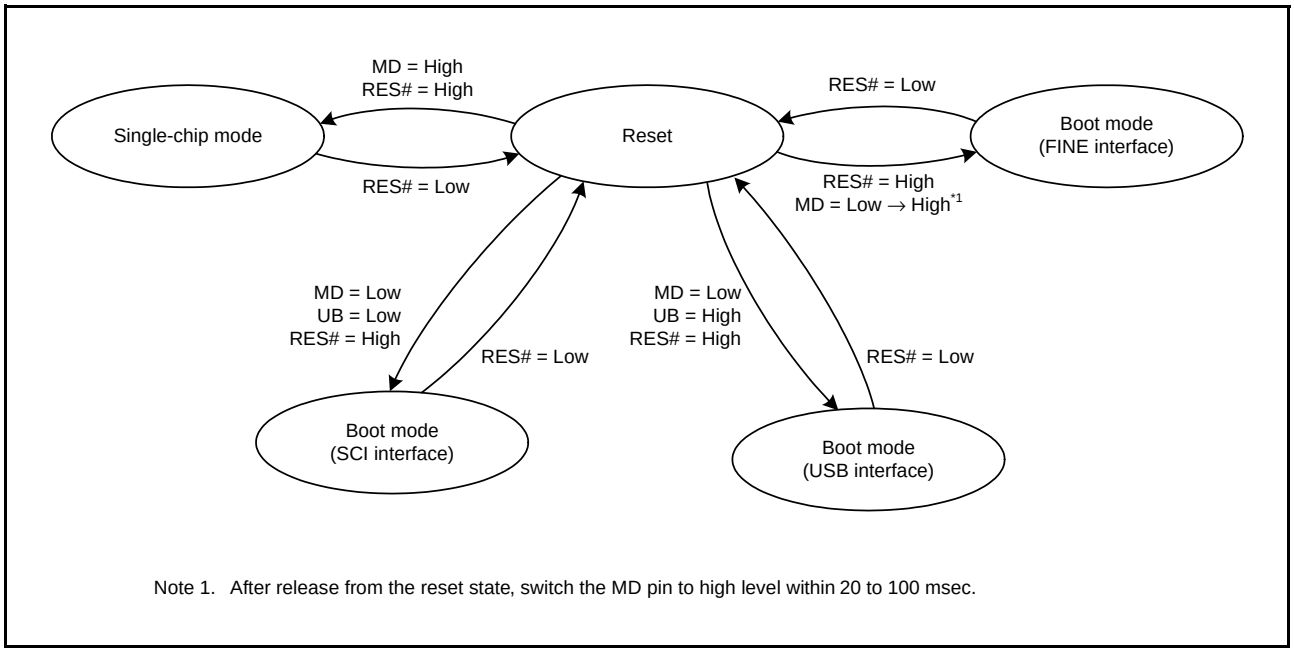
3.3.6 Boot Mode (FINE Interface)

In this mode, the flash memory rewriting program (boot program) stored in a dedicated area within the MCU operates. The on-chip ROM (code flash memory and data flash memory) can be programmed or erased from outside the MCU by using the FINE. For details, see section 55, Flash Memory (FLASH).
The chip starts up in boot mode (FINE interface) when the MD pin is set to the low level at the time of release from the reset state and then is switched to the high level within 20 to 100 msec.

3.4 Transitions of Operating Modes

3.4.1 Operating Mode Transitions Determined by the Mode-Setting Pins

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



Note 1. After release from the reset state, switch the MD pin to high level within 20 to 100 msec.

Figure 3.1 Mode-Setting Pin Level and Operating Mode

3.4.2 Operating Mode Transitions According to Register Setting

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

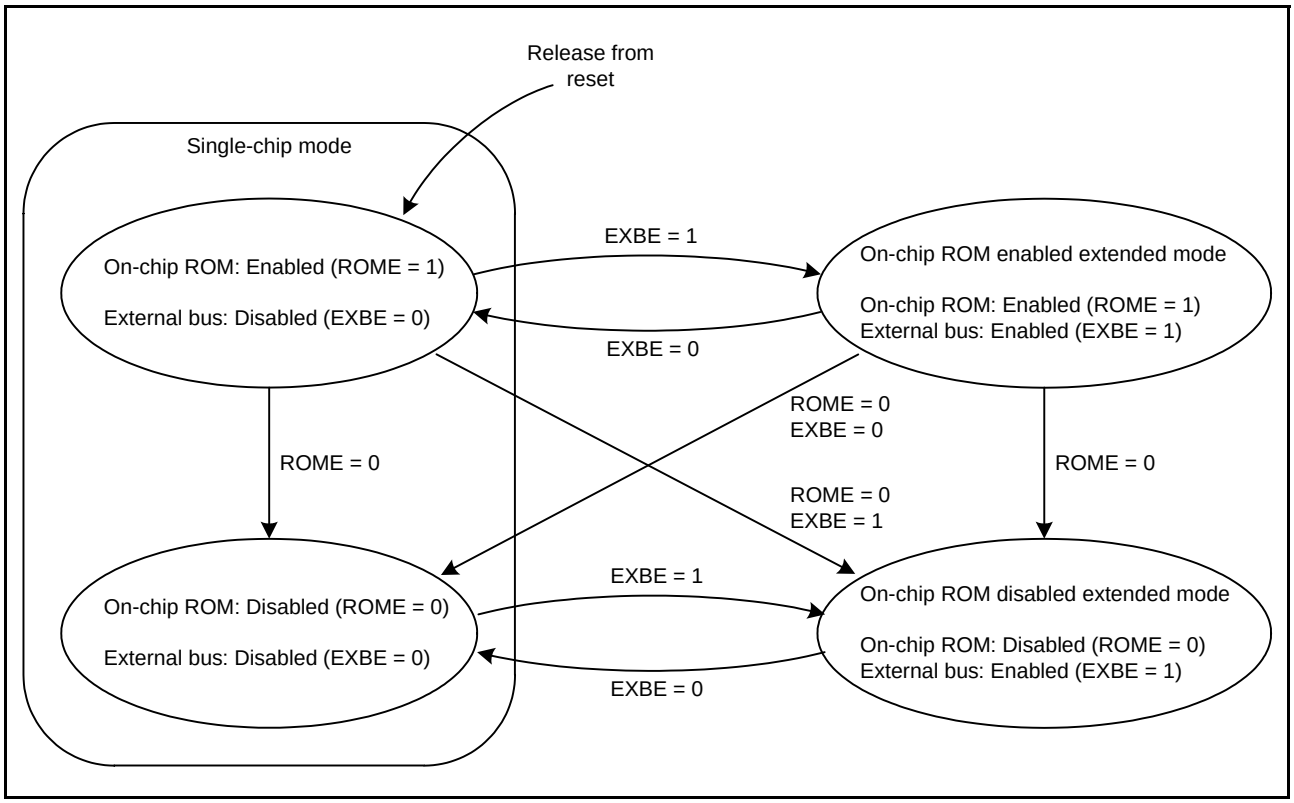
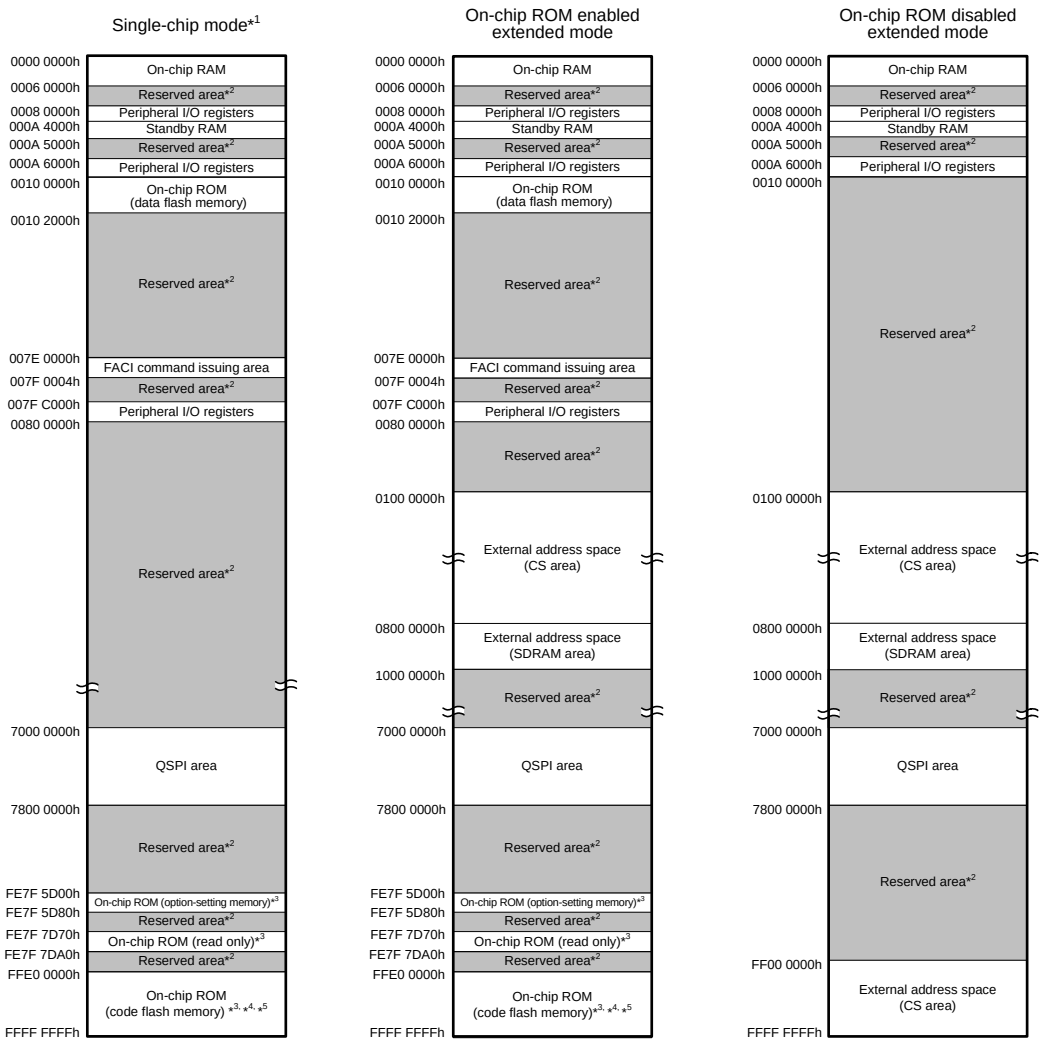


Figure 3.2 Setting of SYSCR0.ROME and EXBE Bits and Operating Modes

4. Address Space

4.1 Address Space

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



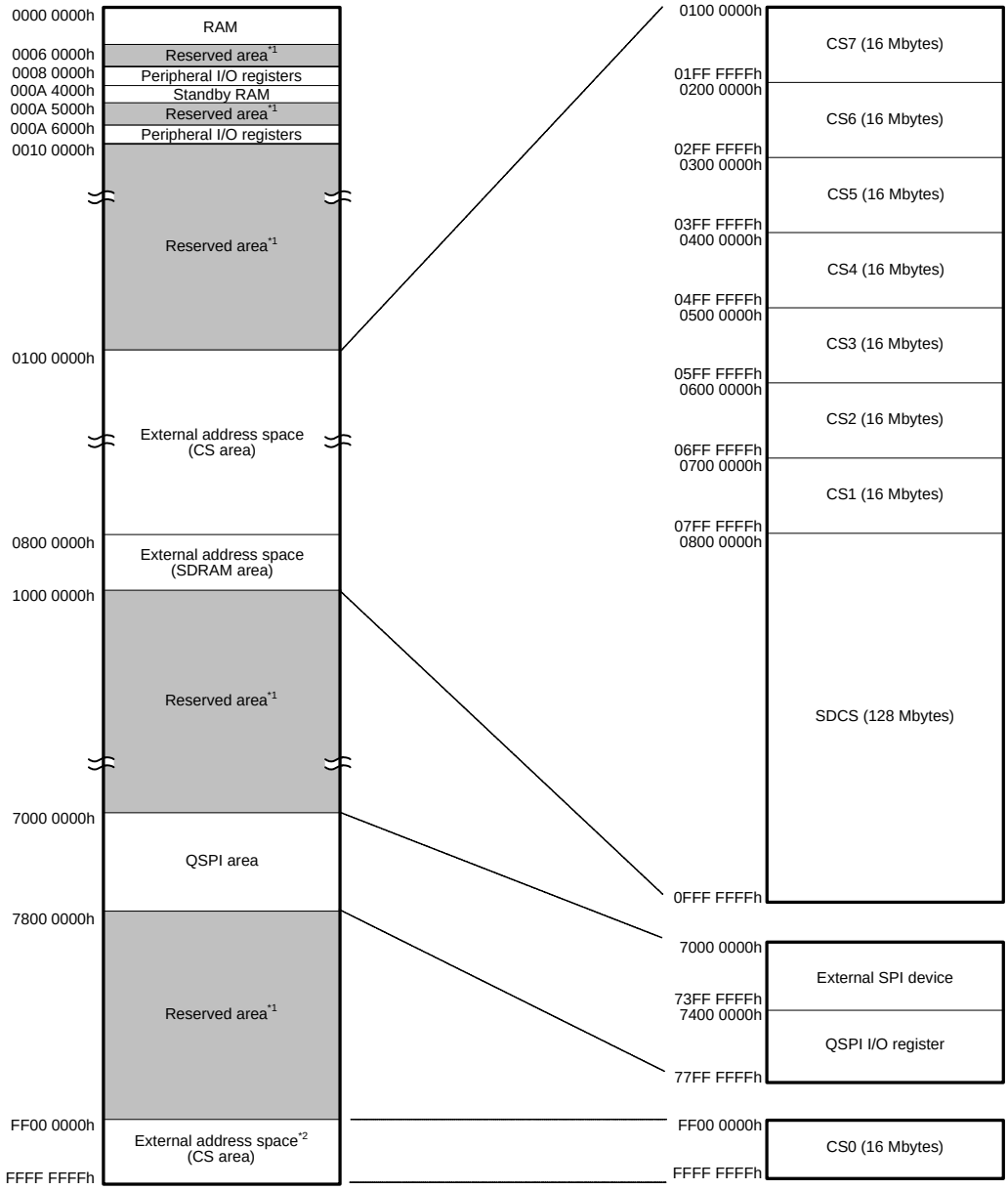
Note 1. The address space in boot mode is the same as the address space in single-chip mode.
Note 2. Reserved areas should not be accessed.
Note 3. The access cycle is 1 cycle and 2 cycles while the ROMWT[1:0] bits are 00b and 01b respectively.
Note 4. The on-chip ROM (code flash memory) can be used in linear mode, where the user area forms a single area, or in dual mode, where the user area is divided into two banks. For details, refer to section 55.3, Structure of Memory in section 55, Flash Memory (FLASH).
Note 5. The capacity of the code flash memory differs depending on the products.

Code Flash Memory			**Data Flash Memory**		**RAM**	
Capacity	**Address**		**Capacity**	**Address**	**Capacity**	**Address**
	Linear mode	**Dual mode (BANKSEL.BANKSWP[2:0] = 111b)**				
2 Mbytes	FFE0 0000h to FFFF FFFFh	bank 1: FFE0 0000h to FFEF FFFFh bank 0: FFF0 0000h to FFFF FFFFh	8 Kbytes	0010 0000h to 0010 1FFFh	384 Kbytes	0000 0000h to 0005 FFFFh
1.5 Mbytes	FFE8 0000h to FFFF FFFFh	bank 1: FFE4 0000h to FFEF FFFFh bank 0: FFF4 0000h to FFFF FFFFh	8 Kbytes	0010 0000h to 0010 1FFFh	384 Kbytes	0000 0000h to 0005 FFFFh
1 Mbyte	FFF0 0000h to FFFF FFFFh	bank 1: FFE8 0000h to FFEF FFFFh bank 0: FFF8 0000h to FFFF FFFFh	8 Kbytes	0010 0000h to 0010 1FFFh	384 Kbytes	0000 0000h to 0005 FFFFh

Figure 4.1 Memory Map in Each Operating Mode

4.2 External Address Space and QSPI Area

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



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

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

5. I/O Registers

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

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

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

(2) Notes on writing to I/O registers

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

[Examples of cases requiring special care]

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

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

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

[Instruction examples]

- Byte-size I/O registers

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

- Word-size I/O registers

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

- Longword-size I/O registers

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

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

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

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

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

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

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

(4) Notes on Sleep Mode and Mode Transitions

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

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

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

5.1 I/O Register Addresses (Address Order)

Table 5.1 List of I/O Registers (Address Order) (1 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 0000h	SYSTEM	Mode Monitor Register	MDMONR	16	16	3 ICLK		section 3
0008 0006h	SYSTEM	System Control Register 0	SYSCR0	16	16	3 ICLK		section 3
0008 0008h	SYSTEM	System Control Register 1	SYSCR1	16	16	3 ICLK		section 3
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		section 11
0008 0014h	SYSTEM	Module Stop Control Register B	MSTPCRB	32	32	3 ICLK		section 11
0008 0018h	SYSTEM	Module Stop Control Register C	MSTPCRC	32	32	3 ICLK		section 11
0008 001Ch	SYSTEM	Module Stop Control Register D	MSTPCRD	32	32	3 ICLK		section 11
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		section 9
0008 0026h	SYSTEM	System Clock Control Register 3	SCKCR3	16	16	3 ICLK		section 9
0008 0028h	SYSTEM	PLL Control Register	PLLCR	16	16	3 ICLK		section 9
0008 002Ah	SYSTEM	PLL Control Register 2	PLLCR2	8	8	3 ICLK		section 9
0008 0030h	SYSTEM	External Bus Clock Control Register	BCKCR	8	8	3 ICLK		section 9
0008 0032h	SYSTEM	Main Clock Oscillator Control Register	MOSCCR	8	8	3 ICLK		section 9
0008 0033h	SYSTEM	Sub-Clock Oscillator Control Register	SOSCCR	8	8	3 ICLK		section 9
0008 0034h	SYSTEM	Low-Speed On-Chip Oscillator Control Register	LOCOCR	8	8	3 ICLK		section 9
0008 0035h	SYSTEM	IWDT-Dedicated On-Chip Oscillator Control Register	ILOCOCR	8	8	3 ICLK		section 9
0008 0036h	SYSTEM	High-Speed On-Chip Oscillator Control Register	HOCOCR	8	8	3 ICLK		section 9
0008 0037h	SYSTEM	High-Speed On-Chip Oscillator Control Register 2	HOCOCR2	8	8	3 ICLK		section 9
0008 0039h	SYSTEM	FLL Control Register 1	FLLCR1	8	8	3 ICLK		section 9
0008 003Ah	SYSTEM	FLL Control Register 2	FLLCR2	16	16	3 ICLK		section 9
0008 003Ch	SYSTEM	Oscillation Stabilization Flag Register	OSCOVFSR	8	8	3 ICLK		section 9
0008 003Eh	SYSTEM	CLKOUT Output Control Register	CKOCR	16	16	3 ICLK		section 9
0008 0040h	SYSTEM	Oscillation Stop Detection Control Register	OSTDCR	8	8	3 ICLK		section 9
0008 0041h	SYSTEM	Oscillation Stop Detection Status Register	OSTDSR	8	8	3 ICLK		section 9
0008 00A0h	SYSTEM	Operating Power Control Register	OPCCR	8	8	3 ICLK		section 11
0008 00A1h	SYSTEM	Sleep Mode Return 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		section 9
0008 00A3h	SYSTEM	Sub-Clock Oscillator Wait Control Register	SOSCWTCR	8	8	3 ICLK		section 9
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		section 6
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		section 8
0008 00E2h	SYSTEM	Voltage Monitoring 2 Circuit Control Register 1	LVD2CR1	8	8	3 ICLK		section 8
0008 00E3h	SYSTEM	Voltage Monitoring 2 Circuit Status Register	LVD2SR	8	8	3 ICLK		section 8
0008 03FEh	SYSTEM	Protect Register	PRCR	16	16	3 ICLK		section 13
0008 1000h	FLASH	ROM Cache Enable Register	ROMCE	16	16	2 ICLK		section 55
0008 1004h	FLASH	ROM Cache Invalidate Register	ROMCIV	16	16	2 ICLK		section 55
0008 101Ch	SYSTEM	ROM Wait Cycle Setting Register	ROMWT	8	8	2 ICLK		section 9
0008 1040h	FLASH	Non-Cacheable Area 0 Address Register	NCRG0	32	32	2 ICLK		section 55
0008 1044h	FLASH	Non-Cacheable Area 0 Setting Register	NCRC0	32	32	2 ICLK		section 55
0008 1048h	FLASH	Non-Cacheable Area 1 Address Register	NCRG1	32	32	2 ICLK		section 55
0008 104Ch	FLASH	Non-Cacheable Area 1 Setting Register	NCRC1	32	32	2 ICLK		section 55
0008 1200h	RAM	RAM Operating Mode Control Register	RAMMODE	8	8	2 ICLK		section 53
0008 1201h	RAM	RAM Error Status Register	RAMSTS	8	8	2 ICLK		section 53
0008 1204h	RAM	RAM Protection Register	RAMPRCR	8	8	2 ICLK		section 53

Table 5.1 List of I/O Registers (Address Order) (2 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 1208h	RAM	RAM Error Address Capture Register	RAMECAD	32	32	2 ICLK		section 53
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		section 16
0008 1308h	BSC	Bus Error Status Register 1	BERSR1	8	8	2 ICLK		section 16
0008 130Ah	BSC	Bus Error Status Register 2	BERSR2	16	16	2 ICLK		section 16
0008 1310h	BSC	Bus Priority Control Register	BUSPRI	16	16	2 ICLK		section 16
0008 2000h	DMAC0	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 18
0008 2004h	DMAC0	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 18
0008 2008h	DMAC0	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 18
0008 200Ch	DMAC0	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 18
0008 2010h	DMAC0	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 18
0008 2013h	DMAC0	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 18
0008 2014h	DMAC0	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 18
0008 2018h	DMAC0	DMA Offset Register	DMOFR	32	32	2 ICLK		section 18
0008 201Ch	DMAC0	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 18
0008 201Dh	DMAC0	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 18
0008 201Eh	DMAC0	DMA Status Register	DMSTS	8	8	2 ICLK		section 18
0008 201Fh	DMAC0	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		section 18
0008 2040h	DMAC1	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 18
0008 2044h	DMAC1	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 18
0008 2048h	DMAC1	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 18
0008 204Ch	DMAC1	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 18
0008 2050h	DMAC1	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 18
0008 2053h	DMAC1	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 18
0008 2054h	DMAC1	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 18
0008 205Ch	DMAC1	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 18
0008 205Dh	DMAC1	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 18
0008 205Eh	DMAC1	DMA Status Register	DMSTS	8	8	2 ICLK		section 18
0008 205Fh	DMAC1	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		section 18
0008 2080h	DMAC2	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 18
0008 2084h	DMAC2	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 18
0008 2088h	DMAC2	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 18
0008 208Ch	DMAC2	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 18
0008 2090h	DMAC2	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 18
0008 2093h	DMAC2	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 18
0008 2094h	DMAC2	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 18
0008 209Ch	DMAC2	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 18
0008 209Dh	DMAC2	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 18
0008 209Eh	DMAC2	DMA Status Register	DMSTS	8	8	2 ICLK		section 18
0008 209Fh	DMAC2	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		section 18
0008 20C0h	DMAC3	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 18
0008 20C4h	DMAC3	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 18
0008 20C8h	DMAC3	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 18
0008 20CCh	DMAC3	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 18
0008 20D0h	DMAC3	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 18
0008 20D3h	DMAC3	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 18
0008 20D4h	DMAC3	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 18
0008 20DCh	DMAC3	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 18
0008 20DDh	DMAC3	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 18
0008 20DEh	DMAC3	DMA Status Register	DMSTS	8	8	2 ICLK		section 18

Table 5.1 List of I/O Registers (Address Order) (3 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 20DFh	DMAC3	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		section 18
0008 2100h	DMAC4	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 18
0008 2104h	DMAC4	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 18
0008 2108h	DMAC4	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 18
0008 210Ch	DMAC4	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 18
0008 2110h	DMAC4	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 18
0008 2113h	DMAC4	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 18
0008 2114h	DMAC4	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 18
0008 211Ch	DMAC4	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 18
0008 211Dh	DMAC4	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 18
0008 211Eh	DMAC4	DMA Status Register	DMSTS	8	8	2 ICLK		section 18
0008 211Fh	DMAC4	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		section 18
0008 2140h	DMAC5	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 18
0008 2144h	DMAC5	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 18
0008 2148h	DMAC5	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 18
0008 214Ch	DMAC5	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 18
0008 2150h	DMAC5	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 18
0008 2153h	DMAC5	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 18
0008 2154h	DMAC5	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 18
0008 215Ch	DMAC5	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 18
0008 215Dh	DMAC5	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 18
0008 215Eh	DMAC5	DMA Status Register	DMSTS	8	8	2 ICLK		section 18
0008 215Fh	DMAC5	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		section 18
0008 2180h	DMAC6	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 18
0008 2184h	DMAC6	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 18
0008 2188h	DMAC6	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 18
0008 218Ch	DMAC6	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 18
0008 2190h	DMAC6	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 18
0008 2193h	DMAC6	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 18
0008 2194h	DMAC6	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 18
0008 219Ch	DMAC6	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 18
0008 219Dh	DMAC6	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 18
0008 219Eh	DMAC6	DMA Status Register	DMSTS	8	8	2 ICLK		section 18
0008 219Fh	DMAC6	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		section 18
0008 21C0h	DMAC7	DMA Source Address Register	DMSAR	32	32	2 ICLK		section 18
0008 21C4h	DMAC7	DMA Destination Address Register	DMDAR	32	32	2 ICLK		section 18
0008 21C8h	DMAC7	DMA Transfer Count Register	DMCRA	32	32	2 ICLK		section 18
0008 21CCh	DMAC7	DMA Block Transfer Count Register	DMCRB	16	16	2 ICLK		section 18
0008 21D0h	DMAC7	DMA Transfer Mode Register	DMTMD	16	16	2 ICLK		section 18
0008 21D3h	DMAC7	DMA Interrupt Setting Register	DMINT	8	8	2 ICLK		section 18
0008 21D4h	DMAC7	DMA Address Mode Register	DMAMD	16	16	2 ICLK		section 18
0008 21DCh	DMAC7	DMA Transfer Enable Register	DMCNT	8	8	2 ICLK		section 18
0008 21DDh	DMAC7	DMA Software Start Register	DMREQ	8	8	2 ICLK		section 18
0008 21DEh	DMAC7	DMA Status Register	DMSTS	8	8	2 ICLK		section 18
0008 21DFh	DMAC7	DMA Request Source Flag Control Register	DMCSL	8	8	2 ICLK		section 18
0008 2200h	DMAC	DMAC Module Start Register	DMAST	8	8	2 ICLK		section 18
0008 2204h	DMAC	DMAC74 Interrupt Status Monitor Register	DMIST	8	8	2 ICLK		section 18
0008 2400h	DTC	DTC Control Register	DTCCR	8	8	2 ICLK		section 20
0008 2404h	DTC	DTC Vector Base Register	DTCVBR	32	32	2 ICLK		section 20
0008 2408h	DTC	DTC Address Mode Register	DTCADMOD	8	8	2 ICLK		section 20

Table 5.1 List of I/O Registers (Address Order) (4 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 240Ch	DTC	DTC Module Start Register	DTCST	8	8	2 ICLK		section 20
0008 240Eh	DTC	DTC Status Register	DTCSTS	16	16	2 ICLK		section 20
0008 2410h	DTC	DTC Index Table Base Register	DTCIBR	32	32	2 ICLK		section 20
0008 2414h	DTC	DTC Operation Register	DTCOR	8	8	2 ICLK		section 20
0008 2416h	DTC	DTC Sequence Transfer Enable Register	DTCSQE	16	16	2 ICLK		section 20
0008 2418h	DTC	DTC Address Displacement Register	DTCDISP	32	32	2 ICLK		section 20
0008 2800h	EXDMAC0	EXDMA Source Address Register	EDMSAR	32	32	1, 2 BCLK		section 19
0008 2804h	EXDMAC0	EXDMA Destination Address Register	EDMDAR	32	32	1, 2 BCLK		section 19
0008 2808h	EXDMAC0	EXDMA Transfer Count Register	EDMCRA	32	32	1, 2 BCLK		section 19
0008 280Ch	EXDMAC0	EXDMA Block Transfer Count Register	EDMCRB	16	16	1, 2 BCLK		section 19
0008 2810h	EXDMAC0	EXDMA Transfer Mode Register	EDMTMD	16	16	1, 2 BCLK		section 19
0008 2812h	EXDMAC0	EXDMA Output Setting Register	EDMOMD	8	8	1, 2 BCLK		section 19
0008 2813h	EXDMAC0	EXDMA Interrupt Setting Register	EDMINT	8	8	1, 2 BCLK		section 19
0008 2814h	EXDMAC0	EXDMA Address Mode Register	EDMAMD	32	32	1, 2 BCLK		section 19
0008 2818h	EXDMAC0	EXDMA Offset Register	EDMOFR	32	32	1, 2 BCLK		section 19
0008 281Ch	EXDMAC0	EXDMA Transfer Enable Register	EDMCNT	8	8	1, 2 BCLK		section 19
0008 281Dh	EXDMAC0	EXDMA Software Start Register	EDMREQ	8	8	1, 2 BCLK		section 19
0008 281Eh	EXDMAC0	EXDMA Status Register	EDMSTS	8	8	1, 2 BCLK		section 19
0008 2820h	EXDMAC0	EXDMA External Request Sense Mode Register	EDMRMD	8	8	1, 2 BCLK		section 19
0008 2821h	EXDMAC0	EXDMA External Request Flag Register	EDMERF	8	8	1, 2 BCLK		section 19
0008 2822h	EXDMAC0	EXDMA Peripheral Request Flag Register	EDMPRF	8	8	1, 2 BCLK		section 19
0008 2840h	EXDMAC1	EXDMA Source Address Register	EDMSAR	32	32	1, 2 BCLK		section 19
0008 2844h	EXDMAC1	EXDMA Destination Address Register	EDMDAR	32	32	1, 2 BCLK		section 19
0008 2848h	EXDMAC1	EXDMA Transfer Count Register	EDMCRA	32	32	1, 2 BCLK		section 19
0008 284Ch	EXDMAC1	EXDMA Block Transfer Count Register	EDMCRB	16	16	1, 2 BCLK		section 19
0008 2850h	EXDMAC1	EXDMA Transfer Mode Register	EDMTMD	16	16	1, 2 BCLK		section 19
0008 2852h	EXDMAC1	EXDMA Output Setting Register	EDMOMD	8	8	1, 2 BCLK		section 19
0008 2853h	EXDMAC1	EXDMA Interrupt Setting Register	EDMINT	8	8	1, 2 BCLK		section 19
0008 2854h	EXDMAC1	EXDMA Address Mode Register	EDMAMD	32	32	1, 2 BCLK		section 19
0008 285Ch	EXDMAC1	EXDMA Transfer Enable Register	EDMCNT	8	8	1, 2 BCLK		section 19
0008 285Dh	EXDMAC1	EXDMA Software Start Register	EDMREQ	8	8	1, 2 BCLK		section 19
0008 285Eh	EXDMAC1	EXDMA Status Register	EDMSTS	8	8	1, 2 BCLK		section 19
0008 2860h	EXDMAC1	EXDMA External Request Sense Mode Register	EDMRMD	8	8	1, 2 BCLK		section 19
0008 2861h	EXDMAC1	EXDMA External Request Flag Register	EDMERF	8	8	1, 2 BCLK		section 19
0008 2862h	EXDMAC1	EXDMA Peripheral Request Flag Register	EDMPRF	8	8	1, 2 BCLK		section 19
0008 2A00h	EXDMAC	EXDMAC Module Start Register	EDMAST	8	8	1, 2 BCLK		section 19
0008 2BE0h	EXDMAC	Cluster Buffer Register 0	CLSBR0	32	32	1, 2 BCLK		section 19
0008 2BE4h	EXDMAC	Cluster Buffer Register 1	CLSBR1	32	32	1, 2 BCLK		section 19
0008 2BE8h	EXDMAC	Cluster Buffer Register 2	CLSBR2	32	32	1, 2 BCLK		section 19
0008 2BECh	EXDMAC	Cluster Buffer Register 3	CLSBR3	32	32	1, 2 BCLK		section 19
0008 2BF0h	EXDMAC	Cluster Buffer Register 4	CLSBR4	32	32	1, 2 BCLK		section 19
0008 2BF4h	EXDMAC	Cluster Buffer Register 5	CLSBR5	32	32	1, 2 BCLK		section 19
0008 2BF8h	EXDMAC	Cluster Buffer Register 6	CLSBR6	32	32	1, 2 BCLK		section 19
0008 2BFCh	EXDMAC	Cluster Buffer Register 7	CLSBR7	32	32	1, 2 BCLK		section 19
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		section 16
0008 3008h	BSC	CS0 Wait Control Register 2	CS0WCR2	32	32	1, 2 BCLK		section 16
0008 3012h	BSC	CS1 Mode Register	CS1MOD	16	16	1, 2 BCLK		section 16
0008 3014h	BSC	CS1 Wait Control Register 1	CS1WCR1	32	32	1, 2 BCLK		section 16
0008 3018h	BSC	CS1 Wait Control Register 2	CS1WCR2	32	32	1, 2 BCLK		section 16

Table 5.1 List of I/O Registers (Address Order) (5 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 3022h	BSC	CS2 Mode Register	CS2MOD	16	16	1, 2 BCLK		section 16
0008 3024h	BSC	CS2 Wait Control Register 1	CS2WCR1	32	32	1, 2 BCLK		section 16
0008 3028h	BSC	CS2 Wait Control Register 2	CS2WCR2	32	32	1, 2 BCLK		section 16
0008 3032h	BSC	CS3 Mode Register	CS3MOD	16	16	1, 2 BCLK		section 16
0008 3034h	BSC	CS3 Wait Control Register 1	CS3WCR1	32	32	1, 2 BCLK		section 16
0008 3038h	BSC	CS3 Wait Control Register 2	CS3WCR2	32	32	1, 2 BCLK		section 16
0008 3042h	BSC	CS4 Mode Register	CS4MOD	16	16	1, 2 BCLK		section 16
0008 3044h	BSC	CS4 Wait Control Register 1	CS4WCR1	32	32	1, 2 BCLK		section 16
0008 3048h	BSC	CS4 Wait Control Register 2	CS4WCR2	32	32	1, 2 BCLK		section 16
0008 3052h	BSC	CS5 Mode Register	CS5MOD	16	16	1, 2 BCLK		section 16
0008 3054h	BSC	CS5 Wait Control Register 1	CS5WCR1	32	32	1, 2 BCLK		section 16
0008 3058h	BSC	CS5 Wait Control Register 2	CS5WCR2	32	32	1, 2 BCLK		section 16
0008 3062h	BSC	CS6 Mode Register	CS6MOD	16	16	1, 2 BCLK		section 16
0008 3064h	BSC	CS6 Wait Control Register 1	CS6WCR1	32	32	1, 2 BCLK		section 16
0008 3068h	BSC	CS6 Wait Control Register 2	CS6WCR2	32	32	1, 2 BCLK		section 16
0008 3072h	BSC	CS7 Mode Register	CS7MOD	16	16	1, 2 BCLK		section 16
0008 3074h	BSC	CS7 Wait Control Register 1	CS7WCR1	32	32	1, 2 BCLK		section 16
0008 3078h	BSC	CS7 Wait Control Register 2	CS7WCR2	32	32	1, 2 BCLK		section 16
0008 3802h	BSC	CS0 Control Register	CS0CR	16	16	1, 2 BCLK		section 16
0008 380Ah	BSC	CS0 Recovery Cycle Register	CS0REC	16	16	1, 2 BCLK		section 16
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		section 16
0008 3822h	BSC	CS2 Control Register	CS2CR	16	16	1, 2 BCLK		section 16
0008 382Ah	BSC	CS2 Recovery Cycle Register	CS2REC	16	16	1, 2 BCLK		section 16
0008 3832h	BSC	CS3 Control Register	CS3CR	16	16	1, 2 BCLK		section 16
0008 383Ah	BSC	CS3 Recovery Cycle Register	CS3REC	16	16	1, 2 BCLK		section 16
0008 3842h	BSC	CS4 Control Register	CS4CR	16	16	1, 2 BCLK		section 16
0008 384Ah	BSC	CS4 Recovery Cycle Register	CS4REC	16	16	1, 2 BCLK		section 16
0008 3852h	BSC	CS5 Control Register	CS5CR	16	16	1, 2 BCLK		section 16
0008 385Ah	BSC	CS5 Recovery Cycle Register	CS5REC	16	16	1, 2 BCLK		section 16
0008 3862h	BSC	CS6 Control Register	CS6CR	16	16	1, 2 BCLK		section 16
0008 386Ah	BSC	CS6 Recovery Cycle Register	CS6REC	16	16	1, 2 BCLK		section 16
0008 3872h	BSC	CS7 Control Register	CS7CR	16	16	1, 2 BCLK		section 16
0008 387Ah	BSC	CS7 Recovery Cycle Register	CS7REC	16	16	1, 2 BCLK		section 16
0008 3880h	BSC	CS Recovery Cycle Insertion Enable Register	CSRECEN	16	16	1, 2 BCLK		section 16
0008 3C00h	BSC	SDC Control Register	SDCCR	8	8	1, 2 BCLK		section 16
0008 3C01h	BSC	SDC Mode Register	SDCMOD	8	8	1, 2 BCLK		section 16
0008 3C02h	BSC	SDRAM Access Mode Register	SDAMOD	8	8	1, 2 BCLK		section 16
0008 3C10h	BSC	SDRAM Self-Refresh Control Register	SDSELF	8	8	1, 2 BCLK		section 16
0008 3C14h	BSC	SDRAM Refresh Control Register	SDRFCR	16	16	1, 2 BCLK		section 16
0008 3C16h	BSC	SDRAM Auto-Refresh Control Register	SDRFEN	8	8	1, 2 BCLK		section 16
0008 3C20h	BSC	SDRAM Initialization Sequence Control Register	SDICR	8	8	1, 2 BCLK		section 16
0008 3C24h	BSC	SDRAM Initialization Register	SDIR	16	16	1, 2 BCLK		section 16
0008 3C40h	BSC	SDRAM Address Register	SDADR	8	8	1, 2 BCLK		section 16
0008 3C44h	BSC	SDRAM Timing Register	SDTR	32	32	1, 2 BCLK		section 16
0008 3C48h	BSC	SDRAM Mode Register	SDMOD	16	16	1, 2 BCLK		section 16
0008 3C50h	BSC	SDRAM Status Register	SDSR	8	8	1, 2 BCLK		section 16
0008 6400h	MPU	Region-0 Start Page Number Register	RSPAGE0	32	32	1 ICLK		section 17
0008 6404h	MPU	Region-0 End Page Number Register	REPAGE0	32	32	1 ICLK		section 17
0008 6408h	MPU	Region-1 Start Page Number Register	RSPAGE1	32	32	1 ICLK		section 17

Table 5.1 List of I/O Registers (Address Order) (6 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 640Ch	MPU	Region-1 End Page Number Register	REPAGE1	32	32	1 ICLK		section 17
0008 6410h	MPU	Region-2 Start Page Number Register	RSPAGE2	32	32	1 ICLK		section 17
0008 6414h	MPU	Region-2 End Page Number Register	REPAGE2	32	32	1 ICLK		section 17
0008 6418h	MPU	Region-3 Start Page Number Register	RSPAGE3	32	32	1 ICLK		section 17
0008 641Ch	MPU	Region-3 End Page Number Register	REPAGE3	32	32	1 ICLK		section 17
0008 6420h	MPU	Region-4 Start Page Number Register	RSPAGE4	32	32	1 ICLK		section 17
0008 6424h	MPU	Region-4 End Page Number Register	REPAGE4	32	32	1 ICLK		section 17
0008 6428h	MPU	Region-5 Start Page Number Register	RSPAGE5	32	32	1 ICLK		section 17
0008 642Ch	MPU	Region-5 End Page Number Register	REPAGE5	32	32	1 ICLK		section 17
0008 6430h	MPU	Region-6 Start Page Number Register	RSPAGE6	32	32	1 ICLK		section 17
0008 6434h	MPU	Region-6 End Page Number Register	REPAGE6	32	32	1 ICLK		section 17
0008 6438h	MPU	Region-7 Start Page Number Register	RSPAGE7	32	32	1 ICLK		section 17
0008 643Ch	MPU	Region-7 End Page Number Register	REPAGE7	32	32	1 ICLK		section 17
0008 6500h	MPU	Memory-Protection Enable Register	MPEN	32	32	1 ICLK		section 17
0008 6504h	MPU	Background Access Control Register	MPBAC	32	32	1 ICLK		section 17
0008 6508h	MPU	Memory-Protection Error Status-Clearing Register	MPECLR	32	32	1 ICLK		section 17
0008 650Ch	MPU	Memory-Protection Error Status Register	MPESTS	32	32	1 ICLK		section 17
0008 6514h	MPU	Data Memory-Protection Error Address Register	MPDEA	32	32	1 ICLK		section 17
0008 6520h	MPU	Region Search Address Register	MPSA	32	32	1 ICLK		section 17
0008 6524h	MPU	Region Search Operation Register	MPOPS	16	16	1 ICLK		section 17
0008 6526h	MPU	Region Invalidation Operation Register	MPOPI	16	16	1 ICLK		section 17
0008 6528h	MPU	Instruction-Hit Region Register	MHITI	32	32	1 ICLK		section 17
0008 652Ch	MPU	Data-Hit Region Register	MHITD	32	32	1 ICLK		section 17
0008 7010h to 0008 70FFh	ICU	Interrupt Request Register 016 to Interrupt Request Register 255	IR016 to IR255	8	8	2 ICLK		section 15
0008 711Ah to 0008 71FFh	ICU	DTC Transfer Request Enable Register 026 to DTC Transfer Request Enable Register 255	DTCER026 to DTCER255	8	8	2 ICLK		section 15
0008 7202h to 0008 721Fh	ICU	Interrupt Request Enable Register 02 to Interrupt Request Enable Register 1F	IER02 to IER1F	8	8	2 ICLK		section 15
0008 72E0h	ICU	Software Interrupt Generation Register	SWINTR	8	8	2 ICLK		section 15
0008 72E1h	ICU	Software Interrupt 2 Generation Register	SWINT2R	8	8	2 ICLK		section 15
0008 72F0h	ICU	Fast Interrupt Set Register	FIR	16	16	2 ICLK		section 15
0008 7300h to 0008 73FFh	ICU	Interrupt Source Priority Register 000 to Interrupt Source Priority Register 255	IPR000 to IPR255	8	8	2 ICLK		section 15
0008 7400h	ICU	DMAC Trigger Select Register 0	DMRSR0	8	8	2 ICLK		section 15
0008 7404h	ICU	DMAC Trigger Select Register 1	DMRSR1	8	8	2 ICLK		section 15
0008 7408h	ICU	DMAC Trigger Select Register 2	DMRSR2	8	8	2 ICLK		section 15
0008 740Ch	ICU	DMAC Trigger Select Register 3	DMRSR3	8	8	2 ICLK		section 15
0008 7410h	ICU	DMAC Trigger Select Register 4	DMRSR4	8	8	2 ICLK		section 15
0008 7414h	ICU	DMAC Trigger Select Register 5	DMRSR5	8	8	2 ICLK		section 15
0008 7418h	ICU	DMAC Trigger Select Register 6	DMRSR6	8	8	2 ICLK		section 15
0008 741Ch	ICU	DMAC Trigger Select Register 7	DMRSR7	8	8	2 ICLK		section 15
0008 7500h to 0008 750Fh	ICU	IRQ Control Register 0 to IRQ Control Register 15	IRQCR0 to IRQCR15	8	8	2 ICLK		section 15
0008 7520h	ICU	IRQ Pin Digital Filter Enable Register 0	IRQFLTE0	8	8	2 ICLK		section 15
0008 7521h	ICU	IRQ Pin Digital Filter Enable Register 1	IRQFLTE1	8	8	2 ICLK		section 15
0008 7528h	ICU	IRQ Pin Digital Filter Setting Register 0	IRQFLTC0	16	16	2 ICLK		section 15
0008 752Ah	ICU	IRQ Pin Digital Filter Setting Register 1	IRQFLTC1	16	16	2 ICLK		section 15
0008 7580h	ICU	Non-Maskable Interrupt Status Register	NMISR	8	8	2 ICLK		section 15
0008 7581h	ICU	Non-Maskable Interrupt Enable Register	NMIER	8	8	2 ICLK		section 15
0008 7582h	ICU	Non-Maskable Interrupt Status Clear Register	NMICLR	8	8	2 ICLK		section 15
0008 7583h	ICU	NMI Pin Interrupt Control Register	NMICR	8	8	2 ICLK		section 15

Table 5.1 List of I/O Registers (Address Order) (7 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 7584h	ICU	Expanded Non-Maskable Interrupt Status Register	EXNMISR	8	8	2 ICLK		section 15
0008 7585h	ICU	Expanded Non-Maskable Interrupt Enable Register	EXNMIER	8	8	2 ICLK		section 15
0008 7586h	ICU	Expanded Non-Maskable Interrupt Status Clear Register	EXNMICLR	8	8	2 ICLK		section 15
0008 7590h	ICU	NMI Pin Digital Filter Enable Register	NMIFLTE	8	8	2 ICLK		section 15
0008 7594h	ICU	NMI Pin Digital Filter Setting Register	NMIFLTC	8	8	2 ICLK		section 15
0008 75B0h	ICU	Group IE0 Interrupt Request Register	GRPIE0	32	32	2 ICLK		section 15
0008 75B4h	ICU	Group IE0 Interrupt Request Enable Register	GENIE0	32	32	2 ICLK		section 15
0008 75B8h	ICU	Group IE0 Interrupt Clear Register	GCRIE0	32	32	2 ICLK		section 15
0008 7600h	ICU	Group BE0 Interrupt Request Register	GRPBE0	32	32	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7630h	ICU	Group BL0 Interrupt Request Register	GRPBL0	32	32	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7634h	ICU	Group BL1 Interrupt Request Register	GRPBL1	32	32	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7640h	ICU	Group BE0 Interrupt Request Enable Register	GENBE0	32	32	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7670h	ICU	Group BL0 Interrupt Request Enable Register	GENBL0	32	32	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7674h	ICU	Group BL1 Interrupt Request Enable Register	GENBL1	32	32	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7680h	ICU	Group BE0 Interrupt Clear Register	GCRBE0	32	32	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7700h	ICU	Software Configurable Interrupt B Request Register 0	PIBR0	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7701h	ICU	Software Configurable Interrupt B Request Register 1	PIBR1	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7702h	ICU	Software Configurable Interrupt B Request Register 2	PIBR2	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7703h	ICU	Software Configurable Interrupt B Request Register 3	PIBR3	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7704h	ICU	Software Configurable Interrupt B Request Register 4	PIBR4	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7705h	ICU	Software Configurable Interrupt B Request Register 5	PIBR5	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7706h	ICU	Software Configurable Interrupt B Request Register 6	PIBR6	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7707h	ICU	Software Configurable Interrupt B Request Register 7	PIBR7	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7708h	ICU	Software Configurable Interrupt B Request Register 8	PIBR8	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7709h	ICU	Software Configurable Interrupt B Request Register 9	PIBR9	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 770Ah	ICU	Software Configurable Interrupt B Request Register A	PIBRA	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 770Bh	ICU	Software Configurable Interrupt B Request Register B	PIBRB	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 770Ch	ICU	Software Configurable Interrupt B Request Register C	PIBRC	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7780h	ICU	Software Configurable Interrupt B Source Select Register X128	SLIBXR128	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7781h	ICU	Software Configurable Interrupt B Source Select Register X129	SLIBXR129	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7782h	ICU	Software Configurable Interrupt B Source Select Register X130	SLIBXR130	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7783h	ICU	Software Configurable Interrupt B Source Select Register X131	SLIBXR131	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7784h	ICU	Software Configurable Interrupt B Source Select Register X132	SLIBXR132	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7785h	ICU	Software Configurable Interrupt B Source Select Register X133	SLIBXR133	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7786h	ICU	Software Configurable Interrupt B Source Select Register X134	SLIBXR134	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15

Table 5.1 List of I/O Registers (Address Order) (8 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 7787h	ICU	Software Configurable Interrupt B Source Select Register X135	SLIBXR135	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7788h	ICU	Software Configurable Interrupt B Source Select Register X136	SLIBXR136	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7789h	ICU	Software Configurable Interrupt B Source Select Register X137	SLIBXR137	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 778Ah	ICU	Software Configurable Interrupt B Source Select Register X138	SLIBXR138	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 778Bh	ICU	Software Configurable Interrupt B Source Select Register X139	SLIBXR139	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 778Ch	ICU	Software Configurable Interrupt B Source Select Register X140	SLIBXR140	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 778Dh	ICU	Software Configurable Interrupt B Source Select Register X141	SLIBXR141	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 778Eh	ICU	Software Configurable Interrupt B Source Select Register X142	SLIBXR142	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 778Fh	ICU	Software Configurable Interrupt B Source Select Register X143	SLIBXR143	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7790h	ICU	Software Configurable Interrupt B Source Select Register 144	SLIBR144	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7791h	ICU	Software Configurable Interrupt B Source Select Register 145	SLIBR145	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7792h	ICU	Software Configurable Interrupt B Source Select Register 146	SLIBR146	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7793h	ICU	Software Configurable Interrupt B Source Select Register 147	SLIBR147	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7794h	ICU	Software Configurable Interrupt B Source Select Register 148	SLIBR148	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7795h	ICU	Software Configurable Interrupt B Source Select Register 149	SLIBR149	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7796h	ICU	Software Configurable Interrupt B Source Select Register 150	SLIBR150	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7797h	ICU	Software Configurable Interrupt B Source Select Register 151	SLIBR151	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7798h	ICU	Software Configurable Interrupt B Source Select Register 152	SLIBR152	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7799h	ICU	Software Configurable Interrupt B Source Select Register 153	SLIBR153	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 779Ah	ICU	Software Configurable Interrupt B Source Select Register 154	SLIBR154	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 779Bh	ICU	Software Configurable Interrupt B Source Select Register 155	SLIBR155	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 779Ch	ICU	Software Configurable Interrupt B Source Select Register 156	SLIBR156	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 779Dh	ICU	Software Configurable Interrupt B Source Select Register 157	SLIBR157	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 779Eh	ICU	Software Configurable Interrupt B Source Select Register 158	SLIBR158	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 779Fh	ICU	Software Configurable Interrupt B Source Select Register 159	SLIBR159	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77A0h	ICU	Software Configurable Interrupt B Source Select Register 160	SLIBR160	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77A1h	ICU	Software Configurable Interrupt B Source Select Register 161	SLIBR161	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77A2h	ICU	Software Configurable Interrupt B Source Select Register 162	SLIBR162	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77A3h	ICU	Software Configurable Interrupt B Source Select Register 163	SLIBR163	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77A4h	ICU	Software Configurable Interrupt B Source Select Register 164	SLIBR164	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77A5h	ICU	Software Configurable Interrupt B Source Select Register 165	SLIBR165	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15

Table 5.1 List of I/O Registers (Address Order) (9 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 77A6h	ICU	Software Configurable Interrupt B Source Select Register 166	SLIBR166	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77A7h	ICU	Software Configurable Interrupt B Source Select Register 167	SLIBR167	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77A8h	ICU	Software Configurable Interrupt B Source Select Register 168	SLIBR168	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77A9h	ICU	Software Configurable Interrupt B Source Select Register 169	SLIBR169	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77AAh	ICU	Software Configurable Interrupt B Source Select Register 170	SLIBR170	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77ABh	ICU	Software Configurable Interrupt B Source Select Register 171	SLIBR171	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77ACh	ICU	Software Configurable Interrupt B Source Select Register 172	SLIBR172	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77ADh	ICU	Software Configurable Interrupt B Source Select Register 173	SLIBR173	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77AEh	ICU	Software Configurable Interrupt B Source Select Register 174	SLIBR174	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77AFh	ICU	Software Configurable Interrupt B Source Select Register 175	SLIBR175	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77B0h	ICU	Software Configurable Interrupt B Source Select Register 176	SLIBR176	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77B1h	ICU	Software Configurable Interrupt B Source Select Register 177	SLIBR177	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77B2h	ICU	Software Configurable Interrupt B Source Select Register 178	SLIBR178	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77B3h	ICU	Software Configurable Interrupt B Source Select Register 179	SLIBR179	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77B4h	ICU	Software Configurable Interrupt B Source Select Register 180	SLIBR180	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77B5h	ICU	Software Configurable Interrupt B Source Select Register 181	SLIBR181	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77B6h	ICU	Software Configurable Interrupt B Source Select Register 182	SLIBR182	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77B7h	ICU	Software Configurable Interrupt B Source Select Register 183	SLIBR183	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77B8h	ICU	Software Configurable Interrupt B Source Select Register 184	SLIBR184	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77B9h	ICU	Software Configurable Interrupt B Source Select Register 185	SLIBR185	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77BAh	ICU	Software Configurable Interrupt B Source Select Register 186	SLIBR186	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77BBh	ICU	Software Configurable Interrupt B Source Select Register 187	SLIBR187	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77BCh	ICU	Software Configurable Interrupt B Source Select Register 188	SLIBR188	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77BDh	ICU	Software Configurable Interrupt B Source Select Register 189	SLIBR189	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77BEh	ICU	Software Configurable Interrupt B Source Select Register 190	SLIBR190	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77BFh	ICU	Software Configurable Interrupt B Source Select Register 191	SLIBR191	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77C0h	ICU	Software Configurable Interrupt B Source Select Register 192	SLIBR192	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77C1h	ICU	Software Configurable Interrupt B Source Select Register 193	SLIBR193	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77C2h	ICU	Software Configurable Interrupt B Source Select Register 194	SLIBR194	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77C3h	ICU	Software Configurable Interrupt B Source Select Register 195	SLIBR195	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77C4h	ICU	Software Configurable Interrupt B Source Select Register 196	SLIBR196	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15

Table 5.1 List of I/O Registers (Address Order) (10 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 77C5h	ICU	Software Configurable Interrupt B Source Select Register 197	SLIBR197	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77C6h	ICU	Software Configurable Interrupt B Source Select Register 198	SLIBR198	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77C7h	ICU	Software Configurable Interrupt B Source Select Register 199	SLIBR199	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77C8h	ICU	Software Configurable Interrupt B Source Select Register 200	SLIBR200	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77C9h	ICU	Software Configurable Interrupt B Source Select Register 201	SLIBR201	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77CAh	ICU	Software Configurable Interrupt B Source Select Register 202	SLIBR202	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77CBh	ICU	Software Configurable Interrupt B Source Select Register 203	SLIBR203	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77CCh	ICU	Software Configurable Interrupt B Source Select Register 204	SLIBR204	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77CDh	ICU	Software Configurable Interrupt B Source Select Register 205	SLIBR205	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77CEh	ICU	Software Configurable Interrupt B Source Select Register 206	SLIBR206	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 77CFh	ICU	Software Configurable Interrupt B Source Select Register 207	SLIBR207	8	8	2 ICLK to 1 PCLKB	2 ICLK	section 15
0008 7830h	ICU	Group AL0 Interrupt Request Register	GRPAL0	32	32	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 7834h	ICU	Group AL1 Interrupt Request Register	GRPAL1	32	32	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 7870h	ICU	Group AL0 Interrupt Request Enable Register	GENAL0	32	32	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 7874h	ICU	Group AL1 Interrupt Request Enable Register	GENAL1	32	32	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 7900h	ICU	Software Configurable Interrupt A Request Register 0	PIAR0	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 7901h	ICU	Software Configurable Interrupt A Request Register 1	PIAR1	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 7902h	ICU	Software Configurable Interrupt A Request Register 2	PIAR2	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 7903h	ICU	Software Configurable Interrupt A Request Register 3	PIAR3	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 7904h	ICU	Software Configurable Interrupt A Request Register 4	PIAR4	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 7905h	ICU	Software Configurable Interrupt A Request Register 5	PIAR5	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 790Bh	ICU	Software Configurable Interrupt A Request Register B	PIARB	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79D0h	ICU	Software Configurable Interrupt A Source Select Register 208	SLIAR208	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79D1h	ICU	Software Configurable Interrupt A Source Select Register 209	SLIAR209	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79D2h	ICU	Software Configurable Interrupt A Source Select Register 210	SLIAR210	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79D3h	ICU	Software Configurable Interrupt A Source Select Register 211	SLIAR211	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79D4h	ICU	Software Configurable Interrupt A Source Select Register 212	SLIAR212	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79D5h	ICU	Software Configurable Interrupt A Source Select Register 213	SLIAR213	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79D6h	ICU	Software Configurable Interrupt A Source Select Register 214	SLIAR214	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79D7h	ICU	Software Configurable Interrupt A Source Select Register 215	SLIAR215	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79D8h	ICU	Software Configurable Interrupt A Source Select Register 216	SLIAR216	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79D9h	ICU	Software Configurable Interrupt A Source Select Register 217	SLIAR217	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15

Table 5.1 List of I/O Registers (Address Order) (11 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 79DAh	ICU	Software Configurable Interrupt A Source Select Register 218	SLIAR218	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79DBh	ICU	Software Configurable Interrupt A Source Select Register 219	SLIAR219	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79DCh	ICU	Software Configurable Interrupt A Source Select Register 220	SLIAR220	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79DDh	ICU	Software Configurable Interrupt A Source Select Register 221	SLIAR221	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79DEh	ICU	Software Configurable Interrupt A Source Select Register 222	SLIAR222	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79DFh	ICU	Software Configurable Interrupt A Source Select Register 223	SLIAR223	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79E0h	ICU	Software Configurable Interrupt A Source Select Register 224	SLIAR224	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79E1h	ICU	Software Configurable Interrupt A Source Select Register 225	SLIAR225	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79E2h	ICU	Software Configurable Interrupt A Source Select Register 226	SLIAR226	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79E3h	ICU	Software Configurable Interrupt A Source Select Register 227	SLIAR227	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79E4h	ICU	Software Configurable Interrupt A Source Select Register 228	SLIAR228	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79E5h	ICU	Software Configurable Interrupt A Source Select Register 229	SLIAR229	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79E6h	ICU	Software Configurable Interrupt A Source Select Register 230	SLIAR230	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79E7h	ICU	Software Configurable Interrupt A Source Select Register 231	SLIAR231	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79E8h	ICU	Software Configurable Interrupt A Source Select Register 232	SLIAR232	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79E9h	ICU	Software Configurable Interrupt A Source Select Register 233	SLIAR233	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79EAh	ICU	Software Configurable Interrupt A Source Select Register 234	SLIAR234	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79EBh	ICU	Software Configurable Interrupt A Source Select Register 235	SLIAR235	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79ECh	ICU	Software Configurable Interrupt A Source Select Register 236	SLIAR236	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79EDh	ICU	Software Configurable Interrupt A Source Select Register 237	SLIAR237	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79EEh	ICU	Software Configurable Interrupt A Source Select Register 238	SLIAR238	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79EFh	ICU	Software Configurable Interrupt A Source Select Register 239	SLIAR239	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79F0h	ICU	Software Configurable Interrupt A Source Select Register 240	SLIAR240	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79F1h	ICU	Software Configurable Interrupt A Source Select Register 241	SLIAR241	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79F2h	ICU	Software Configurable Interrupt A Source Select Register 242	SLIAR242	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79F3h	ICU	Software Configurable Interrupt A Source Select Register 243	SLIAR243	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79F4h	ICU	Software Configurable Interrupt A Source Select Register 244	SLIAR244	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79F5h	ICU	Software Configurable Interrupt A Source Select Register 245	SLIAR245	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79F6h	ICU	Software Configurable Interrupt A Source Select Register 246	SLIAR246	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79F7h	ICU	Software Configurable Interrupt A Source Select Register 247	SLIAR247	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79F8h	ICU	Software Configurable Interrupt A Source Select Register 248	SLIAR248	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15

Table 5.1 List of I/O Registers (Address Order) (12 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 79F9h	ICU	Software Configurable Interrupt A Source Select Register 249	SLIAR249	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79FAh	ICU	Software Configurable Interrupt A Source Select Register 250	SLIAR250	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79FBh	ICU	Software Configurable Interrupt A Source Select Register 251	SLIAR251	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79FCh	ICU	Software Configurable Interrupt A Source Select Register 252	SLIAR252	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79FDh	ICU	Software Configurable Interrupt A Source Select Register 253	SLIAR253	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79FEh	ICU	Software Configurable Interrupt A Source Select Register 254	SLIAR254	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 79FFh	ICU	Software Configurable Interrupt A Source Select Register 255	SLIAR255	8	8	2 ICLK to 1 PCLKA	2 ICLK	section 15
0008 7A00h	ICU	Software Configurable Interrupt Source Select Register Write Protect Register	SLIPRCR	8	8	2 ICLK to 1 PCLKA/B	2 ICLK	section 15
0008 7A01h	ICU	EXDMAC Trigger Select Register	SELEXDR	8	8	2 ICLK to 1 PCLKA/B	2 ICLK	section 15
0008 8000h	CMT	Compare Match Timer Start Register 0	CMSTR0	16	16	2, 3 PCLKB	2 ICLK	section 29
0008 8002h	CMT0	Compare Match Timer Control Register	CMCR	16	16	2, 3 PCLKB	2 ICLK	section 29
0008 8004h	CMT0	Compare Match Counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK	section 29
0008 8006h	CMT0	Compare Match Constant Register	CMCOR	16	16	2, 3 PCLKB	2 ICLK	section 29
0008 8008h	CMT1	Compare Match Timer Control Register	CMCR	16	16	2, 3 PCLKB	2 ICLK	section 29
0008 800Ah	CMT1	Compare Match Counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK	section 29
0008 800Ch	CMT1	Compare Match Constant Register	CMCOR	16	16	2, 3 PCLKB	2 ICLK	section 29
0008 8010h	CMT	Compare Match Timer Start Register 1	CMSTR1	16	16	2, 3 PCLKB	2 ICLK	section 29
0008 8012h	CMT2	Compare Match Timer Control Register	CMCR	16	16	2, 3 PCLKB	2 ICLK	section 29
0008 8014h	CMT2	Compare Match Counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK	section 29
0008 8016h	CMT2	Compare Match Constant Register	CMCOR	16	16	2, 3 PCLKB	2 ICLK	section 29
0008 8018h	CMT3	Compare Match Timer Control Register	CMCR	16	16	2, 3 PCLKB	2 ICLK	section 29
0008 801Ah	CMT3	Compare Match Counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK	section 29
0008 801Ch	CMT3	Compare Match Constant Register	CMCOR	16	16	2, 3 PCLKB	2 ICLK	section 29
0008 8020h	WDT	WDT Refresh Register	WDTRR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8022h	WDT	WDT Control Register	WDTCR	16	16	2, 3 PCLKB	2 ICLK	section 32
0008 8024h	WDT	WDT Status Register	WDTSR	16	16	2, 3 PCLKB	2 ICLK	section 32
0008 8026h	WDT	WDT Reset Control Register	WDTRCR	8	8	2, 3 PCLKB	2 ICLK	section 32
0008 8030h	IWDT	IWDT Refresh Register	IWDTRR	8	8	2, 3 PCLKB	2 ICLK	section 33
0008 8032h	IWDT	IWDT Control Register	IWDTCR	16	16	2, 3 PCLKB	2 ICLK	section 33
0008 8034h	IWDT	IWDT Status Register	IWDTSR	16	16	2, 3 PCLKB	2 ICLK	section 33
0008 8036h	IWDT	IWDT Reset Control Register	IWDTRCR	8	8	2, 3 PCLKB	2 ICLK	section 33
0008 8038h	IWDT	IWDT Count Stop Control Register	IWDTCSTPR	8	8	2, 3 PCLKB	2 ICLK	section 33
0008 8100h	TPUA	Timer Start Register	TSTR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8101h	TPUA	Timer Synchronous Register	TSYR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8108h	TPU0	Noise Filter Control Register	NFCR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8109h	TPU1	Noise Filter Control Register	NFCR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 810Ah	TPU2	Noise Filter Control Register	NFCR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 810Bh	TPU3	Noise Filter Control Register	NFCR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 810Ch	TPU4	Noise Filter Control Register	NFCR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 810Dh	TPU5	Noise Filter Control Register	NFCR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8110h	TPU0	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8111h	TPU0	Timer Mode Register	TMDR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8112h	TPU0	Timer I/O Control Register H	TIORH	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8113h	TPU0	Timer I/O Control Register L	TIORL	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8114h	TPU0	Timer Interrupt Enable Register	TIER	8	8	2, 3 PCLKB	2 ICLK	section 26

Table 5.1 List of I/O Registers (Address Order) (13 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 8115h	TPU0	Timer Status Register	TSR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8116h	TPU0	Timer Counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	section 26
0008 8118h	TPU0	Timer General Register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	section 26
0008 811Ah	TPU0	Timer General Register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	section 26
0008 811Ch	TPU0	Timer General Register C	TGRC	16	16	2, 3 PCLKB	2 ICLK	section 26
0008 811Eh	TPU0	Timer General Register D	TGRD	16	16	2, 3 PCLKB	2 ICLK	section 26
0008 8120h	TPU1	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8121h	TPU1	Timer Mode Register	TMDR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8122h	TPU1	Timer I/O Control Register	TIOR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8124h	TPU1	Timer Interrupt Enable Register	TIER	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8125h	TPU1	Timer Status Register	TSR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8126h	TPU1	Timer Counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	section 26
0008 8128h	TPU1	Timer General Register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	section 26
0008 812Ah	TPU1	Timer General Register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	section 26
0008 8130h	TPU2	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8131h	TPU2	Timer Mode Register	TMDR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8132h	TPU2	Timer I/O Control Register	TIOR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8134h	TPU2	Timer Interrupt Enable Register	TIER	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8135h	TPU2	Timer Status Register	TSR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8136h	TPU2	Timer Counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	section 26
0008 8138h	TPU2	Timer General Register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	section 26
0008 813Ah	TPU2	Timer General Register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	section 26
0008 8140h	TPU3	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8141h	TPU3	Timer Mode Register	TMDR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8142h	TPU3	Timer I/O Control Register H	TIORH	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8143h	TPU3	Timer I/O Control Register L	TIORL	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8144h	TPU3	Timer Interrupt Enable Register	TIER	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8145h	TPU3	Timer Status Register	TSR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8146h	TPU3	Timer Counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	section 26
0008 8148h	TPU3	Timer General Register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	section 26
0008 814Ah	TPU3	Timer General Register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	section 26
0008 814Ch	TPU3	Timer General Register C	TGRC	16	16	2, 3 PCLKB	2 ICLK	section 26
0008 814Eh	TPU3	Timer General Register D	TGRD	16	16	2, 3 PCLKB	2 ICLK	section 26
0008 8150h	TPU4	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8151h	TPU4	Timer Mode Register	TMDR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8152h	TPU4	Timer I/O Control Register	TIOR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8154h	TPU4	Timer Interrupt Enable Register	TIER	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8155h	TPU4	Timer Status Register	TSR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8156h	TPU4	Timer Counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	section 26
0008 8158h	TPU4	Timer General Register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	section 26
0008 815Ah	TPU4	Timer General Register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	section 26
0008 8160h	TPU5	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8161h	TPU5	Timer Mode Register	TMDR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8162h	TPU5	Timer I/O Control Register	TIOR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8164h	TPU5	Timer Interrupt Enable Register	TIER	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8165h	TPU5	Timer Status Register	TSR	8	8	2, 3 PCLKB	2 ICLK	section 26
0008 8166h	TPU5	Timer Counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	section 26
0008 8168h	TPU5	Timer General Register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	section 26
0008 816Ah	TPU5	Timer General Register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	section 26
0008 81E6h	PPG0	PPG Output Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 27

Table 5.1 List of I/O Registers (Address Order) (14 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 81E7h	PPG0	PPG Output Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 81E8h	PPG0	Next Data Enable Register H	NDERH	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 81E9h	PPG0	Next Data Enable Register L	NDERL	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 81EAh	PPG0	Output Data Register H	PODRH	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 81EBh	PPG0	Output Data Register L	PODRL	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 81ECh	PPG0	Next Data Register H	NDRH	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 81EDh	PPG0	Next Data Register L	NDRL	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 81EEh	PPG0	Next Data Register H2	NDRH2	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 81EFh	PPG0	Next Data Register L2	NDRL2	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 81F0h	PPG1	PPG Trigger Select Register	PTRSLR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 81F6h	PPG1	PPG Output Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 81F7h	PPG1	PPG Output Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 81F8h	PPG1	Next Data Enable Register H	NDERH	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 81F9h	PPG1	Next Data Enable Register L	NDERL	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 81FAh	PPG1	Output Data Register H	PODRH	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 81FBh	PPG1	Output Data Register L	PODRL	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 81FCh	PPG1	Next Data Register H	NDRH	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 81FDh	PPG1	Next Data Register L	NDRL	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 81FEh	PPG1	Next Data Register H2	NDRH2	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 81FFh	PPG1	Next Data Register L2	NDRL2	8	8	2, 3 PCLKB	2 ICLK	section 27
0008 8200h	TMR0	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 8201h	TMR1	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 8202h	TMR0	Timer Control/Status Register	TCSR	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 8203h	TMR1	Timer Control/Status Register	TCSR	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 8204h	TMR0	Time Constant Register A	TCORA	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 8204h	TMR01	Time Constant Register A	TCORA	16	16	2, 3 PCLKB	2 ICLK	section 28
0008 8205h	TMR1	Time Constant Register A	TCORA	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 8206h	TMR0	Time Constant Register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 8206h	TMR01	Time Constant Register B	TCORB	16	16	2, 3 PCLKB	2 ICLK	section 28
0008 8207h	TMR1	Time Constant Register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 8208h	TMR0	Timer Counter	TCNT	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 8208h	TMR01	Timer Counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	section 28
0008 8209h	TMR1	Timer Counter	TCNT	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 820Ah	TMR0	Timer Counter Control Register	TCCR	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 820Ah	TMR01	Timer Counter Control Register	TCCR	16	16	2, 3 PCLKB	2 ICLK	section 28
0008 820Bh	TMR1	Timer Counter Control Register	TCCR	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 820Ch	TMR0	Timer Counter Start Register	TCSTR	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 820Dh	TMR1	Timer Counter Start Register	TCSTR	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 8210h	TMR2	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 8211h	TMR3	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 8212h	TMR2	Timer Control/Status Register	TCSR	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 8213h	TMR3	Timer Control/Status Register	TCSR	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 8214h	TMR2	Time Constant Register A	TCORA	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 8214h	TMR23	Time Constant Register A	TCORA	16	16	2, 3 PCLKB	2 ICLK	section 28
0008 8215h	TMR3	Time Constant Register A	TCORA	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 8216h	TMR2	Time Constant Register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 8216h	TMR23	Time Constant Register B	TCORB	16	16	2, 3 PCLKB	2 ICLK	section 28
0008 8217h	TMR3	Time Constant Register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 8218h	TMR2	Timer Counter	TCNT	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 8218h	TMR23	Timer Counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	section 28

Table 5.1 List of I/O Registers (Address Order) (15 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 8219h	TMR3	Timer Counter	TCNT	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 821Ah	TMR2	Timer Counter Control Register	TCCR	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 821Ah	TMR23	Timer Counter Control Register	TCCR	16	16	2, 3 PCLKB	2 ICLK	section 28
0008 821Bh	TMR3	Timer Counter Control Register	TCCR	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 821Ch	TMR2	Timer Counter Start Register	TCSTR	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 821Dh	TMR3	Timer Counter Start Register	TCSTR	8	8	2, 3 PCLKB	2 ICLK	section 28
0008 8280h	CRC	CRC Control Register	CRCCR	8	8	2, 3 PCLKB	2 ICLK	section 43
0008 8284h	CRC	CRC Data Input Register	CRCDIR	32	8, 32	2, 3 PCLKB	2 ICLK	section 43
0008 8288h	CRC	CRC Data Output Register	CRCDOR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	section 43
0008 8300h	RIIC0	I²C-bus Control Register 1	ICCR1	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8301h	RIIC0	I²C-bus Control Register 2	ICCR2	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8302h	RIIC0	I²C-bus Mode Register 1	ICMR1	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8303h	RIIC0	I²C-bus Mode Register 2	ICMR2	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8304h	RIIC0	I²C-bus Mode Register 3	ICMR3	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8305h	RIIC0	I²C-bus Function Enable Register	ICFER	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8306h	RIIC0	I²C-bus Status Enable Register	ICSER	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8307h	RIIC0	I²C-bus Interrupt Enable Register	ICIER	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8308h	RIIC0	I²C-bus Status Register 1	ICSR1	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8309h	RIIC0	I²C-bus Status Register 2	ICSR2	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 830Ah	RIIC0	Slave Address Register L0	SARL0	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 830Bh	RIIC0	Slave Address Register U0	SARU0	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 830Ch	RIIC0	Slave Address Register L1	SARL1	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 830Dh	RIIC0	Slave Address Register U1	SARU1	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 830Eh	RIIC0	Slave Address Register L2	SARL2	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 830Fh	RIIC0	Slave Address Register U2	SARU2	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8310h	RIIC0	I²C-bus Bit Rate Low-Level Register	ICBRL	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8311h	RIIC0	I²C-bus Bit Rate High-Level Register	ICBRH	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8312h	RIIC0	I²C-bus Transmit Data Register	ICDRT	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8313h	RIIC0	I²C-bus Receive Data Register	ICDRR	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8320h	RIIC1	I²C-bus Control Register 1	ICCR1	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8321h	RIIC1	I²C-bus Control Register 2	ICCR2	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8322h	RIIC1	I²C-bus Mode Register 1	ICMR1	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8323h	RIIC1	I²C-bus Mode Register 2	ICMR2	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8324h	RIIC1	I²C-bus Mode Register 3	ICMR3	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8325h	RIIC1	I²C-bus Function Enable Register	ICFER	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8326h	RIIC1	I²C-bus Status Enable Register	ICSER	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8327h	RIIC1	I²C-bus Interrupt Enable Register	ICIER	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8328h	RIIC1	I²C-bus Status Register 1	ICSR1	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8329h	RIIC1	I²C-bus Status Register 2	ICSR2	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 832Ah	RIIC1	Slave Address Register L0	SARL0	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 832Bh	RIIC1	Slave Address Register U0	SARU0	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 832Ch	RIIC1	Slave Address Register L1	SARL1	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 832Dh	RIIC1	Slave Address Register U1	SARU1	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 832Eh	RIIC1	Slave Address Register L2	SARL2	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 832Fh	RIIC1	Slave Address Register U2	SARU2	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8330h	RIIC1	I²C-bus Bit Rate Low-Level Register	ICBRL	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8331h	RIIC1	I²C-bus Bit Rate High-Level Register	ICBRH	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8332h	RIIC1	I²C-bus Transmit Data Register	ICDRT	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8333h	RIIC1	I²C-bus Receive Data Register	ICDRR	8	8	2, 3 PCLKB	2 ICLK	section 37

Table 5.1 List of I/O Registers (Address Order) (16 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 8340h	RIIC2	I²C-bus Control Register 1	ICCR1	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8341h	RIIC2	I²C-bus Control Register 2	ICCR2	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8342h	RIIC2	I²C-bus Mode Register 1	ICMR1	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8343h	RIIC2	I²C-bus Mode Register 2	ICMR2	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8344h	RIIC2	I²C-bus Mode Register 3	ICMR3	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8345h	RIIC2	I²C-bus Function Enable Register	ICFER	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8346h	RIIC2	I²C-bus Status Enable Register	ICSER	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8347h	RIIC2	I²C-bus Interrupt Enable Register	ICIER	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8348h	RIIC2	I²C-bus Status Register 1	ICSR1	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8349h	RIIC2	I²C-bus Status Register 2	ICSR2	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 834Ah	RIIC2	Slave Address Register L0	SARL0	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 834Bh	RIIC2	Slave Address Register U0	SARU0	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 834Ch	RIIC2	Slave Address Register L1	SARL1	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 834Dh	RIIC2	Slave Address Register U1	SARU1	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 834Eh	RIIC2	Slave Address Register L2	SARL2	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 834Fh	RIIC2	Slave Address Register U2	SARU2	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8350h	RIIC2	I²C-bus Bit Rate Low-Level Register	ICBRL	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8351h	RIIC2	I²C-bus Bit Rate High-Level Register	ICBRH	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8352h	RIIC2	I²C-bus Transmit Data Register	ICDRT	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 8353h	RIIC2	I²C-bus Receive Data Register	ICDRR	8	8	2, 3 PCLKB	2 ICLK	section 37
0008 9000h	S12AD	A/D Control Register	ADCSR	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9004h	S12AD	A/D Channel Select Register A0	ADANSA0	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9008h	S12AD	A/D-Converted Value Addition/Average Function Channel Select Register 0	ADADS0	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 900Ch	S12AD	A/D-Converted Value Addition/Average Count Select Register	ADADC	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 900Eh	S12AD	A/D Control Extended Register	ADCER	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9010h	S12AD	A/D Conversion Start Trigger Select Register	ADSTRGR	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9014h	S12AD	A/D Channel Select Register B0	ADANSB0	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9018h	S12AD	A/D Data Duplication Register	ADDBLDR	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 901Eh	S12AD	A/D Self-Diagnosis Data Register	ADRD	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9020h	S12AD	A/D Data Register 0	ADDR0	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9022h	S12AD	A/D Data Register 1	ADDR1	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9024h	S12AD	A/D Data Register 2	ADDR2	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9026h	S12AD	A/D Data Register 3	ADDR3	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9028h	S12AD	A/D Data Register 4	ADDR4	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 902Ah	S12AD	A/D Data Register 5	ADDR5	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 902Ch	S12AD	A/D Data Register 6	ADDR6	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 902Eh	S12AD	A/D Data Register 7	ADDR7	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9063h	S12AD	A/D Conversion Time Setting Protection Release Register	ADSAMPR	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 906Eh	S12AD	A/D Conversion Time Setting Register	ADSAM	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 907Ah	S12AD	A/D Disconnection Detection Control Register	ADDISCR	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 9080h	S12AD	A/D Group Scan Priority Control Register	ADGSPCR	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9084h	S12AD	A/D Data Duplication Register A	ADDBLDRA	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9086h	S12AD	A/D Data Duplication Register B	ADDBLDRB	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 908Ch	S12AD	A/D Comparison Function Window A/B Status Monitoring Register	ADWINMON	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 9090h	S12AD	A/D Comparison Function Control Register	ADCMPCR	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9094h	S12AD	A/D Comparison Function Window A Channel Select Register 0	ADCMPANSR 0	16	16	2, 3 PCLKB	2 ICLK	section 50

Table 5.1 List of I/O Registers (Address Order) (17 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 9098h	S12AD	A/D Comparison Function Window A Comparison Condition Setting Register 0	ADCMPLR0	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 909Ch	S12AD	A/D Comparison Function Window A Lower Level Setting Register	ADCMPDR0	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 909Eh	S12AD	A/D Comparison Function Window A Upper Level Setting Register	ADCMPDR1	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 90A0h	S12AD	A/D Comparison Function Window A Channel Status Register 0	ADCMPSR0	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 90A6h	S12AD	A/D Comparison Function Window B Channel Select Register	ADCMPBNSR	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 90A8h	S12AD	A/D Comparison Function Window B Lower Level Setting Register	ADWINLLB	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 90AAh	S12AD	A/D Comparison Function Window B Upper Level Setting Register	ADWINULB	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 90ACh	S12AD	A/D Comparison Function Window B Channel Status Register	ADCMPBSR	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 90D4h	S12AD	A/D Channel Select Register C0	ADANSC0	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 90D9h	S12AD	A/D Group C Trigger Select Register	ADGCTRGR	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 90E0h	S12AD	A/D Sampling State Register 0	ADSSTR0	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 90E1h	S12AD	A/D Sampling State Register 1	ADSSTR1	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 90E2h	S12AD	A/D Sampling State Register 2	ADSSTR2	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 90E3h	S12AD	A/D Sampling State Register 3	ADSSTR3	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 90E4h	S12AD	A/D Sampling State Register 4	ADSSTR4	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 90E5h	S12AD	A/D Sampling State Register 5	ADSSTR5	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 90E6h	S12AD	A/D Sampling State Register 6	ADSSTR6	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 90E7h	S12AD	A/D Sampling State Register 7	ADSSTR7	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 9100h	S12AD1	A/D Control Register	ADCSR	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9104h	S12AD1	A/D Channel Select Register A0	ADANSA0	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9108h	S12AD1	A/D-Converted Value Addition/Average Function Channel Select Register 0	ADADS0	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 910Ch	S12AD1	A/D-Converted Value Addition/Average Count Select Register	ADADC	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 910Eh	S12AD1	A/D Control Extended Register	ADCER	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9110h	S12AD1	A/D Conversion Start Trigger Select Register	ADSTRGR	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9112h	S12AD1	A/D Conversion Extended Input Control Register	ADEXICR	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9114h	S12AD1	A/D Channel Select Register B0	ADANSB0	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9118h	S12AD1	A/D Data Duplication Register	ADDBLDR	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 911Ah	S12AD1	A/D Temperature Sensor Data Register	ADTSDR	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 911Ch	S12AD1	A/D Internal Reference Voltage Data Register	ADOCDR	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 911Eh	S12AD1	A/D Self-Diagnosis Data Register	ADRD	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9120h	S12AD1	A/D Data Register 0	ADDR0	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9122h	S12AD1	A/D Data Register 1	ADDR1	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9124h	S12AD1	A/D Data Register 2	ADDR2	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9126h	S12AD1	A/D Data Register 3	ADDR3	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9128h	S12AD1	A/D Data Register 4	ADDR4	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 912Ah	S12AD1	A/D Data Register 5	ADDR5	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 912Ch	S12AD1	A/D Data Register 6	ADDR6	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 912Eh	S12AD1	A/D Data Register 7	ADDR7	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9130h	S12AD1	A/D Data Register 8	ADDR8	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9132h	S12AD1	A/D Data Register 9	ADDR9	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9134h	S12AD1	A/D Data Register 10	ADDR10	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9136h	S12AD1	A/D Data Register 11	ADDR11	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9163h	S12AD1	A/D Conversion Time Setting Protection Release Register	ADSAMPR	8	8	2, 3 PCLKB	2 ICLK	section 50

Table 5.1 List of I/O Registers (Address Order) (18 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 916Eh	S12AD1	A/D Conversion Time Setting Register	ADSAM	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 917Ah	S12AD1	A/D Disconnection Detection Control Register	ADDISCR	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 9180h	S12AD1	A/D Group Scan Priority Control Register	ADGSPCR	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9184h	S12AD1	A/D Data Duplication Register A	ADDBLDRA	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9186h	S12AD1	A/D Data Duplication Register B	ADDBLDRB	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 918Ch	S12AD1	A/D Comparison Function Window A/B Status Monitoring Register	ADWINMON	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 9190h	S12AD1	A/D Comparison Function Control Register	ADCMPCR	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9192h	S12AD1	A/D Comparison Function Window A Extended Input Select Register	ADCMPANSER	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 9193h	S12AD1	A/D Comparison Function Window A Extended Input Comparison Condition Setting Register	ADCMPLER	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 9194h	S12AD1	A/D Comparison Function Window A Channel Select Register 0	ADCMPANSR0	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 9198h	S12AD1	A/D Comparison Function Window A Comparison Condition Setting Register 0	ADCMPLR0	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 919Ch	S12AD1	A/D Comparison Function Window A Lower Level Setting Register	ADCMPDR0	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 919Eh	S12AD1	A/D Comparison Function Window A Upper Level Setting Register	ADCMPDR1	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 91A0h	S12AD1	A/D Comparison Function Window A Channel Status Register 0	ADCMPSR0	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 91A4h	S12AD1	A/D Comparison Function Window A Extended Input Channel Status Register	ADCMPSER	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 91A6h	S12AD1	A/D Comparison Function Window B Channel Select Register	ADCMPBNSR	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 91A8h	S12AD1	A/D Comparison Function Window B Lower Level Setting Register	ADWINLLB	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 91AAh	S12AD1	A/D Comparison Function Window B Upper Level Setting Register	ADWINULB	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 91ACh	S12AD1	A/D Comparison Function Window B Channel Status Register	ADCMPBSR	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 91D4h	S12AD1	A/D Channel Select Register C0	ADANSC0	16	16	2, 3 PCLKB	2 ICLK	section 50
0008 91D8h	S12AD1	A/D Group C Extended Input Control Register	ADGCEXCR	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 91D9h	S12AD1	A/D Group C Trigger Select Register	ADGCTRGR	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 91DEh	S12AD1	A/D Sampling State Register T	ADSSTRT	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 91DFh	S12AD1	A/D Sampling State Register O	ADSSTRO	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 91E0h	S12AD1	A/D Sampling State Register 0	ADSSTR0	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 91E1h	S12AD1	A/D Sampling State Register 1	ADSSTR1	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 91E2h	S12AD1	A/D Sampling State Register 2	ADSSTR2	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 91E3h	S12AD1	A/D Sampling State Register 3	ADSSTR3	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 91E4h	S12AD1	A/D Sampling State Register 4	ADSSTR4	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 91E5h	S12AD1	A/D Sampling State Register 5	ADSSTR5	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 91E6h	S12AD1	A/D Sampling State Register 6	ADSSTR6	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 91E7h	S12AD1	A/D Sampling State Register 7	ADSSTR7	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 91E8h	S12AD1	A/D Sampling State Register 8	ADSSTR8	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 91E9h	S12AD1	A/D Sampling State Register 9	ADSSTR9	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 91EAh	S12AD1	A/D Sampling State Register 10	ADSSTR10	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 91EBh	S12AD1	A/D Sampling State Register 11	ADSSTR11	8	8	2, 3 PCLKB	2 ICLK	section 50
0008 A000h	SCI0	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A000h	SMCI0	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A001h	SCI0	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A002h	SCI0	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A002h	SMCI0	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A003h	SCI0	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 35

Table 5.1 List of I/O Registers (Address Order) (19 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 A004h	SCI0	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A004h	SMCI0	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A005h	SCI0	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A006h	SCI0	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A006h	SMCI0	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A007h	SCI0	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A008h	SCI0	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A009h	SCI0	I^2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A00Ah	SCI0	I^2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A00Bh	SCI0	I^2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A00Ch	SCI0	I^2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A00Dh	SCI0	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A00Eh	SCI0	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A00Fh	SCI0	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A00Eh	SCI0	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A010h	SCI0	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A011h	SCI0	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A010h	SCI0	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A012h	SCI0	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A013h	SCI0	Data Comparison Control Register	DCCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A01Ah	SCI0	Comparison Data Register	CDR.H	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A01Bh	SCI0	Comparison Data Register	CDR.L	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A01Ah	SCI0	Comparison Data Register	CDR	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A01Ch	SCI0	Serial Port Register	SPTR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A01Dh	SCI0	Transmit/Receive Timing Select Register	TMGR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A020h	SCI1	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A020h	SMCI1	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A021h	SCI1	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A022h	SCI1	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A022h	SMCI1	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A023h	SCI1	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A024h	SCI1	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A024h	SMCI1	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A025h	SCI1	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A026h	SCI1	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A026h	SMCI1	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A027h	SCI1	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A028h	SCI1	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A029h	SCI1	I^2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A02Ah	SCI1	I^2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A02Bh	SCI1	I^2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A02Ch	SCI1	I^2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A02Dh	SCI1	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A02Eh	SCI1	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A02Fh	SCI1	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A02Eh	SCI1	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A030h	SCI1	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A031h	SCI1	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A030h	SCI1	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A032h	SCI1	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 35

Table 5.1 List of I/O Registers (Address Order) (20 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 A033h	SCI1	Data Comparison Control Register	DCCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A03Ah	SCI1	Comparison Data Register	CDR.H	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A03Bh	SCI1	Comparison Data Register	CDR.L	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A03Ah	SCI1	Comparison Data Register	CDR	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A03Ch	SCI1	Serial Port Register	SPTR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A03Dh	SCI1	Transmit/Receive Timing Select Register	TMGR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A040h	SCI2	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A040h	SMCI2	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A041h	SCI2	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A042h	SCI2	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A042h	SMCI2	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A043h	SCI2	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A044h	SCI2	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A044h	SMCI2	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A045h	SCI2	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A046h	SCI2	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A046h	SMCI2	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A047h	SCI2	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A048h	SCI2	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A049h	SCI2	I^2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A04Ah	SCI2	I^2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A04Bh	SCI2	I^2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A04Ch	SCI2	I^2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A04Dh	SCI2	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A04Eh	SCI2	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A04Fh	SCI2	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A04Eh	SCI2	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A050h	SCI2	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A051h	SCI2	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A050h	SCI2	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A052h	SCI2	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A053h	SCI2	Data Comparison Control Register	DCCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A05Ah	SCI2	Comparison Data Register	CDR.H	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A05Bh	SCI2	Comparison Data Register	CDR.L	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A05Ah	SCI2	Comparison Data Register	CDR	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A05Ch	SCI2	Serial Port Register	SPTR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A05Dh	SCI2	Transmit/Receive Timing Select Register	TMGR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A060h	SCI3	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A060h	SMCI3	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A061h	SCI3	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A062h	SCI3	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A062h	SMCI3	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A063h	SCI3	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A064h	SCI3	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A064h	SMCI3	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A065h	SCI3	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A066h	SCI3	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A066h	SMCI3	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A067h	SCI3	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A068h	SCI3	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 35

Table 5.1 List of I/O Registers (Address Order) (21 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 A069h	SCI3	I²C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A06Ah	SCI3	I²C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A06Bh	SCI3	I²C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A06Ch	SCI3	I²C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A06Dh	SCI3	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A06Eh	SCI3	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A06Fh	SCI3	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A06Eh	SCI3	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A070h	SCI3	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A071h	SCI3	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A070h	SCI3	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A072h	SCI3	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A073h	SCI3	Data Comparison Control Register	DCCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A07Ah	SCI3	Comparison Data Register	CDR.H	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A07Bh	SCI3	Comparison Data Register	CDR.L	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A07Ah	SCI3	Comparison Data Register	CDR	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A07Ch	SCI3	Serial Port Register	SPTR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A07Dh	SCI3	Transmit/Receive Timing Select Register	TMGR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A080h	SCI4	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A080h	SMCI4	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A081h	SCI4	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A082h	SCI4	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A082h	SMCI4	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A083h	SCI4	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A084h	SCI4	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A084h	SMCI4	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A085h	SCI4	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A086h	SCI4	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A086h	SMCI4	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A087h	SCI4	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A088h	SCI4	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A089h	SCI4	I²C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A08Ah	SCI4	I²C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A08Bh	SCI4	I²C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A08Ch	SCI4	I²C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A08Dh	SCI4	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A08Eh	SCI4	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A08Fh	SCI4	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A08Eh	SCI4	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A090h	SCI4	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A091h	SCI4	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A090h	SCI4	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A092h	SCI4	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A093h	SCI4	Data Comparison Control Register	DCCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A09Ah	SCI4	Comparison Data Register	CDR.H	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A09Bh	SCI4	Comparison Data Register	CDR.L	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A09Ah	SCI4	Comparison Data Register	CDR	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A09Ch	SCI4	Serial Port Register	SPTR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A09Dh	SCI4	Transmit/Receive Timing Select Register	TMGR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0A0h	SCI5	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 35

Table 5.1 List of I/O Registers (Address Order) (22 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 A0A0h	SMCI5	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0A1h	SCI5	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0A2h	SCI5	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0A2h	SMCI5	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0A3h	SCI5	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0A4h	SCI5	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0A4h	SMCI5	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0A5h	SCI5	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0A6h	SCI5	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0A6h	SMCI5	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0A7h	SCI5	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0A8h	SCI5	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0A9h	SCI5	I^2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0AAh	SCI5	I^2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0ABh	SCI5	I^2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0ACh	SCI5	I^2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0ADh	SCI5	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0AEh	SCI5	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0AFh	SCI5	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0AEh	SCI5	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A0B0h	SCI5	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0B1h	SCI5	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0B0h	SCI5	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A0B2h	SCI5	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0B3h	SCI5	Data Comparison Control Register	DCCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0BAh	SCI5	Comparison Data Register	CDR.H	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0BBh	SCI5	Comparison Data Register	CDR.L	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0BAh	SCI5	Comparison Data Register	CDR	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A0BCh	SCI5	Serial Port Register	SPTR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0BDh	SCI5	Transmit/Receive Timing Select Register	TMGR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0C0h	SCI6	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0C0h	SMCI6	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0C1h	SCI6	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0C2h	SCI6	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0C2h	SMCI6	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0C3h	SCI6	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0C4h	SCI6	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0C4h	SMCI6	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0C5h	SCI6	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0C6h	SCI6	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0C6h	SMCI6	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0C7h	SCI6	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0C8h	SCI6	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0C9h	SCI6	I^2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0CAh	SCI6	I^2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0CBh	SCI6	I^2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0CCh	SCI6	I^2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0CDh	SCI6	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0CEh	SCI6	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0CFh	SCI6	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 35

Table 5.1 List of I/O Registers (Address Order) (23 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 A0CEh	SCI6	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A0D0h	SCI6	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0D1h	SCI6	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0D0h	SCI6	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A0D2h	SCI6	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0D3h	SCI6	Data Comparison Control Register	DCCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0DAh	SCI6	Comparison Data Register	CDR.H	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0DBh	SCI6	Comparison Data Register	CDR.L	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0DAh	SCI6	Comparison Data Register	CDR	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A0DCh	SCI6	Serial Port Register	SPTR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0DDh	SCI6	Transmit/Receive Timing Select Register	TMGR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0E0h	SCI7	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0E0h	SMCI7	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0E1h	SCI7	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0E2h	SCI7	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0E2h	SMCI7	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0E3h	SCI7	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0E4h	SCI7	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0E4h	SMCI7	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0E5h	SCI7	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0E6h	SCI7	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0E6h	SMCI7	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0E7h	SCI7	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0E8h	SCI7	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0E9h	SCI7	I^2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0EAh	SCI7	I^2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0EBh	SCI7	I^2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0ECh	SCI7	I^2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0EDh	SCI7	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0EEh	SCI7	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0EFh	SCI7	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0EEh	SCI7	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A0F0h	SCI7	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0F1h	SCI7	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0F0h	SCI7	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A0F2h	SCI7	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0F3h	SCI7	Data Comparison Control Register	DCCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0FAh	SCI7	Comparison Data Register	CDR.H	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0FBh	SCI7	Comparison Data Register	CDR.L	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0FAh	SCI7	Comparison Data Register	CDR	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A0FCh	SCI7	Serial Port Register	SPTR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A0FDh	SCI7	Transmit/Receive Timing Select Register	TMGR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A100h	SCI8	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A100h	SMCI8	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A101h	SCI8	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A102h	SCI8	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A102h	SMCI8	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A103h	SCI8	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A104h	SCI8	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A104h	SMCI8	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 35

Table 5.1 List of I/O Registers (Address Order) (24 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 A105h	SCI8	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A106h	SCI8	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A106h	SMCI8	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A107h	SCI8	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A108h	SCI8	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A109h	SCI8	I²C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A10Ah	SCI8	I²C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A10Bh	SCI8	I²C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A10Ch	SCI8	I²C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A10Dh	SCI8	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A10Eh	SCI8	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A10Fh	SCI8	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A10Eh	SCI8	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A110h	SCI8	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A111h	SCI8	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A110h	SCI8	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A112h	SCI8	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A113h	SCI8	Data Comparison Control Register	DCCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A11Ah	SCI8	Comparison Data Register	CDR.H	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A11Bh	SCI8	Comparison Data Register	CDR.L	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A11Ah	SCI8	Comparison Data Register	CDR	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A11Ch	SCI8	Serial Port Register	SPTR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A11Dh	SCI8	Transmit/Receive Timing Select Register	TMGR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A120h	SCI9	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A120h	SMCI9	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A121h	SCI9	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A122h	SCI9	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A122h	SMCI9	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A123h	SCI9	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A124h	SCI9	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A124h	SMCI9	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A125h	SCI9	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A126h	SCI9	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A126h	SMCI9	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A127h	SCI9	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A128h	SCI9	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A129h	SCI9	I²C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A12Ah	SCI9	I²C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A12Bh	SCI9	I²C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A12Ch	SCI9	I²C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A12Dh	SCI9	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A12Eh	SCI9	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A12Fh	SCI9	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A12Eh	SCI9	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A130h	SCI9	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A131h	SCI9	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A130h	SCI9	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A132h	SCI9	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A133h	SCI9	Data Comparison Control Register	DCCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A13Ah	SCI9	Comparison Data Register	CDR.H	8	8	2, 3 PCLKB	2 ICLK	section 35

Table 5.1 List of I/O Registers (Address Order) (25 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 A13Bh	SCI9	Comparison Data Register	CDR.L	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A13Ah	SCI9	Comparison Data Register	CDR	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 A13Ch	SCI9	Serial Port Register	SPTR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A13Dh	SCI9	Transmit/Receive Timing Select Register	TMGR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 A500h	SSIE0	Control Register	SSICR	32	32	2, 3 PCLKB	2 ICLK	section 45
0008 A504h	SSIE0	Status Register	SSISR	32	32	2, 3 PCLKB	2 ICLK	section 45
0008 A510h	SSIE0	FIFO Control Register	SSIFCR	32	32	2, 3 PCLKB	2 ICLK	section 45
0008 A514h	SSIE0	FIFO Status Register	SSIFSR	32	32	2, 3 PCLKB	2 ICLK	section 45
0008 A518h	SSIE0	Transmit FIFO Data Register	SSIFTDR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	section 45
0008 A51Ch	SSIE0	Receive FIFO Data Register	SSIFRDR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	section 45
0008 A520h	SSIE0	Audio Format Register	SSIOFR	32	32	2, 3 PCLKB	2 ICLK	section 45
0008 A524h	SSIE0	FIFO Status Control Register	SSISCR	32	32	2, 3 PCLKB	2 ICLK	section 45
0008 AC00h	SDHI	Command Register	SDCMD	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 AC08h	SDHI	Argument Register	SDARG	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 AC10h	SDHI	Data Stop Register	SDSTOP	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 AC14h	SDHI	Block Count Register	SDBLKCNT	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 AC18h	SDHI	Response Register 10	SDRSP10	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 AC20h	SDHI	Response Register 32	SDRSP32	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 AC28h	SDHI	Response Register 54	SDRSP54	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 AC30h	SDHI	Response Register 76	SDRSP76	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 AC38h	SDHI	SD Status Register 1	SDSTS1	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 AC3Ch	SDHI	SD Status Register 2	SDSTS2	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 AC40h	SDHI	SD Interrupt Mask Register 1	SDIMSK1	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 AC44h	SDHI	SD Interrupt Mask Register 2	SDIMSK2	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 AC48h	SDHI	SDHI Clock Control Register	SDCLKCR	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 AC4Ch	SDHI	Transfer Data Size Register	SDSIZE	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 AC50h	SDHI	Card Access Option Register	SDOPT	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 AC58h	SDHI	SD Error Status Register 1	SDERSTS1	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 AC5Ch	SDHI	SD Error Status Register 2	SDERSTS2	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 AC60h	SDHI	SD Buffer Register	SDBUFR	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 AC68h	SDHI	SDIO Mode Control Register	SDIOMD	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 AC6Ch	SDHI	SDIO Status Register	SDIOSTS	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 AC70h	SDHI	SDIO Interrupt Mask Register	SDIOIMSK	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 ADB0h	SDHI	DMA Transfer Enable Register	SDDMAEN	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 ADC0h	SDHI	SDHI Software Reset Register	SDRST	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 ADC4h	SDHI	Version Register	SDVER	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 ADE0h	SDHI	Swap Control Register	SDSWAP	32	32	2, 3 PCLKB	2 ICLK	section 44
0008 B000h	CAC	CAC Control Register 0	CACR0	8	8	2, 3 PCLKB	2 ICLK	section 10
0008 B001h	CAC	CAC Control Register 1	CACR1	8	8	2, 3 PCLKB	2 ICLK	section 10
0008 B002h	CAC	CAC Control Register 2	CACR2	8	8	2, 3 PCLKB	2 ICLK	section 10
0008 B003h	CAC	CAC Interrupt Request Enable Register	CAICR	8	8	2, 3 PCLKB	2 ICLK	section 10
0008 B004h	CAC	CAC Status Register	CASTR	8	8	2, 3 PCLKB	2 ICLK	section 10
0008 B006h	CAC	CAC Upper-Limit Value Setting Register	CAULVR	16	16	2, 3 PCLKB	2 ICLK	section 10
0008 B008h	CAC	CAC Lower-Limit Value Setting Register	CALLVR	16	16	2, 3 PCLKB	2 ICLK	section 10
0008 B00Ah	CAC	CAC Counter Buffer Register	CACNTBR	16	16	2, 3 PCLKB	2 ICLK	section 10
0008 B100h	ELC	Event Link Control Register	ELCR	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B101h	ELC	Event Link Setting Register 0	ELSR0	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B104h	ELC	Event Link Setting Register 3	ELSR3	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B105h	ELC	Event Link Setting Register 4	ELSR4	8	8	2, 3 PCLKB	2 ICLK	section 21

Table 5.1 List of I/O Registers (Address Order) (26 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 B108h	ELC	Event Link Setting Register 7	ELSR7	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B10Bh	ELC	Event Link Setting Register 10	ELSR10	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B10Ch	ELC	Event Link Setting Register 11	ELSR11	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B10Dh	ELC	Event Link Setting Register 12	ELSR12	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B10Eh	ELC	Event Link Setting Register 13	ELSR13	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B10Fh	ELC	Event Link Setting Register 14	ELSR14	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B110h	ELC	Event Link Setting Register 15	ELSR15	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B113h	ELC	Event Link Setting Register 18	ELSR18	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B114h	ELC	Event Link Setting Register 19	ELSR19	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B115h	ELC	Event Link Setting Register 20	ELSR20	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B116h	ELC	Event Link Setting Register 21	ELSR21	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B117h	ELC	Event Link Setting Register 22	ELSR22	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B118h	ELC	Event Link Setting Register 23	ELSR23	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B119h	ELC	Event Link Setting Register 24	ELSR24	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B11Ah	ELC	Event Link Setting Register 25	ELSR25	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B11Bh	ELC	Event Link Setting Register 26	ELSR26	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B11Ch	ELC	Event Link Setting Register 27	ELSR27	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B11Dh	ELC	Event Link Setting Register 28	ELSR28	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B11Fh	ELC	Event Link Option Setting Register A	ELOPA	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B120h	ELC	Event Link Option Setting Register B	ELOPB	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B121h	ELC	Event Link Option Setting Register C	ELOPC	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B122h	ELC	Event Link Option Setting Register D	ELOPD	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B123h	ELC	Port Group Setting Register 1	PGR1	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B124h	ELC	Port Group Setting Register 2	PGR2	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B125h	ELC	Port Group Control Register 1	PGC1	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B126h	ELC	Port Group Control Register 2	PGC2	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B127h	ELC	Port Buffer Register 1	PDBF1	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B128h	ELC	Port Buffer Register 2	PDBF2	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B129h	ELC	Event Link Port Setting Register 0	PEL0	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B12Ah	ELC	Event Link Port Setting Register 1	PEL1	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B12Bh	ELC	Event Link Port Setting Register 2	PEL2	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B12Ch	ELC	Event Link Port Setting Register 3	PEL3	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B12Dh	ELC	Event Link Software Event Generation Register	ELSEGR	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B131h	ELC	Event Link Setting Register 33	ELSR33	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B133h	ELC	Event Link Setting Register 35	ELSR35	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B134h	ELC	Event Link Setting Register 36	ELSR36	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B135h	ELC	Event Link Setting Register 37	ELSR37	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B136h	ELC	Event Link Setting Register 38	ELSR38	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B13Dh	ELC	Event Link Setting Register 45	ELSR45	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B13Fh	ELC	Event Link Option Setting Register F	ELOPF	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B141h	ELC	Event Link Option Setting Register H	ELOPH	8	8	2, 3 PCLKB	2 ICLK	section 21
0008 B300h	SCI12	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B300h	SMCI12	Serial Mode Register	SMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B301h	SCI12	Bit Rate Register	BRR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B302h	SCI12	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B302h	SMCI12	Serial Control Register	SCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B303h	SCI12	Transmit Data Register	TDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B304h	SCI12	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B304h	SMCI12	Serial Status Register	SSR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B305h	SCI12	Receive Data Register	RDR	8	8	2, 3 PCLKB	2 ICLK	section 35

Table 5.1 List of I/O Registers (Address Order) (27 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 B306h	SCI12	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B306h	SMCI12	Smart Card Mode Register	SCMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B307h	SCI12	Serial Extended Mode Register	SEMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B308h	SCI12	Noise Filter Setting Register	SNFR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B309h	SCI12	I2C Mode Register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B30Ah	SCI12	I2C Mode Register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B30Bh	SCI12	I2C Mode Register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B30Ch	SCI12	I2C Status Register	SISR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B30Dh	SCI12	SPI Mode Register	SPMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B30Eh	SCI12	Transmit Data Register H	TDRH	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B30Fh	SCI12	Transmit Data Register L	TDRL	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B30Eh	SCI12	Transmit Data Register HL	TDRHL	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 B310h	SCI12	Receive Data Register H	RDRH	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B311h	SCI12	Receive Data Register L	RDRL	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B310h	SCI12	Receive Data Register HL	RDRHL	16	16	4, 5 PCLKB	2 ICLK	section 35
0008 B312h	SCI12	Modulation Duty Register	MDDR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B320h	SCI12	Extended Serial Module Enable Register	ESMER	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B321h	SCI12	Control Register 0	CR0	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B322h	SCI12	Control Register 1	CR1	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B323h	SCI12	Control Register 2	CR2	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B324h	SCI12	Control Register 3	CR3	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B325h	SCI12	Port Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B326h	SCI12	Interrupt Control Register	ICR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B327h	SCI12	Status Register	STR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B328h	SCI12	Status Clear Register	STCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B329h	SCI12	Control Field 0 Data Register	CF0DR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B32Ah	SCI12	Control Field 0 Compare Enable Register	CF0CR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B32Bh	SCI12	Control Field 0 Receive Data Register	CF0RR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B32Ch	SCI12	Primary Control Field 1 Data Register	PCF1DR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B32Dh	SCI12	Secondary Control Field 1 Data Register	SCF1DR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B32Eh	SCI12	Control Field 1 Compare Enable Register	CF1CR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B32Fh	SCI12	Control Field 1 Receive Data Register	CF1RR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B330h	SCI12	Timer Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B331h	SCI12	Timer Mode Register	TMR	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B332h	SCI12	Timer Prescaler Register	TPRE	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 B333h	SCI12	Timer Count Register	TCNT	8	8	2, 3 PCLKB	2 ICLK	section 35
0008 C000h	PORT0	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C001h	PORT1	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C002h	PORT2	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C003h	PORT3	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C004h	PORT4	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C005h	PORT5	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C006h	PORT6	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C007h	PORT7	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C008h	PORT8	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C009h	PORT9	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C00Ah	PORTA	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C00Bh	PORTB	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C00Ch	PORTC	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C00Dh	PORTD	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 22

Table 5.1 List of I/O Registers (Address Order) (28 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 C00Eh	PORTE	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C00Fh	PORTF	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C011h	PORTH	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C012h	PORTJ	Port Direction Register	PDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C020h	PORT0	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C021h	PORT1	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C022h	PORT2	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C023h	PORT3	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C024h	PORT4	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C025h	PORT5	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C026h	PORT6	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C027h	PORT7	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C028h	PORT8	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C029h	PORT9	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C02Ah	PORTA	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C02Bh	PORTB	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C02Ch	PORTC	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C02Dh	PORTD	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C02Eh	PORTE	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C02Fh	PORTF	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C031h	PORTH	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C032h	PORTJ	Port Output Data Register	PODR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C040h	PORT0	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C041h	PORT1	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C042h	PORT2	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C043h	PORT3	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C044h	PORT4	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C045h	PORT5	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C046h	PORT6	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C047h	PORT7	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C048h	PORT8	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C049h	PORT9	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C04Ah	PORTA	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C04Bh	PORTB	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C04Ch	PORTC	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C04Dh	PORTD	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C04Eh	PORTE	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C04Fh	PORTF	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C051h	PORTH	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C052h	PORTJ	Port Input Register	PIDR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C060h	PORT0	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C061h	PORT1	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C062h	PORT2	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C063h	PORT3	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C064h	PORT4	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C065h	PORT5	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C066h	PORT6	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C067h	PORT7	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C068h	PORT8	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C069h	PORT9	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 22

Table 5.1 List of I/O Registers (Address Order) (29 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 C06Ah	PORTA	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C06Bh	PORTB	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C06Ch	PORTC	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C06Dh	PORTD	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C06Eh	PORTE	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C06Fh	PORTF	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C071h	PORTH	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C072h	PORTJ	Port Mode Register	PMR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C080h	PORT0	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C081h	PORT0	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C082h	PORT1	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C083h	PORT1	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C084h	PORT2	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C085h	PORT2	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C086h	PORT3	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C087h	PORT3	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C088h	PORT4	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C089h	PORT4	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C08Ah	PORT5	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C08Bh	PORT5	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C08Ch	PORT6	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C08Dh	PORT6	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C08Eh	PORT7	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C08Fh	PORT7	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C090h	PORT8	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C091h	PORT8	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C092h	PORT9	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C094h	PORTA	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C095h	PORTA	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C096h	PORTB	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C097h	PORTB	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C098h	PORTC	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C099h	PORTC	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C09Ah	PORTD	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C09Bh	PORTD	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C09Ch	PORTE	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C09Dh	PORTE	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C09Fh	PORTF	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0A2h	PORTH	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0A4h	PORTJ	Open-Drain Control Register 0	ODR0	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0A5h	PORTJ	Open-Drain Control Register 1	ODR1	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0C0h	PORT0	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0C1h	PORT1	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0C2h	PORT2	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0C3h	PORT3	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0C4h	PORT4	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0C5h	PORT5	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0C6h	PORT6	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0C7h	PORT7	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0C8h	PORT8	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 22

Table 5.1 List of I/O Registers (Address Order) (30 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 C0C9h	PORT9	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0CAh	PORTA	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0CBh	PORTB	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0CCh	PORTC	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0CDh	PORTD	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0CEh	PORTE	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0CFh	PORTF	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0D1h	PORTH	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0D2h	PORTJ	Pull-Up Resistor Control Register	PCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0E0h	PORT0	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0E1h	PORT1	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0E2h	PORT2	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0E5h	PORT5	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0E7h	PORT7	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0E8h	PORT8	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0E9h	PORT9	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0EAh	PORTA	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0EBh	PORTB	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0ECh	PORTC	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0EDh	PORTD	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0EEh	PORTE	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C0F1h	PORTH	Drive Capacity Control Register	DSCR	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C100h	MPC	CS Output Enable Register	PFCSE	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C102h	MPC	CS Output Pin Select Register 0	PFCSS0	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C103h	MPC	CS Output Pin Select Register 1	PFCSS1	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C104h	MPC	Address Output Enable Register 0	PFAOE0	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C105h	MPC	Address Output Enable Register 1	PFAOE1	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C106h	MPC	External Bus Control Register 0	PFBCR0	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C107h	MPC	External Bus Control Register 1	PFBCR1	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C108h	MPC	External Bus Control Register 2	PFBCR2	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C109h	MPC	External Bus Control Register 3	PFBCR3	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C11Fh	MPC	Write-Protect Register	PWPR	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C128h	PORT0	Drive Capacity Control Register 2	DSCR2	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C129h	PORT1	Drive Capacity Control Register 2	DSCR2	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C12Ah	PORT2	Drive Capacity Control Register 2	DSCR2	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C12Bh	PORT3	Drive Capacity Control Register 2	DSCR2	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C12Dh	PORT5	Drive Capacity Control Register 2	DSCR2	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C12Fh	PORT7	Drive Capacity Control Register 2	DSCR2	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C130h	PORT8	Drive Capacity Control Register 2	DSCR2	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C131h	PORT9	Drive Capacity Control Register 2	DSCR2	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C132h	PORTA	Drive Capacity Control Register 2	DSCR2	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C133h	PORTB	Drive Capacity Control Register 2	DSCR2	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C134h	PORTC	Drive Capacity Control Register 2	DSCR2	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C135h	PORTD	Drive Capacity Control Register 2	DSCR2	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C136h	PORTE	Drive Capacity Control Register 2	DSCR2	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C139h	PORTH	Drive Capacity Control Register 2	DSCR2	8	8	2, 3 PCLKB	2 ICLK	section 22
0008 C140h	MPC	P00 Pin Function Control Register	P00PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C141h	MPC	P01 Pin Function Control Register	P01PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C142h	MPC	P02 Pin Function Control Register	P02PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C143h	MPC	P03 Pin Function Control Register	P03PFS	8	8	2, 3 PCLKB	2 ICLK	section 23

Table 5.1 List of I/O Registers (Address Order) (31 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 C145h	MPC	P05 Pin Function Control Register	P05PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C147h	MPC	P07 Pin Function Control Register	P07PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C14Ah	MPC	P12 Pin Function Control Register	P12PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C14Bh	MPC	P13 Pin Function Control Register	P13PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C14Ch	MPC	P14 Pin Function Control Register	P14PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C14Dh	MPC	P15 Pin Function Control Register	P15PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C14Eh	MPC	P16 Pin Function Control Register	P16PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C14Fh	MPC	P17 Pin Function Control Register	P17PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C150h	MPC	P20 Pin Function Control Register	P20PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C151h	MPC	P21 Pin Function Control Register	P21PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C152h	MPC	P22 Pin Function Control Register	P22PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C153h	MPC	P23 Pin Function Control Register	P23PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C154h	MPC	P24 Pin Function Control Register	P24PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C155h	MPC	P25 Pin Function Control Register	P25PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C156h	MPC	P26 Pin Function Control Register	P26PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C157h	MPC	P27 Pin Function Control Register	P27PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C158h	MPC	P30 Pin Function Control Register	P30PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C159h	MPC	P31 Pin Function Control Register	P31PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C15Ah	MPC	P32 Pin Function Control Register	P32PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C15Bh	MPC	P33 Pin Function Control Register	P33PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C15Ch	MPC	P34 Pin Function Control Register	P34PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C160h	MPC	P40 Pin Function Control Register	P40PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C161h	MPC	P41 Pin Function Control Register	P41PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C162h	MPC	P42 Pin Function Control Register	P42PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C163h	MPC	P43 Pin Function Control Register	P43PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C164h	MPC	P44 Pin Function Control Register	P44PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C165h	MPC	P45 Pin Function Control Register	P45PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C166h	MPC	P46 Pin Function Control Register	P46PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C167h	MPC	P47 Pin Function Control Register	P47PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C168h	MPC	P50 Pin Function Control Register	P50PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C169h	MPC	P51 Pin Function Control Register	P51PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C16Ah	MPC	P52 Pin Function Control Register	P52PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C16Bh	MPC	P53 Pin Function Control Register	P53PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C16Ch	MPC	P54 Pin Function Control Register	P54PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C16Dh	MPC	P55 Pin Function Control Register	P55PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C16Eh	MPC	P56 Pin Function Control Register	P56PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C170h	MPC	P60 Pin Function Control Register	P60PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C171h	MPC	P61 Pin Function Control Register	P61PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C172h	MPC	P62 Pin Function Control Register	P62PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C173h	MPC	P63 Pin Function Control Register	P63PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C174h	MPC	P64 Pin Function Control Register	P64PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C175h	MPC	P65 Pin Function Control Register	P65PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C176h	MPC	P66 Pin Function Control Register	P66PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C177h	MPC	P67 Pin Function Control Register	P67PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C178h	MPC	P70 Pin Function Control Register	P70PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C179h	MPC	P71 Pin Function Control Register	P71PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C17Ah	MPC	P72 Pin Function Control Register	P72PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C17Bh	MPC	P73 Pin Function Control Register	P73PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C17Ch	MPC	P74 Pin Function Control Register	P74PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C17Dh	MPC	P75 Pin Function Control Register	P75PFS	8	8	2, 3 PCLKB	2 ICLK	section 23

Table 5.1 List of I/O Registers (Address Order) (32 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 C17Eh	MPC	P76 Pin Function Control Register	P76PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C17Fh	MPC	P77 Pin Function Control Register	P77PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C180h	MPC	P80 Pin Function Control Register	P80PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C181h	MPC	P81 Pin Function Control Register	P81PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C182h	MPC	P82 Pin Function Control Register	P82PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C183h	MPC	P83 Pin Function Control Register	P83PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C186h	MPC	P86 Pin Function Control Register	P86PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C187h	MPC	P87 Pin Function Control Register	P87PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C188h	MPC	P90 Pin Function Control Register	P90PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C189h	MPC	P91 Pin Function Control Register	P91PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C18Ah	MPC	P92 Pin Function Control Register	P92PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C18Bh	MPC	P93 Pin Function Control Register	P93PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C190h	MPC	PA0 Pin Function Control Register	PA0PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C191h	MPC	PA1 Pin Function Control Register	PA1PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C192h	MPC	PA2 Pin Function Control Register	PA2PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C193h	MPC	PA3 Pin Function Control Register	PA3PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C194h	MPC	PA4 Pin Function Control Register	PA4PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C195h	MPC	PA5 Pin Function Control Register	PA5PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C196h	MPC	PA6 Pin Function Control Register	PA6PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C197h	MPC	PA7 Pin Function Control Register	PA7PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C198h	MPC	PB0 Pin Function Control Register	PB0PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C199h	MPC	PB1 Pin Function Control Register	PB1PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C19Ah	MPC	PB2 Pin Function Control Register	PB2PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C19Bh	MPC	PB3 Pin Function Control Register	PB3PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C19Ch	MPC	PB4 Pin Function Control Register	PB4PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C19Dh	MPC	PB5 Pin Function Control Register	PB5PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C19Eh	MPC	PB6 Pin Function Control Register	PB6PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C19Fh	MPC	PB7 Pin Function Control Register	PB7PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1A0h	MPC	PC0 Pin Function Control Register	PC0PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1A1h	MPC	PC1 Pin Function Control Register	PC1PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1A2h	MPC	PC2 Pin Function Control Register	PC2PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1A3h	MPC	PC3 Pin Function Control Register	PC3PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1A4h	MPC	PC4 Pin Function Control Register	PC4PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1A5h	MPC	PC5 Pin Function Control Register	PC5PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1A6h	MPC	PC6 Pin Function Control Register	PC6PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1A7h	MPC	PC7 Pin Function Control Register	PC7PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1A8h	MPC	PD0 Pin Function Control Register	PD0PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1A9h	MPC	PD1 Pin Function Control Register	PD1PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1AAh	MPC	PD2 Pin Function Control Register	PD2PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1ABh	MPC	PD3 Pin Function Control Register	PD3PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1ACh	MPC	PD4 Pin Function Control Register	PD4PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1ADh	MPC	PD5 Pin Function Control Register	PD5PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1AEh	MPC	PD6 Pin Function Control Register	PD6PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1AFh	MPC	PD7 Pin Function Control Register	PD7PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1B0h	MPC	PE0 Pin Function Control Register	PE0PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1B1h	MPC	PE1 Pin Function Control Register	PE1PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1B2h	MPC	PE2 Pin Function Control Register	PE2PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1B3h	MPC	PE3 Pin Function Control Register	PE3PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1B4h	MPC	PE4 Pin Function Control Register	PE4PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1B5h	MPC	PE5 Pin Function Control Register	PE5PFS	8	8	2, 3 PCLKB	2 ICLK	section 23

Table 5.1 List of I/O Registers (Address Order) (33 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 C1B6h	MPC	PE6 Pin Function Control Register	PE6PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1B7h	MPC	PE7 Pin Function Control Register	PE7PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1BDh	MPC	PF5 Pin Function Control Register	PF5PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1C9h	MPC	PH1 Pin Function Control Register	PH1PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1CAh	MPC	PH2 Pin Function Control Register	PH2PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1D3h	MPC	PJ3 Pin Function Control Register	PJ3PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
0008 C1D5h	MPC	PJ5 Pin Function Control Register	PJ5PFS	8	8	2, 3 PCLKB	2 ICLK	section 23
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	section 11
0008 C283h	SYSTEM	Deep Standby Interrupt Enable Register 1	DPSIER1	8	8	4, 5 PCLKB	2, 3 ICLK	section 11
0008 C284h	SYSTEM	Deep Standby Interrupt Enable Register 2	DPSIER2	8	8	4, 5 PCLKB	2, 3 ICLK	section 11
0008 C285h	SYSTEM	Deep Standby Interrupt Enable Register 3	DPSIER3	8	8	4, 5 PCLKB	2, 3 ICLK	section 11
0008 C286h	SYSTEM	Deep Standby Interrupt Flag Register 0	DPSIFR0	8	8	4, 5 PCLKB	2, 3 ICLK	section 11
0008 C287h	SYSTEM	Deep Standby Interrupt Flag Register 1	DPSIFR1	8	8	4, 5 PCLKB	2, 3 ICLK	section 11
0008 C288h	SYSTEM	Deep Standby Interrupt Flag Register 2	DPSIFR2	8	8	4, 5 PCLKB	2, 3 ICLK	section 11
0008 C289h	SYSTEM	Deep Standby Interrupt Flag Register 3	DPSIFR3	8	8	4, 5 PCLKB	2, 3 ICLK	section 11
0008 C28Ah	SYSTEM	Deep Standby Interrupt Edge Register 0	DPSIEGR0	8	8	4, 5 PCLKB	2, 3 ICLK	section 11
0008 C28Bh	SYSTEM	Deep Standby Interrupt Edge Register 1	DPSIEGR1	8	8	4, 5 PCLKB	2, 3 ICLK	section 11
0008 C28Ch	SYSTEM	Deep Standby Interrupt Edge Register 2	DPSIEGR2	8	8	4, 5 PCLKB	2, 3 ICLK	section 11
0008 C28Dh	SYSTEM	Deep Standby Interrupt Edge Register 3	DPSIEGR3	8	8	4, 5 PCLKB	2, 3 ICLK	section 11
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 Supply Control Register	HOCOPCR	8	8	4, 5 PCLKB	2, 3 ICLK	section 9
0008 C296h	FLASH	Flash P/E Protect Register	FWEPROR	8	8	2 ICLK		section 55
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	section 8
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	section 8
0008 C29Ch	REMC0	Remote Control Signal Receive Pin Control Register	REMPC	8	8	4, 5 PCLKB	2, 3 ICLK	section 46
0008 C400h	RTC	64-Hz Counter	R64CNT	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C402h	RTC	Second Counter	RSECCNT	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C402h	RTC	Binary Counter 0	BCNT0	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C404h	RTC	Minute Counter	RMINCNT	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C404h	RTC	Binary Counter 1	BCNT1	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C406h	RTC	Hour Counter	RHRCNT	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C406h	RTC	Binary Counter 2	BCNT2	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C408h	RTC	Day-of-Week Counter	RWKCNT	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C408h	RTC	Binary Counter 3	BCNT3	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C40Ah	RTC	Date Counter	RDAYCNT	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C40Ch	RTC	Month Counter	RMONCNT	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C40Eh	RTC	Year Counter	RYRCNT	16	16	2, 3 PCLKB	2 ICLK	section 31
0008 C410h	RTC	Second Alarm Register	RSECAR	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C410h	RTC	Binary Counter 0 Alarm Register	BCNT0AR	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C412h	RTC	Minute Alarm Register	RMINAR	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C412h	RTC	Binary Counter 1 Alarm Register	BCNT1AR	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C414h	RTC	Hour Alarm Register	RHRAR	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C414h	RTC	Binary Counter 2 Alarm Register	BCNT2AR	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C416h	RTC	Day-of-Week Alarm Register	RWKAR	8	8	2, 3 PCLKB	2 ICLK	section 31

Table 5.1 List of I/O Registers (Address Order) (34 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 C416h	RTC	Binary Counter 3 Alarm Register	BCNT3AR	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C418h	RTC	Date Alarm Register	RDAYAR	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C418h	RTC	Binary Counter 0 Alarm Enable Register	BCNT0AER	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C41Ah	RTC	Month Alarm Register	RMONAR	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C41Ah	RTC	Binary Counter 1 Alarm Enable Register	BCNT1AER	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C41Ch	RTC	Year Alarm Register	RYRAR	16	16	2, 3 PCLKB	2 ICLK	section 31
0008 C41Ch	RTC	Binary Counter 2 Alarm Enable Register	BCNT2AER	16	16	2, 3 PCLKB	2 ICLK	section 31
0008 C41Eh	RTC	Year Alarm Enable Register	RYRAREN	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C41Eh	RTC	Binary Counter 3 Alarm Enable Register	BCNT3AER	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C422h	RTC	RTC Control Register 1	RCR1	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C424h	RTC	RTC Control Register 2	RCR2	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C426h	RTC	RTC Control Register 3	RCR3	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C428h	RTC	RTC Control Register 4	RCR4	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C42Ah	RTC	Frequency Register H	RFRH	16	16	2, 3 PCLKB	2 ICLK	section 31
0008 C42Ch	RTC	Frequency Register L	RFRL	16	16	2, 3 PCLKB	2 ICLK	section 31
0008 C42Eh	RTC	Time Error Adjustment Register	RADJ	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C440h	RTC	Time Capture Control Register 0	RTCCR0	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C442h	RTC	Time Capture Control Register 1	RTCCR1	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C444h	RTC	Time Capture Control Register 2	RTCCR2	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C452h	RTC	Second Capture Register 0	RSECCP0	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C452h	RTC	BCNT0 Capture Register 0	BCNT0CP0	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C454h	RTC	Minute Capture Register 0	RMINCP0	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C454h	RTC	BCNT1 Capture Register 0	BCNT1CP0	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C456h	RTC	Hour Capture Register 0	RHRCP0	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C456h	RTC	BCNT2 Capture Register 0	BCNT2CP0	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C45Ah	RTC	Date Capture Register 0	RDAYCP0	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C45Ah	RTC	BCNT3 Capture Register 0	BCNT3CP0	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C45Ch	RTC	Month Capture Register 0	RMONCP0	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C462h	RTC	Second Capture Register 1	RSECCP1	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C462h	RTC	BCNT0 Capture Register 1	BCNT0CP1	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C464h	RTC	Minute Capture Register 1	RMINCP1	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C464h	RTC	BCNT1 Capture Register 1	BCNT1CP1	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C466h	RTC	Hour Capture Register 1	RHRCP1	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C466h	RTC	BCNT2 Capture Register 1	BCNT2CP1	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C46Ah	RTC	Date Capture Register 1	RDAYCP1	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C46Ah	RTC	BCNT3 Capture Register 1	BCNT3CP1	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C46Ch	RTC	Month Capture Register 1	RMONCP1	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C472h	RTC	Second Capture Register 2	RSECCP2	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C472h	RTC	BCNT0 Capture Register 2	BCNT0CP2	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C474h	RTC	Minute Capture Register 2	RMINCP2	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C474h	RTC	BCNT1 Capture Register 2	BCNT1CP2	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C476h	RTC	Hour Capture Register 2	RHRCP2	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C476h	RTC	BCNT2 Capture Register 2	BCNT2CP2	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C47Ah	RTC	Date Capture Register 2	RDAYCP2	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C47Ah	RTC	BCNT3 Capture Register 2	BCNT3CP2	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C47Ch	RTC	Month Capture Register 2	RMONCP2	8	8	2, 3 PCLKB	2 ICLK	section 31
0008 C4C0h	POE3	Input Level Control/Status Register 1	ICSR1	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 C4C2h	POE3	Output Level Control/Status Register 1	OCSR1	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 C4C4h	POE3	Input Level Control/Status Register 2	ICSR2	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 C4C6h	POE3	Output Level Control/Status Register 2	OCSR2	16	16	2, 3 PCLKB	2 ICLK	section 25

Table 5.1 List of I/O Registers (Address Order) (35 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0008 C4C8h	POE3	Input Level Control/Status Register 3	ICSR3	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 C4CAh	POE3	Software Port Output Enable Register	SPOER	8	8	2, 3 PCLKB	2 ICLK	section 25
0008 C4CBh	POE3	Port Output Enable Control Register 1	POECR1	8	8	2, 3 PCLKB	2 ICLK	section 25
0008 C4CCh	POE3	Port Output Enable Control Register 2	POECR2	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 C4D0h	POE3	Port Output Enable Control Register 4	POECR4	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 C4D2h	POE3	Port Output Enable Control Register 5	POECR5	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 C4D6h	POE3	Input Level Control/Status Register 4	ICSR4	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 C4D8h	POE3	Input Level Control/Status Register 5	ICSR5	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 C4DAh	POE3	Active Level Setting Register 1	ALR1	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 C4DCh	POE3	Input Level Control/Status Register 6	ICSR6	16	16	2, 3 PCLKB	2 ICLK	section 25
0008 C4E4h	POE3	MTU0 Pin Select Register 1	M0SELR1	8	8	2, 3 PCLKB	2 ICLK	section 25
0008 C4E5h	POE3	MTU0 Pin Select Register 2	M0SELR2	8	8	2, 3 PCLKB	2 ICLK	section 25
0008 C4E6h	POE3	MTU3 Pin Select Register	M3SELR	8	8	2, 3 PCLKB	2 ICLK	section 25
0008 C4E7h	POE3	MTU4 Pin Select Register 1	M4SELR1	8	8	2, 3 PCLKB	2 ICLK	section 25
0008 C4E8h	POE3	MTU4 Pin Select Register 2	M4SELR2	8	8	2, 3 PCLKB	2 ICLK	section 25
0008 C500h	TEMPS	Temperature Sensor Control Register	TSCR	8	8	2, 3 PCLKB	2 ICLK	section 51
0008 CC00h	SYSTEM	Sub-Clock Oscillator Control Register 2	SOSCCR2	8	8	5, 6 PCLKB	1 to 3 ICLK	section 9
0008 CC01h	SYSTEM	Backup Domain Sub-Clock Control Register	BKSCCR	8	8	5, 6 PCLKB	1 to 3 ICLK	section 9
0008 CC46h	SYSTEM	Backup Domain Power Status Register	BKPSR	8	8	5, 6 PCLKB	1 to 3 ICLK	section 12
0008 CC48h	SYSTEM	Tamper Status Register	TAMPSR	8	8	5, 6 PCLKB	1 to 3 ICLK	section 12
0008 CC49h	SYSTEM	Tamper Control Register	TAMPCR	8	8	5, 6 PCLKB	1 to 3 ICLK	section 12
0008 CC4Ah	SYSTEM	Time Capture Event Control Register	TCECR	8	8	5, 6 PCLKB	1 to 3 ICLK	section 12
0008 CC4Ch	SYSTEM	Tamper/RTCIC Input Control Register 1	TAMPICR1	8	8	5, 6 PCLKB	1 to 3 ICLK	section 12
0008 CC4Dh	SYSTEM	Tamper/RTCIC Input Control Register 2	TAMPICR2	8	8	5, 6 PCLKB	1 to 3 ICLK	section 12
0008 CC4Eh	SYSTEM	Tamper/RTCIC Input Monitoring Register	TAMPIMR	8	8	5, 6 PCLKB	1 to 3 ICLK	section 12
0008 CE00h to 0008 CE7Fh	SYSTEM	Backup Register 0 to Backup Register 127	BKR0 to BKR127	8	8	5, 6 PCLKB	1 to 3 ICLK	section 12
0009 0200h to 0009 03FFh	CAN0	Mailbox Register 0 to Mailbox Register 31	MB0 to MB31	128	8, 16, 32*2	2, 3 PCLKB	2 ICLK	section 39
0009 0400h to 0009 041Fh	CAN0	Mask Register 0 to Mask Register 7	MKR0 to MKR7	32	8, 16, 32	2, 3 PCLKB	2 ICLK	section 39
0009 0420h	CAN0	FIFO Received ID Compare Register 0	FIDCR0	32	8, 16, 32	2, 3 PCLKB	2 ICLK	section 39
0009 0424h	CAN0	FIFO Received ID Compare Register 1	FIDCR1	32	8, 16, 32	2, 3 PCLKB	2 ICLK	section 39
0009 0428h	CAN0	Mask Invalid Register	MKIVLR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	section 39
0009 042Ch	CAN0	Mailbox Interrupt Enable Register	MIER	32	8, 16, 32	2, 3 PCLKB	2 ICLK	section 39
0009 0820h to 0009 083Fh	CAN0	Message Control Register 0 to Message Control Register 31	MCTL0 to MCTL31	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 0840h	CAN0	Control Register	CTLR	16	8, 16	2, 3 PCLKB	2 ICLK	section 39
0009 0842h	CAN0	Status Register	STR	16	8, 16	2, 3 PCLKB	2 ICLK	section 39
0009 0844h	CAN0	Bit Configuration Register	BCR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	section 39
0009 0848h	CAN0	Receive FIFO Control Register	RFCR	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 0849h	CAN0	Receive FIFO Pointer Control Register	RFPCR	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 084Ah	CAN0	Transmit FIFO Control Register	TFCR	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 084Bh	CAN0	Transmit FIFO Pointer Control Register	TFPCR	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 084Ch	CAN0	Error Interrupt Enable Register	EIER	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 084Dh	CAN0	Error Interrupt Factor Judge Register	EIFR	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 084Eh	CAN0	Receive Error Count Register	RECR	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 084Fh	CAN0	Transmit Error Count Register	TECR	8	8	2, 3 PCLKB	2 ICLK	section 39

Table 5.1 List of I/O Registers (Address Order) (36 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0009 0850h	CAN0	Error Code Store Register	ECSR	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 0851h	CAN0	Channel Search Support Register	CSSR	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 0852h	CAN0	Mailbox Search Status Register	MSSR	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 0853h	CAN0	Mailbox Search Mode Register	MSMR	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 0854h	CAN0	Time Stamp Register	TSR	16	16	2, 3 PCLKB	2 ICLK	section 39
0009 0856h	CAN0	Acceptance Filter Support Register	AFSR	16	16	2, 3 PCLKB	2 ICLK	section 39
0009 0858h	CAN0	Test Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 1200h to 0009 13FFh	CAN1	Mailbox Register 0 to Mailbox Register 31	MB0 to MB31	128	8, 16, 32*2	2, 3 PCLKB	2 ICLK	section 39
0009 1400h to 0009 141Fh	CAN1	Mask Register 0 to Mask Register 7	MKR0 to MKR7	32	8, 16, 32	2, 3 PCLKB	2 ICLK	section 39
0009 1420h	CAN1	FIFO Received ID Compare Register 0	FIDCR0	32	8, 16, 32	2, 3 PCLKB	2 ICLK	section 39
0009 1424h	CAN1	FIFO Received ID Compare Register 1	FIDCR1	32	8, 16, 32	2, 3 PCLKB	2 ICLK	section 39
0009 1428h	CAN1	Mask Invalid Register	MKIVLR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	section 39
0009 142Ch	CAN1	Mailbox Interrupt Enable Register	MIER	32	8, 16, 32	2, 3 PCLKB	2 ICLK	section 39
0009 1820h to 0009 183Fh	CAN1	Message Control Register 0 to Message Control Register 31	MCTL0 to MCTL31	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 1840h	CAN1	Control Register	CTLR	16	8, 16	2, 3 PCLKB	2 ICLK	section 39
0009 1842h	CAN1	Status Register	STR	16	8, 16	2, 3 PCLKB	2 ICLK	section 39
0009 1844h	CAN1	Bit Configuration Register	BCR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	section 39
0009 1848h	CAN1	Receive FIFO Control Register	RFCR	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 1849h	CAN1	Receive FIFO Pointer Control Register	RFPCR	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 184Ah	CAN1	Transmit FIFO Control Register	TFCR	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 184Bh	CAN1	Transmit FIFO Pointer Control Register	TFPCR	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 184Ch	CAN1	Error Interrupt Enable Register	EIER	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 184Dh	CAN1	Error Interrupt Factor Judge Register	EIFR	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 184Eh	CAN1	Receive Error Count Register	RECR	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 184Fh	CAN1	Transmit Error Count Register	TECR	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 1850h	CAN1	Error Code Store Register	ECSR	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 1851h	CAN1	Channel Search Support Register	CSSR	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 1852h	CAN1	Mailbox Search Status Register	MSSR	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 1853h	CAN1	Mailbox Search Mode Register	MSMR	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 1854h	CAN1	Time Stamp Register	TSR	16	16	2, 3 PCLKB	2 ICLK	section 39
0009 1856h	CAN1	Acceptance Filter Support Register	AFSR	16	16	2, 3 PCLKB	2 ICLK	section 39
0009 1858h	CAN1	Test Control Register	TCR	8	8	2, 3 PCLKB	2 ICLK	section 39
0009 4200h	CMTW0	Timer Start Register	CMWSTR	16	16	2, 3 PCLKB	2 ICLK	section 30
0009 4204h	CMTW0	Timer Control Register	CMWCR	16	16	2, 3 PCLKB	2 ICLK	section 30
0009 4208h	CMTW0	Timer I/O Control Register	CMWIOR	16	16	2, 3 PCLKB	2 ICLK	section 30
0009 4210h	CMTW0	Timer Counter	CMWCNT	32	32	2, 3 PCLKB	2 ICLK	section 30
0009 4214h	CMTW0	Compare Match Constant Register	CMWCOR	32	32	2, 3 PCLKB	2 ICLK	section 30
0009 4218h	CMTW0	Input Capture Register 0	CMWICR0	32	32	2, 3 PCLKB	2 ICLK	section 30
0009 421Ch	CMTW0	Input Capture Register 1	CMWICR1	32	32	2, 3 PCLKB	2 ICLK	section 30
0009 4220h	CMTW0	Output Compare Register 0	CMWOCR0	32	32	2, 3 PCLKB	2 ICLK	section 30
0009 4224h	CMTW0	Output Compare Register 1	CMWOCR1	32	32	2, 3 PCLKB	2 ICLK	section 30
0009 4280h	CMTW1	Timer Start Register	CMWSTR	16	16	2, 3 PCLKB	2 ICLK	section 30
0009 4284h	CMTW1	Timer Control Register	CMWCR	16	16	2, 3 PCLKB	2 ICLK	section 30
0009 4288h	CMTW1	Timer I/O Control Register	CMWIOR	16	16	2, 3 PCLKB	2 ICLK	section 30
0009 4290h	CMTW1	Timer Counter	CMWCNT	32	32	2, 3 PCLKB	2 ICLK	section 30
0009 4294h	CMTW1	Compare Match Constant Register	CMWCOR	32	32	2, 3 PCLKB	2 ICLK	section 30

Table 5.1 List of I/O Registers (Address Order) (37 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
0009 4298h	CMTW1	Input Capture Register 0	CMWICR0	32	32	2, 3 PCLKB	2 ICLK	section 30
0009 429Ch	CMTW1	Input Capture Register 1	CMWICR1	32	32	2, 3 PCLKB	2 ICLK	section 30
0009 42A0h	CMTW1	Output Compare Register 0	CMWOCR0	32	32	2, 3 PCLKB	2 ICLK	section 30
0009 42A4h	CMTW1	Output Compare Register 1	CMWOCR1	32	32	2, 3 PCLKB	2 ICLK	section 30
000A 0000h	USB0	System Configuration Control Register	SYSCFG	16	16	3, 4 PCLKB	2 ICLK	section 34
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)*1	section 34
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)*1	section 34
000A 0014h	USB0	CFIFO Port Register	CFIFO	16	8, 16	3, 4 PCLKB	2 ICLK	section 34
000A 0018h	USB0	D0FIFO Port Register	D0FIFO	16	8, 16	3, 4 PCLKB	2 ICLK	section 34
000A 001Ch	USB0	D1FIFO Port Register	D1FIFO	16	8, 16	3, 4 PCLKB	2 ICLK	section 34
000A 0020h	USB0	CFIFO Port Select Register	CFIFOSEL	16	16	3, 4 PCLKB	2 ICLK	section 34
000A 0022h	USB0	CFIFO Port Control Register	CFIFOCTR	16	16	3, 4 PCLKB	2 ICLK	section 34
000A 0028h	USB0	D0FIFO Port Select Register	D0FIFOSEL	16	16	3, 4 PCLKB	2 ICLK	section 34
000A 002Ah	USB0	D0FIFO Port Control Register	D0FIFOCTR	16	16	3, 4 PCLKB	2 ICLK	section 34
000A 002Ch	USB0	D1FIFO Port Select Register	D1FIFOSEL	16	16	3, 4 PCLKB	2 ICLK	section 34
000A 002Eh	USB0	D1FIFO Port Control Register	D1FIFOCTR	16	16	3, 4 PCLKB	2 ICLK	section 34
000A 0030h	USB0	Interrupt Enable Register 0	INTENB0	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0032h	USB0	Interrupt Enable Register 1	INTENB1	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0036h	USB0	BRDY Interrupt Enable Register	BRDYENB	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0038h	USB0	NRDY Interrupt Enable Register	NRDYENB	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 003Ah	USB0	BEMP Interrupt Enable Register	BEMPENB	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 003Ch	USB0	SOF Output Configuration Register	SOFCFG	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0040h	USB0	Interrupt Status Register 0	INTSTS0	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0042h	USB0	Interrupt Status Register 1	INTSTS1	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0046h	USB0	BRDY Interrupt Status Register	BRDYSTS	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34

Table 5.1 List of I/O Registers (Address Order) (38 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000A 0048h	USB0	NRDY Interrupt Status Register	NRDYSTS	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 004Ah	USB0	BEMP Interrupt Status Register	BEMPSTS	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 004Ch	USB0	Frame Number Register	FRMNUM	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 004Eh	USB0	Device State Change Register	DVCHGR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0050h	USB0	USB Address Register	USBADDR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0054h	USB0	USB Request Type Register	USBREQ	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0056h	USB0	USB Request Value Register	USBVAL	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0058h	USB0	USB Request Index Register	USBINDX	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 005Ah	USB0	USB Request Length Register	USBLENG	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 005Ch	USB0	DCP Configuration Register	DCPCFG	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 005Eh	USB0	DCP Maximum Packet Size Register	DCPMAXP	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0060h	USB0	DCP Control Register	DCPCTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0064h	USB0	Pipe Window Select Register	PIPESEL	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0068h	USB0	Pipe Configuration Register	PIPECFG	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 006Ch	USB0	Pipe Maximum Packet Size Register	PIPEMAXP	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 006Eh	USB0	Pipe Cycle Control Register	PIPEPERI	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34

Table 5.1 List of I/O Registers (Address Order) (39 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000A 0070h	USB0	Pipe 1 Control Register	PIPE1CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0072h	USB0	Pipe 2 Control Register	PIPE2CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0074h	USB0	Pipe 3 Control Register	PIPE3CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0076h	USB0	Pipe 4 Control Register	PIPE4CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0078h	USB0	Pipe 5 Control Register	PIPE5CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 007Ah	USB0	Pipe 6 Control Register	PIPE6CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 007Ch	USB0	Pipe 7 Control Register	PIPE7CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 007Eh	USB0	Pipe 8 Control Register	PIPE8CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0080h	USB0	Pipe 9 Control Register	PIPE9CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0090h	USB0	Pipe 1 Transaction Counter Enable Register	PIPE1TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0092h	USB0	Pipe 1 Transaction Counter Register	PIPE1TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0094h	USB0	Pipe 2 Transaction Counter Enable Register	PIPE2TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0096h	USB0	Pipe 2 Transaction Counter Register	PIPE2TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0098h	USB0	Pipe 3 Transaction Counter Enable Register	PIPE3TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 009Ah	USB0	Pipe 3 Transaction Counter Register	PIPE3TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 009Ch	USB0	Pipe 4 Transaction Counter Enable Register	PIPE4TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34

Table 5.1 List of I/O Registers (Address Order) (40 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000A 009Eh	USB0	Pipe 4 Transaction Counter Register	PIPE4TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 00A0h	USB0	Pipe 5 Transaction Counter Enable Register	PIPE5TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 00A2h	USB0	Pipe 5 Transaction Counter Register	PIPE5TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 00D0h	USB0	Device Address 0 Configuration Register	DEVADD0	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 00D2h	USB0	Device Address 1 Configuration Register	DEVADD1	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 00D4h	USB0	Device Address 2 Configuration Register	DEVADD2	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 00D6h	USB0	Device Address 3 Configuration Register	DEVADD3	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 00D8h	USB0	Device Address 4 Configuration Register	DEVADD4	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 00DAh	USB0	Device Address 5 Configuration Register	DEVADD5	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 00F0h	USB0	PHY Cross Point Adjustment Register	PHYSLEW	32	32	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0200h	USB1	System Configuration Control Register	SYSCFG	16	16	3, 4 PCLKB	2 ICLK	section 34
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)*1	section 34
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)*1	section 34
000A 0214h	USB1	CFIFO Port Register	CFIFO	16	8, 16	3, 4 PCLKB	2 ICLK	section 34
000A 0218h	USB1	D0FIFO Port Register	D0FIFO	16	8, 16	3, 4 PCLKB	2 ICLK	section 34
000A 021Ch	USB1	D1FIFO Port Register	D1FIFO	16	8, 16	3, 4 PCLKB	2 ICLK	section 34
000A 0220h	USB1	CFIFO Port Select Register	CFIFOSEL	16	16	3, 4 PCLKB	2 ICLK	section 34
000A 0222h	USB1	CFIFO Port Control Register	CFIFOCTR	16	16	3, 4 PCLKB	2 ICLK	section 34
000A 0228h	USB1	D0FIFO Port Select Register	D0FIFOSEL	16	16	3, 4 PCLKB	2 ICLK	section 34
000A 022Ah	USB1	D0FIFO Port Control Register	D0FIFOCTR	16	16	3, 4 PCLKB	2 ICLK	section 34
000A 022Ch	USB1	D1FIFO Port Select Register	D1FIFOSEL	16	16	3, 4 PCLKB	2 ICLK	section 34
000A 022Eh	USB1	D1FIFO Port Control Register	D1FIFOCTR	16	16	3, 4 PCLKB	2 ICLK	section 34

Table 5.1 List of I/O Registers (Address Order) (41 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000A 0230h	USB1	Interrupt Enable Register 0	INTENB0	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0232h	USB1	Interrupt Enable Register 1	INTENB1	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0236h	USB1	BRDY Interrupt Enable Register	BRDYENB	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0238h	USB1	NRDY Interrupt Enable Register	NRDYENB	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 023Ah	USB1	BEMP Interrupt Enable Register	BEMPENB	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 023Ch	USB1	SOF Output Configuration Register	SOFCFG	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0240h	USB1	Interrupt Status Register 0	INTSTS0	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0242h	USB1	Interrupt Status Register 1	INTSTS1	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0246h	USB1	BRDY Interrupt Status Register	BRDYSTS	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0248h	USB1	NRDY Interrupt Status Register	NRDYSTS	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 024Ah	USB1	BEMP Interrupt Status Register	BEMPSTS	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 024Ch	USB1	Frame Number Register	FRMNUM	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0254h	USB1	USB Request Type Register	USBREQ	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0256h	USB1	USB Request Value Register	USBVAL	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0258h	USB1	USB Request Index Register	USBINDX	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 025Ah	USB1	USB Request Length Register	USBLENG	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34

Table 5.1 List of I/O Registers (Address Order) (42 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000A 025Ch	USB1	DCP Configuration Register	DCPCFG	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 025Eh	USB1	DCP Maximum Packet Size Register	DCPMAXP	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0260h	USB1	DCP Control Register	DCPCTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0264h	USB1	Pipe Window Select Register	PIPESEL	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0268h	USB1	Pipe Configuration Register	PIPECFG	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 026Ch	USB1	Pipe Maximum Packet Size Register	PIPEMAXP	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 026Eh	USB1	Pipe Cycle Control Register	PIPEPERI	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0270h	USB1	Pipe 1 Control Register	PIPE1CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0272h	USB1	Pipe 2 Control Register	PIPE2CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0274h	USB1	Pipe 3 Control Register	PIPE3CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0276h	USB1	Pipe 4 Control Register	PIPE4CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0278h	USB1	Pipe 5 Control Register	PIPE5CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 027Ah	USB1	Pipe 6 Control Register	PIPE6CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 027Ch	USB1	Pipe 7 Control Register	PIPE7CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 027Eh	USB1	Pipe 8 Control Register	PIPE8CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0280h	USB1	Pipe 9 Control Register	PIPE9CTR	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34

Table 5.1 List of I/O Registers (Address Order) (43 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000A 0290h	USB1	Pipe 1 Transaction Counter Enable Register	PIPE1TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0292h	USB1	Pipe 1 Transaction Counter Register	PIPE1TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0294h	USB1	Pipe 2 Transaction Counter Enable Register	PIPE2TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0296h	USB1	Pipe 2 Transaction Counter Register	PIPE2TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0298h	USB1	Pipe 3 Transaction Counter Enable Register	PIPE3TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 029Ah	USB1	Pipe 3 Transaction Counter Register	PIPE3TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 029Ch	USB1	Pipe 4 Transaction Counter Enable Register	PIPE4TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 029Eh	USB1	Pipe 4 Transaction Counter Register	PIPE4TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 02A0h	USB1	Pipe 5 Transaction Counter Enable Register	PIPE5TRE	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 02A2h	USB1	Pipe 5 Transaction Counter Register	PIPE5TRN	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 02D0h	USB1	Device Address 0 Configuration Register	DEVADD0	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 02D2h	USB1	Device Address 1 Configuration Register	DEVADD1	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 02D4h	USB1	Device Address 2 Configuration Register	DEVADD2	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 02D6h	USB1	Device Address 3 Configuration Register	DEVADD3	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 02D8h	USB1	Device Address 4 Configuration Register	DEVADD4	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 02DAh	USB1	Device Address 5 Configuration Register	DEVADD5	16	16	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34

Table 5.1 List of I/O Registers (Address Order) (44 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000A 02F0h	USB1	PHY Cross Point Adjustment Register	PHYSLEW	32	32	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0400h	USB	Deep Standby USB Transceiver Control/Pin Monitoring Register	DPUSR0R	32	32	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0404h	USB	Deep Standby USB Suspend/Resume Interrupt Register	DPUSR1R	32	32	9 PCLKB or more	Frequency with 1 + 9 × (frequency ratio of ICLK/ PCLKB)*1	section 34
000A 0580h	DOC	DOC Control Register	DOCR	8	8	2, 3 PCLKB	2 ICLK	section 52
000A 0584h	DOC	DOC Status Register	DOSR	8	8	2, 3 PCLKB	2 ICLK	section 52
000A 0588h	DOC	DOC Status Clear Register	DOSCR	8	8	2, 3 PCLKB	2 ICLK	section 52
000A 058Ch	DOC	DOC Data Input Register	DODIR	32	16, 32	2, 3 PCLKB	2 ICLK	section 52
000A 0590h	DOC	DOC Data Setting Register 0	DODSR0	32	16, 32	2, 3 PCLKB	2 ICLK	section 52
000A 0594h	DOC	DOC Data Setting Register 1	DODSR1	32	16, 32	2, 3 PCLKB	2 ICLK	section 52
000A 0900h	CTSU	CTSU Control Register 0	CTSUCR0	8	8	2, 3 PCLKB	2 ICLK	section 47
000A 0901h	CTSU	CTSU Control Register 1	CTSUCR1	8	8	2, 3 PCLKB	2 ICLK	section 47
000A 0902h	CTSU	CTSU Synchronous Noise Reduction Setting Register	CTSUSDPRS	8	8	2, 3 PCLKB	2 ICLK	section 47
000A 0903h	CTSU	CTSU Sensor Stabilization Wait Control Register	CTSUSST	8	8	2, 3 PCLKB	2 ICLK	section 47
000A 0904h	CTSU	CTSU Measurement Channel Register 0	CTSUMCH0	8	8	2, 3 PCLKB	2 ICLK	section 47
000A 0905h	CTSU	CTSU Measurement Channel Register 1	CTSUMCH1	8	8	2, 3 PCLKB	2 ICLK	section 47
000A 0906h	CTSU	CTSU Channel Enable Control Register 0	CTSUCHAC0	8	8	2, 3 PCLKB	2 ICLK	section 47
000A 0907h	CTSU	CTSU Channel Enable Control Register 1	CTSUCHAC1	8	8	2, 3 PCLKB	2 ICLK	section 47
000A 0908h	CTSU	CTSU Channel Enable Control Register 2	CTSUCHAC2	8	8	2, 3 PCLKB	2 ICLK	section 47
000A 090Bh	CTSU	CTSU Channel Transmit/Receive Control Register 0	CTSUCHTRC 0	8	8	2, 3 PCLKB	2 ICLK	section 47
000A 090Ch	CTSU	CTSU Channel Transmit/Receive Control Register 1	CTSUCHTRC 1	8	8	2, 3 PCLKB	2 ICLK	section 47
000A 090Dh	CTSU	CTSU Channel Transmit/Receive Control Register 2	CTSUCHTRC 2	8	8	2, 3 PCLKB	2 ICLK	section 47
000A 0910h	CTSU	CTSU High-Pass Noise Reduction Control Register	CTSUDCLKC	8	8	2, 3 PCLKB	2 ICLK	section 47
000A 0911h	CTSU	CTSU Status Register	CTSUST	8	8	2, 3 PCLKB	2 ICLK	section 47
000A 0912h	CTSU	CTSU High-Pass Noise Reduction Spectrum Diffusion Control Register	CTSUSSC	16	16	2, 3 PCLKB	2 ICLK	section 47
000A 0914h	CTSU	CTSU Sensor Offset Register 0	CTSUSO0	16	16	2, 3 PCLKB	2 ICLK	section 47
000A 0916h	CTSU	CTSU Sensor Offset Register 1	CTSUSO1	16	16	2, 3 PCLKB	2 ICLK	section 47
000A 0918h	CTSU	CTSU Sensor Counter	CTSUSC	16	16	2, 3 PCLKB	2 ICLK	section 47
000A 091Ah	CTSU	CTSU Reference Counter	CTSURC	16	16	2, 3 PCLKB	2 ICLK	section 47
000A 091Ch	CTSU	CTSU Error Status Register	CTSUERRS	16	16	2, 3 PCLKB	2 ICLK	section 47
000A 0B00h	REMC0	Function Select Register 0	REMCON0	8	8	2, 3 PCLKB	2 ICLK	section 46
000A 0B01h	REMC0	Function Select Register 1	REMCON1	8	8	2, 3 PCLKB	2 ICLK	section 46
000A 0B02h	REMC0	Status Register	REMSTS	8	8	2, 3 PCLKB	2 ICLK	section 46
000A 0B03h	REMC0	Interrupt Control Register	REMINT	8	8	2, 3 PCLKB	2 ICLK	section 46
000A 0B05h	REMC0	Compare Control Register	REMCPC	8	8	2, 3 PCLKB	2 ICLK	section 46
000A 0B06h	REMC0	Compare Value Setting Register	REMCPD	16	16	2, 3 PCLKB	2 ICLK	section 46
000A 0B08h	REMC0	Header Pattern Minimum Width Setting Register	HDPMIN	16	16	2, 3 PCLKB	2 ICLK	section 46
000A 0B0Ah	REMC0	Header Pattern Maximum Width Setting Register	HDPMAX	16	16	2, 3 PCLKB	2 ICLK	section 46
000A 0B0Ch	REMC0	Data ‘0’ Pattern Minimum Width Setting Register	D0PMIN	8	8	2, 3 PCLKB	2 ICLK	section 46
000A 0B0Dh	REMC0	Data ‘0’ Pattern Maximum Width Setting Register	D0PMAX	8	8	2, 3 PCLKB	2 ICLK	section 46
000A 0B0Eh	REMC0	Data ‘1’ Pattern Minimum Width Setting Register	D1PMIN	8	8	2, 3 PCLKB	2 ICLK	section 46
000A 0B0Fh	REMC0	Data ‘1’ Pattern Maximum Width Setting Register	D1PMAX	8	8	2, 3 PCLKB	2 ICLK	section 46

Table 5.1 List of I/O Registers (Address Order) (45 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000A 0B10h	REMC0	Special Data Pattern Minimum Width Setting Register	SDPMIN	16	16	2, 3 PCLKB	2 ICLK	section 46
000A 0B12h	REMC0	Special Data Pattern Maximum Width Setting Register	SDPMAX	16	16	2, 3 PCLKB	2 ICLK	section 46
000A 0B14h	REMC0	Pattern End Setting Register	REMPE	16	16	2, 3 PCLKB	2 ICLK	section 46
000A 0B17h	REMC0	Receive Bit Count Register	REMRBIT	8	8	2, 3 PCLKB	2 ICLK	section 46
000A 0B18h	REMC0	Receive Data 0 Register	REMDAT0	8	8	2, 3 PCLKB	2 ICLK	section 46
000A 0B19h	REMC0	Receive Data 1 Register	REMDAT1	8	8	2, 3 PCLKB	2 ICLK	section 46
000A 0B1Ah	REMC0	Receive Data 2 Register	REMDAT2	8	8	2, 3 PCLKB	2 ICLK	section 46
000A 0B1Bh	REMC0	Receive Data 3 Register	REMDAT3	8	8	2, 3 PCLKB	2 ICLK	section 46
000A 0B1Ch	REMC0	Receive Data 4 Register	REMDAT4	8	8	2, 3 PCLKB	2 ICLK	section 46
000A 0B1Dh	REMC0	Receive Data 5 Register	REMDAT5	8	8	2, 3 PCLKB	2 ICLK	section 46
000A 0B1Eh	REMC0	Receive Data 6 Register	REMDAT6	8	8	2, 3 PCLKB	2 ICLK	section 46
000A 0B1Fh	REMC0	Receive Data 7 Register	REMDAT7	8	8	2, 3 PCLKB	2 ICLK	section 46
000A 0B20h	REMC0	Measurement Result Register	REMTIM	16	16	2, 3 PCLKB	2 ICLK	section 46
000C 1200h	MTU3	Timer Control Register	TCR	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1201h	MTU4	Timer Control Register	TCR	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1202h	MTU3	Timer Mode Register 1	TMDR1	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1203h	MTU4	Timer Mode Register 1	TMDR1	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1204h	MTU3	Timer I/O Control Register H	TIORH	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1205h	MTU3	Timer I/O Control Register L	TIORL	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1206h	MTU4	Timer I/O Control Register H	TIORH	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1207h	MTU4	Timer I/O Control Register L	TIORL	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1208h	MTU3	Timer Interrupt Enable Register	TIER	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1209h	MTU4	Timer Interrupt Enable Register	TIER	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 120Ah	MTU	Timer Output Master Enable Register A	TOERA	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 120Dh	MTU	Timer Gate Control Register A	TGCRA	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 120Eh	MTU	Timer Output Control Register 1A	TOCR1A	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 120Fh	MTU	Timer Output Control Register 2A	TOCR2A	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1210h	MTU3	Timer Counter	TCNT	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1212h	MTU4	Timer Counter	TCNT	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1214h	MTU	Timer Cycle Data Register A	TCDRA	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1216h	MTU	Timer Dead Time Data Register A	TDDRA	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1218h	MTU3	Timer General Register A	TGRA	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 121Ah	MTU3	Timer General Register B	TGRB	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 121Ch	MTU4	Timer General Register A	TGRA	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 121Eh	MTU4	Timer General Register B	TGRB	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1220h	MTU	Timer Subcounter A	TCNTSA	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1222h	MTU	Timer Cycle Buffer Register A	TCBRA	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1224h	MTU3	Timer General Register C	TGRC	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1226h	MTU3	Timer General Register D	TGRD	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1228h	MTU4	Timer General Register C	TGRC	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 122Ah	MTU4	Timer General Register D	TGRD	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 122Ch	MTU3	Timer Status Register	TSR	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 122Dh	MTU4	Timer Status Register	TSR	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1230h	MTU	Timer Interrupt Skipping Set Register 1A	TITCR1A	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1231h	MTU	Timer Interrupt Skipping Counter 1A	TITCNT1A	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1232h	MTU	Timer Buffer Transfer Set Register A	TBTERA	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1234h	MTU	Timer Dead Time Enable Register A	TDERA	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1236h	MTU	Timer Output Level Buffer Register A	TOLBRA	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1238h	MTU3	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1239h	MTU4	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	4, 5 PCLKA	1, 2 ICLK	section 24

Table 5.1 List of I/O Registers (Address Order) (46 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 123Ah	MTU	Timer Interrupt Skipping Mode Register A	TITMRA	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 123Bh	MTU	Timer Interrupt Skipping Set Register 2A	TITCR2A	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 123Ch	MTU	Timer Interrupt Skipping Counter 2A	TITCNT2A	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1240h	MTU4	Timer A/D Conversion Start Request Control Register	TADCR	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1244h	MTU4	Timer A/D Conversion Start Request Cycle Set Register A	TADCORA	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1246h	MTU4	Timer A/D Conversion Start Request Cycle Set Register B	TADCORB	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1248h	MTU4	Timer A/D Conversion Start Request Cycle Set Buffer Register A	TADCOBRA	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 124Ah	MTU4	Timer A/D Conversion Start Request Cycle Set Buffer Register B	TADCOBRB	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 124Ch	MTU3	Timer Control Register 2	TCR2	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 124Dh	MTU4	Timer Control Register 2	TCR2	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1260h	MTU	Timer Waveform Control Register A	TWCRA	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1270h	MTU	Timer Mode Register 2A	TMDR2A	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1272h	MTU3	Timer General Register E	TGRE	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1274h	MTU4	Timer General Register E	TGRE	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1276h	MTU4	Timer General Register F	TGRF	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1280h	MTU	Timer Start Register A	TSTRA	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1281h	MTU	Timer Synchronous Register A	TSYRA	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1282h	MTU	Timer Counter Synchronous Start Register	TCSYSTR	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1284h	MTU	Timer Read/Write Enable Register A	TRWERA	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1290h	MTU0	Noise Filter Control Register 0	NFCR0	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1291h	MTU1	Noise Filter Control Register 1	NFCR1	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1292h	MTU2	Noise Filter Control Register 2	NFCR2	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1293h	MTU3	Noise Filter Control Register 3	NFCR3	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1294h	MTU4	Noise Filter Control Register 4	NFCR4	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1298h	MTU8	Noise Filter Control Register 8	NFCR8	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1299h	MTU0	Noise Filter Control Register C	NFCRC	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1300h	MTU0	Timer Control Register	TCR	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1301h	MTU0	Timer Mode Register 1	TMDR1	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1302h	MTU0	Timer I/O Control Register H	TIORH	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1303h	MTU0	Timer I/O Control Register L	TIORL	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1304h	MTU0	Timer Interrupt Enable Register	TIER	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1306h	MTU0	Timer Counter	TCNT	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1308h	MTU0	Timer General Register A	TGRA	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 130Ah	MTU0	Timer General Register B	TGRB	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 130Ch	MTU0	Timer General Register C	TGRC	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 130Eh	MTU0	Timer General Register D	TGRD	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1320h	MTU0	Timer General Register E	TGRE	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1322h	MTU0	Timer General Register F	TGRF	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1324h	MTU0	Timer Interrupt Enable Register 2	TIER2	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1326h	MTU0	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1328h	MTU0	Timer Control Register 2	TCR2	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1380h	MTU1	Timer Control Register	TCR	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1381h	MTU1	Timer Mode Register 1	TMDR1	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1382h	MTU1	Timer I/O Control Register	TIOR	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1384h	MTU1	Timer Interrupt Enable Register	TIER	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1385h	MTU1	Timer Status Register	TSR	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1386h	MTU1	Timer Counter	TCNT	16	16	4, 5 PCLKA	1, 2 ICLK	section 24

Table 5.1 List of I/O Registers (Address Order) (47 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 1388h	MTU1	Timer General Register A	TGRA	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 138Ah	MTU1	Timer General Register B	TGRB	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1390h	MTU1	Timer Input Capture Control Register	TICCR	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1391h	MTU1	Timer Mode Register 3	TMDR3	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1394h	MTU1	Timer Control Register 2	TCR2	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 13A0h	MTU1	Timer Longword Counter	TCNTLW	32	32	4, 5 PCLKA	1, 2 ICLK	section 24
000C 13A4h	MTU1	Timer Longword General Register	TGRALW	32	32	4, 5 PCLKA	1, 2 ICLK	section 24
000C 13A8h	MTU1	Timer Longword General Register	TGRBLW	32	32	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1400h	MTU2	Timer Control Register	TCR	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1401h	MTU2	Timer Mode Register 1	TMDR1	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1402h	MTU2	Timer I/O Control Register	TIOR	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1404h	MTU2	Timer Interrupt Enable Register	TIER	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1405h	MTU2	Timer Status Register	TSR	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1406h	MTU2	Timer Counter	TCNT	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1408h	MTU2	Timer General Register A	TGRA	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 140Ah	MTU2	Timer General Register B	TGRB	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 140Ch	MTU2	Timer Control Register 2	TCR2	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1600h	MTU8	Timer Control Register	TCR	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1601h	MTU8	Timer Mode Register 1	TMDR1	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1602h	MTU8	Timer I/O Control Register H	TIORH	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1603h	MTU8	Timer I/O Control Register L	TIORL	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1604h	MTU8	Timer Interrupt Enable Register	TIER	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1606h	MTU8	Timer Control Register 2	TCR2	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1608h	MTU8	Timer Counter	TCNT	32	32	4, 5 PCLKA	1, 2 ICLK	section 24
000C 160Ch	MTU8	Timer General Register A	TGRA	32	32	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1610h	MTU8	Timer General Register B	TGRB	32	32	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1614h	MTU8	Timer General Register C	TGRC	32	32	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1618h	MTU8	Timer General Register D	TGRD	32	32	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A00h	MTU6	Timer Control Register	TCR	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A01h	MTU7	Timer Control Register	TCR	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A02h	MTU6	Timer Mode Register 1	TMDR1	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A03h	MTU7	Timer Mode Register 1	TMDR1	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A04h	MTU6	Timer I/O Control Register H	TIORH	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A05h	MTU6	Timer I/O Control Register L	TIORL	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A06h	MTU7	Timer I/O Control Register H	TIORH	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A07h	MTU7	Timer I/O Control Register L	TIORL	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A08h	MTU6	Timer Interrupt Enable Register	TIER	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A09h	MTU7	Timer Interrupt Enable Register	TIER	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A0Ah	MTU	Timer Output Master Enable Register B	TOERB	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A0Eh	MTU	Timer Output Control Register 1B	TOCR1B	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A0Fh	MTU	Timer Output Control Register 2B	TOCR2B	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A10h	MTU6	Timer Counter	TCNT	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A12h	MTU7	Timer Counter	TCNT	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A14h	MTU	Timer Cycle Data Register B	TCDRB	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A16h	MTU	Timer Dead Time Data Register B	TDDRB	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A18h	MTU6	Timer General Register A	TGRA	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A1Ah	MTU6	Timer General Register B	TGRB	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A1Ch	MTU7	Timer General Register A	TGRA	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A1Eh	MTU7	Timer General Register B	TGRB	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A20h	MTU	Timer Subcounter B	TCNTSB	16	16	4, 5 PCLKA	1, 2 ICLK	section 24

Table 5.1 List of I/O Registers (Address Order) (48 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 1A22h	MTU	Timer Cycle Buffer Register B	TCBRB	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A24h	MTU6	Timer General Register C	TGRC	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A26h	MTU6	Timer General Register D	TGRD	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A28h	MTU7	Timer General Register C	TGRC	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A2Ah	MTU7	Timer General Register D	TGRD	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A2Ch	MTU6	Timer Status Register	TSR	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A2Dh	MTU7	Timer Status Register	TSR	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A30h	MTU	Timer Interrupt Skipping Set Register 1B	TITCR1B	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A31h	MTU	Timer Interrupt Skipping Counter 1B	TITCNT1B	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A32h	MTU	Timer Buffer Transfer Set Register B	TBTERB	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A34h	MTU	Timer Dead Time Enable Register B	TDERB	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A36h	MTU	Timer Output Level Buffer Register B	TOLBRB	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A38h	MTU6	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A39h	MTU7	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A3Ah	MTU	Timer Interrupt Skipping Mode Register B	TITMRB	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A3Bh	MTU	Timer Interrupt Skipping Set Register 2B	TITCR2B	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A3Ch	MTU	Timer Interrupt Skipping Counter 2B	TITCNT2B	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A40h	MTU7	Timer A/D Conversion Start Request Control Register	TADCR	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A44h	MTU7	Timer A/D Conversion Start Request Cycle Set Register A	TADCORA	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A46h	MTU7	Timer A/D Conversion Start Request Cycle Set Register B	TADCORB	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A48h	MTU7	Timer A/D Conversion Start Request Cycle Set Buffer Register A	TADCOBRA	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A4Ah	MTU7	Timer A/D Conversion Start Request Cycle Set Buffer Register B	TADCOBRB	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A4Ch	MTU6	Timer Control Register 2	TCR2	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A4Dh	MTU7	Timer Control Register 2	TCR2	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A50h	MTU6	Timer Synchronous Clear Register	TSYCR	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A60h	MTU	Timer Waveform Control Register B	TWCRB	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A70h	MTU	Timer Mode Register 2B	TMDR2B	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A72h	MTU6	Timer General Register E	TGRE	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A74h	MTU7	Timer General Register E	TGRE	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A76h	MTU7	Timer General Register F	TGRF	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A80h	MTU	Timer Start Register B	TSTRB	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A81h	MTU	Timer Synchronous Register B	TSYRB	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A84h	MTU	Timer Read/Write Enable Register B	TRWERB	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A93h	MTU6	Noise Filter Control Register 6	NFCR6	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A94h	MTU7	Noise Filter Control Register 7	NFCR7	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1A95h	MTU5	Noise Filter Control Register 5	NFCR5	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1C80h	MTU5	Timer Counter U	TCNTU	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1C82h	MTU5	Timer General Register U	TGRU	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1C84h	MTU5	Timer Control Register U	TCRU	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1C85h	MTU5	Timer Control Register 2	TCR2U	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1C86h	MTU5	Timer I/O Control Register U	TIORU	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1C90h	MTU5	Timer Counter V	TCNTV	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1C92h	MTU5	Timer General Register V	TGRV	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1C94h	MTU5	Timer Control Register V	TCRV	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1C95h	MTU5	Timer Control Register 2	TCR2V	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1C96h	MTU5	Timer I/O Control Register V	TIORV	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1CA0h	MTU5	Timer Counter W	TCNTW	16	16	4, 5 PCLKA	1, 2 ICLK	section 24

Table 5.1 List of I/O Registers (Address Order) (49 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000C 1CA2h	MTU5	Timer General Register W	TGRW	16	16	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1CA4h	MTU5	Timer Control Register W	TCRW	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1CA5h	MTU5	Timer Control Register 2	TCR2W	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1CA6h	MTU5	Timer I/O Control Register W	TIORW	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1CB2h	MTU5	Timer Interrupt Enable Register	TIER	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1CB4h	MTU5	Timer Start Register	TSTR	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000C 1CB6h	MTU5	Timer Compare Match Clear Register	TCNTCMPCLR	8	8	4, 5 PCLKA	1, 2 ICLK	section 24
000D 0040h	SCI10	Serial Mode Register	SMR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0040h	SMCI10	Serial Mode Register	SMR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0041h	SCI10	Bit Rate Register	BRR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0042h	SCI10	Serial Control Register	SCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0042h	SMCI10	Serial Control Register	SCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0043h	SCI10	Transmit Data Register	TDR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0044h	SCI10	Serial Status Register	SSR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0044h	SMCI10	Serial Status Register	SSR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0044h	SCI10	Serial Status Register	SSRFIFO	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0045h	SCI10	Receive Data Register	RDR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0046h	SCI10	Smart Card Mode Register	SCMR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0046h	SMCI10	Smart Card Mode Register	SCMR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0047h	SCI10	Serial Extended Mode Register	SEMR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0048h	SCI10	Noise Filter Setting Register	SNFR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0049h	SCI10	I^2C Mode Register 1	SIMR1	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 004Ah	SCI10	I^2C Mode Register 2	SIMR2	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 004Bh	SCI10	I^2C Mode Register 3	SIMR3	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 004Ch	SCI10	I^2C Status Register	SISR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 004Dh	SCI10	SPI Mode Register	SPMR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 004Eh	SCI10	Transmit Data Register H	TDRH	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 004Fh	SCI10	Transmit Data Register L	TDRL	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 004Eh	SCI10	Transmit Data Register HL	TDRHL	16	16	5, 6 PCLKA	1 to 3 ICLK	section 35
000D 004Eh	SCI10	Transmit FIFO Data Register	FTDR.H	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 004Fh	SCI10	Transmit FIFO Data Register	FTDR.L	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 004Eh	SCI10	Transmit FIFO Data Register	FTDR	16	16	5, 6 PCLKA	1 to 3 ICLK	section 35
000D 0050h	SCI10	Receive Data Register H	RDRH	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0051h	SCI10	Receive Data Register L	RDRL	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0050h	SCI10	Receive Data Register HL	RDRHL	16	16	5, 6 PCLKA	1 to 3 ICLK	section 35
000D 0050h	SCI10	Receive FIFO Data Register	FRDR.H	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0051h	SCI10	Receive FIFO Data Register	FRDR.L	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0050h	SCI10	Receive FIFO Data Register	FRDR	16	16	5, 6 PCLKA	1 to 3 ICLK	section 35
000D 0052h	SCI10	Modulation Duty Register	MDDR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0053h	SCI10	Data Comparison Control Register	DCCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0054h	SCI10	FIFO Control Register	FCR.H	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0055h	SCI10	FIFO Control Register	FCR.L	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0054h	SCI10	FIFO Control Register	FCR	16	16	5, 6 PCLKA	1 to 3 ICLK	section 35
000D 0056h	SCI10	FIFO Data Count Register	FDR.H	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0057h	SCI10	FIFO Data Count Register	FDR.L	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0056h	SCI10	FIFO Data Count Register	FDR	16	16	5, 6 PCLKA	1 to 3 ICLK	section 35
000D 0058h	SCI10	Line Status Register	LSR.H	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0059h	SCI10	Line Status Register	LSR.L	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0058h	SCI10	Line Status Register	LSR	16	16	5, 6 PCLKA	1 to 3 ICLK	section 35

Table 5.1 List of I/O Registers (Address Order) (50 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000D 005Ah	SCI10	Comparison Data Register	CDR.H	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 005Bh	SCI10	Comparison Data Register	CDR.L	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 005Ah	SCI10	Comparison Data Register	CDR	16	16	5, 6 PCLKA	1 to 3 ICLK	section 35
000D 005Ch	SCI10	Serial Port Register	SPTR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 005Dh	SCI10	Transmit/Receive Timing Select Register	TMGR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0060h	SCI11	Serial Mode Register	SMR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0060h	SMCI11	Serial Mode Register	SMR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0061h	SCI11	Bit Rate Register	BRR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0062h	SCI11	Serial Control Register	SCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0062h	SMCI11	Serial Control Register	SCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0063h	SCI11	Transmit Data Register	TDR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0064h	SCI11	Serial Status Register	SSR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0064h	SMCI11	Serial Status Register	SSR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0064h	SCI11	Serial Status Register	SSRFIFO	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0065h	SCI11	Receive Data Register	RDR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0066h	SCI11	Smart Card Mode Register	SCMR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0066h	SMCI11	Smart Card Mode Register	SCMR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0067h	SCI11	Serial Extended Mode Register	SEMR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0068h	SCI11	Noise Filter Setting Register	SNFR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0069h	SCI11	I^2C Mode Register 1	SIMR1	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 006Ah	SCI11	I^2C Mode Register 2	SIMR2	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 006Bh	SCI11	I^2C Mode Register 3	SIMR3	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 006Ch	SCI11	I^2C Status Register	SISR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 006Dh	SCI11	SPI Mode Register	SPMR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 006Eh	SCI11	Transmit Data Register H	TDRH	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 006Fh	SCI11	Transmit Data Register L	TDRL	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 006Eh	SCI11	Transmit Data Register HL	TDRHL	16	16	5, 6 PCLKA	1 to 3 ICLK	section 35
000D 006Eh	SCI11	Transmit FIFO Data Register	FTDR.H	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 006Fh	SCI11	Transmit FIFO Data Register	FTDR.L	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 006Eh	SCI11	Transmit FIFO Data Register	FTDR	16	16	5, 6 PCLKA	1 to 3 ICLK	section 35
000D 0070h	SCI11	Receive Data Register H	RDRH	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0071h	SCI11	Receive Data Register L	RDRL	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0070h	SCI11	Receive Data Register HL	RDRHL	16	16	5, 6 PCLKA	1 to 3 ICLK	section 35
000D 0070h	SCI11	Receive FIFO Data Register	FRDR.H	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0071h	SCI11	Receive FIFO Data Register	FRDR.L	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0070h	SCI11	Receive FIFO Data Register	FRDR	16	16	5, 6 PCLKA	1 to 3 ICLK	section 35
000D 0072h	SCI11	Modulation Duty Register	MDDR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0073h	SCI11	Data Comparison Control Register	DCCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0074h	SCI11	FIFO Control Register	FCR.H	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0075h	SCI11	FIFO Control Register	FCR.L	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0074h	SCI11	FIFO Control Register	FCR	16	16	5, 6 PCLKA	1 to 3 ICLK	section 35
000D 0076h	SCI11	FIFO Data Count Register	FDR.H	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0077h	SCI11	FIFO Data Count Register	FDR.L	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0076h	SCI11	FIFO Data Count Register	FDR	16	16	5, 6 PCLKA	1 to 3 ICLK	section 35
000D 0078h	SCI11	Line Status Register	LSR.H	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0079h	SCI11	Line Status Register	LSR.L	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0078h	SCI11	Line Status Register	LSR	16	16	5, 6 PCLKA	1 to 3 ICLK	section 35
000D 007Ah	SCI11	Comparison Data Register	CDR.H	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 007Bh	SCI11	Comparison Data Register	CDR.L	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 007Ah	SCI11	Comparison Data Register	CDR	16	16	5, 6 PCLKA	1 to 3 ICLK	section 35

Table 5.1 List of I/O Registers (Address Order) (51 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000D 007Ch	SCI11	Serial Port Register	SPTR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 007Dh	SCI11	Transmit/Receive Timing Select Register	TMGR	8	8	3, 4 PCLKA	1, 2 ICLK	section 35
000D 0100h	RSPI0	RSPI Control Register	SPCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0101h	RSPI0	RSPI Slave Select Polarity Register	SSLP	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0102h	RSPI0	RSPI Pin Control Register	SPPCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0103h	RSPI0	RSPI Status Register	SPSR	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0104h	RSPI0	RSPI Data Register	SPDR	32	8, 16, 32	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0108h	RSPI0	RSPI Sequence Control Register	SPSCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0109h	RSPI0	RSPI Sequence Status Register	SPSSR	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 010Ah	RSPI0	RSPI Bit Rate Register	SPBR	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 010Bh	RSPI0	RSPI Data Control Register	SPDCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 010Ch	RSPI0	RSPI Clock Delay Register	SPCKD	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 010Dh	RSPI0	RSPI Slave Select Negation Delay Register	SSLND	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 010Eh	RSPI0	RSPI Next-Access Delay Register	SPND	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 010Fh	RSPI0	RSPI Control Register 2	SPCR2	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0110h	RSPI0	RSPI Command Register 0	SPCMD0	16	16	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0112h	RSPI0	RSPI Command Register 1	SPCMD1	16	16	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0114h	RSPI0	RSPI Command Register 2	SPCMD2	16	16	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0116h	RSPI0	RSPI Command Register 3	SPCMD3	16	16	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0118h	RSPI0	RSPI Command Register 4	SPCMD4	16	16	3, 4 PCLKA	1, 2 ICLK	section 40
000D 011Ah	RSPI0	RSPI Command Register 5	SPCMD5	16	16	3, 4 PCLKA	1, 2 ICLK	section 40
000D 011Ch	RSPI0	RSPI Command Register 6	SPCMD6	16	16	3, 4 PCLKA	1, 2 ICLK	section 40
000D 011Eh	RSPI0	RSPI Command Register 7	SPCMD7	16	16	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0120h	RSPI0	RSPI Data Control Register 2	SPDCR2	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0121h	RSPI0	RSPI Control Register 3	SPCR3	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0140h	RSPI1	RSPI Control Register	SPCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0141h	RSPI1	RSPI Slave Select Polarity Register	SSLP	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0142h	RSPI1	RSPI Pin Control Register	SPPCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0143h	RSPI1	RSPI Status Register	SPSR	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0144h	RSPI1	RSPI Data Register	SPDR	32	8, 16, 32	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0148h	RSPI1	RSPI Sequence Control Register	SPSCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0149h	RSPI1	RSPI Sequence Status Register	SPSSR	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 014Ah	RSPI1	RSPI Bit Rate Register	SPBR	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 014Bh	RSPI1	RSPI Data Control Register	SPDCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 014Ch	RSPI1	RSPI Clock Delay Register	SPCKD	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 014Dh	RSPI1	RSPI Slave Select Negation Delay Register	SSLND	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 014Eh	RSPI1	RSPI Next-Access Delay Register	SPND	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 014Fh	RSPI1	RSPI Control Register 2	SPCR2	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0150h	RSPI1	RSPI Command Register 0	SPCMD0	16	16	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0152h	RSPI1	RSPI Command Register 1	SPCMD1	16	16	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0154h	RSPI1	RSPI Command Register 2	SPCMD2	16	16	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0156h	RSPI1	RSPI Command Register 3	SPCMD3	16	16	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0158h	RSPI1	RSPI Command Register 4	SPCMD4	16	16	3, 4 PCLKA	1, 2 ICLK	section 40
000D 015Ah	RSPI1	RSPI Command Register 5	SPCMD5	16	16	3, 4 PCLKA	1, 2 ICLK	section 40
000D 015Ch	RSPI1	RSPI Command Register 6	SPCMD6	16	16	3, 4 PCLKA	1, 2 ICLK	section 40
000D 015Eh	RSPI1	RSPI Command Register 7	SPCMD7	16	16	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0160h	RSPI1	RSPI Data Control Register 2	SPDCR2	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0161h	RSPI1	RSPI Control Register 3	SPCR3	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0300h	RSPI2	RSPI Control Register	SPCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 40

Table 5.1 List of I/O Registers (Address Order) (52 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000D 0301h	RSPI2	RSPI Slave Select Polarity Register	SSLP	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0302h	RSPI2	RSPI Pin Control Register	SPPCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0303h	RSPI2	RSPI Status Register	SPSR	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0304h	RSPI2	RSPI Data Register	SPDR	32	8, 16, 32	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0308h	RSPI2	RSPI Sequence Control Register	SPSCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0309h	RSPI2	RSPI Sequence Status Register	SPSSR	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 030Ah	RSPI2	RSPI Bit Rate Register	SPBR	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 030Bh	RSPI2	RSPI Data Control Register	SPDCR	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 030Ch	RSPI2	RSPI Clock Delay Register	SPCKD	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 030Dh	RSPI2	RSPI Slave Select Negation Delay Register	SSLND	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 030Eh	RSPI2	RSPI Next-Access Delay Register	SPND	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 030Fh	RSPI2	RSPI Control Register 2	SPCR2	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0310h	RSPI2	RSPI Command Register 0	SPCMD0	16	16	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0312h	RSPI2	RSPI Command Register 1	SPCMD1	16	16	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0314h	RSPI2	RSPI Command Register 2	SPCMD2	16	16	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0316h	RSPI2	RSPI Command Register 3	SPCMD3	16	16	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0318h	RSPI2	RSPI Command Register 4	SPCMD4	16	16	3, 4 PCLKA	1, 2 ICLK	section 40
000D 031Ah	RSPI2	RSPI Command Register 5	SPCMD5	16	16	3, 4 PCLKA	1, 2 ICLK	section 40
000D 031Ch	RSPI2	RSPI Command Register 6	SPCMD6	16	16	3, 4 PCLKA	1, 2 ICLK	section 40
000D 031Eh	RSPI2	RSPI Command Register 7	SPCMD7	16	16	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0320h	RSPI2	RSPI Data Control Register 2	SPDCR2	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000D 0321h	RSPI2	RSPI Control Register 3	SPCR3	8	8	3, 4 PCLKA	1, 2 ICLK	section 40
000E 2000h	RSCI10	Receive Data Register	RDR	32	8, 16, 32	2, 3 PCLKA	2ICLK	section 36
000E 2004h	RSCI10	Transmit Data Register	TDR	32	8, 16, 32	2, 3 PCLKA	2ICLK	section 36
000E 2008h	RSCI10	Control Register 0	SCR0	32	32	2, 3 PCLKA	2ICLK	section 36
000E 200Ch	RSCI10	Control Register 1	SCR1	32	32	2, 3 PCLKA	2ICLK	section 36
000E 2010h	RSCI10	Control Register 2	SCR2	32	32	2, 3 PCLKA	2ICLK	section 36
000E 2014h	RSCI10	Control Register 3	SCR3	32	32	2, 3 PCLKA	2ICLK	section 36
000E 2018h	RSCI10	Control Register 4	SCR4	32	32	2, 3 PCLKA	2ICLK	section 36
000E 201Eh	RSCI10	HBS Support Mode Control Register	HBSCR	8	8	2, 3 PCLKA	2ICLK	section 36
000E 2020h	RSCI10	I^2C Mode Register	SIMR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 2024h	RSCI10	FIFO Control Register	FCR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 202Ch	RSCI10	Manchester Mode Control Register	MMCR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 2030h	RSCI10	DE Signal Control Register	DECR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 2034h	RSCI10	Extended Serial Mode Control Register 0	XCR0	32	32	2, 3 PCLKA	2ICLK	section 36
000E 2038h	RSCI10	Extended Serial Mode Control Register 1	XCR1	32	32	2, 3 PCLKA	2ICLK	section 36
000E 203Ch	RSCI10	Extended Serial Mode Control Register 2	XCR2	32	32	2, 3 PCLKA	2ICLK	section 36
000E 2048h	RSCI10	Status Register	SSR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 204Ch	RSCI10	I^2C Status Register	SISR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 2050h	RSCI10	Receive FIFO Status Register	RFSR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 2054h	RSCI10	Transmit FIFO Status Register	TFSR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 2058h	RSCI10	Manchester Mode Status Register	MMSR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 205Ch	RSCI10	Extended Serial Mode Status Register 0	XSR0	32	32	2, 3 PCLKA	2ICLK	section 36
000E 2060h	RSCI10	Extended Serial Mode Status Register 1	XSR1	32	32	2, 3 PCLKA	2ICLK	section 36
000E 2068h	RSCI10	Status Clear Register	SSCR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 206Ch	RSCI10	I^2C Status Clear Register	SISCR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 2070h	RSCI10	Receive FIFO Status Clear Register	RFSCR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 2074h	RSCI10	Manchester Mode Status Clear Register	MMSCR	32	32	2, 3 PCLKA	2ICLK	section 36

Table 5.1 List of I/O Registers (Address Order) (53 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000E 2078h	RSCI10	Extended Serial Mode Status Clear Register	XSCR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 2080h	RSCI11	Receive Data Register	RDR	32	8, 16, 32	2, 3 PCLKA	2ICLK	section 36
000E 2084h	RSCI11	Transmit Data Register	TDR	32	8, 16, 32	2, 3 PCLKA	2ICLK	section 36
000E 2088h	RSCI11	Control Register 0	SCR0	32	32	2, 3 PCLKA	2ICLK	section 36
000E 208Ch	RSCI11	Control Register 1	SCR1	32	32	2, 3 PCLKA	2ICLK	section 36
000E 2090h	RSCI11	Control Register 2	SCR2	32	32	2, 3 PCLKA	2ICLK	section 36
000E 2094h	RSCI11	Control Register 3	SCR3	32	32	2, 3 PCLKA	2ICLK	section 36
000E 2098h	RSCI11	Control Register 4	SCR4	32	32	2, 3 PCLKA	2ICLK	section 36
000E 209Eh	RSCI11	HBS Support Mode Control Register	HBSCR	8	8	2, 3 PCLKA	2ICLK	section 36
000E 20A0h	RSCI11	I^2C Mode Register	SIMR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 20A4h	RSCI11	FIFO Control Register	FCR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 20ACh	RSCI11	Manchester Mode Control Register	MMCR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 20B0h	RSCI11	DE Signal Control Register	DECR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 20B4h	RSCI11	Extended Serial Mode Control Register 0	XCR0	32	32	2, 3 PCLKA	2ICLK	section 36
000E 20B8h	RSCI11	Extended Serial Mode Control Register 1	XCR1	32	32	2, 3 PCLKA	2ICLK	section 36
000E 20BCh	RSCI11	Extended Serial Mode Control Register 2	XCR2	32	32	2, 3 PCLKA	2ICLK	section 36
000E 20C8h	RSCI11	Status Register	SSR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 20CCh	RSCI11	I^2C Status Register	SISR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 20D0h	RSCI11	Receive FIFO Status Register	RFSR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 20D4h	RSCI11	Transmit FIFO Status Register	TFSR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 20D8h	RSCI11	Manchester Mode Status Register	MMSR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 20DCh	RSCI11	Extended Serial Mode Status Register 0	XSR0	32	32	2, 3 PCLKA	2ICLK	section 36
000E 20E0h	RSCI11	Extended Serial Mode Status Register 1	XSR1	32	32	2, 3 PCLKA	2ICLK	section 36
000E 20E8h	RSCI11	Status Clear Register	SSCR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 20ECh	RSCI11	I^2C Status Clear Register	SISCR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 20F0h	RSCI11	Receive FIFO Status Clear Register	RFSCR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 20F4h	RSCI11	Manchester Mode Status Clear Register	MMSCR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 20F8h	RSCI11	Extended Serial Mode Status Clear Register	XSCR	32	32	2, 3 PCLKA	2ICLK	section 36
000E 2800h	RSPIA0	RSPI Data Register	SPDR	32	8, 16, 32	2, 3 PCLKA	2ICLK	section 41
000E 2804h	RSPIA0	RSPI Clock Delay Register	SPCKD	8	8	2, 3 PCLKA	2ICLK	section 41
000E 2805h	RSPIA0	RSPI Slave Select Negation Delay Register	SSLND	8	8	2, 3 PCLKA	2ICLK	section 41
000E 2806h	RSPIA0	RSPI Next-Access Delay Register	SPND	8	8	2, 3 PCLKA	2ICLK	section 41
000E 2808h	RSPIA0	RSPI Control Register	SPCR	32	32	2, 3 PCLKA	2ICLK	section 41
000E 280Ch	RSPIA0	RSPI Receive-Only Mode Control Register	SPRMCR	8	8	2, 3 PCLKA	2ICLK	section 41
000E 280Dh	RSPIA0	RSPI Receive Data Ready Detect Condition Setting Register	SPDRCSR	8	8	2, 3 PCLKA	2ICLK	section 41
000E 280Eh	RSPIA0	RSPI Pin Control Register	SPPCR	8	8	2, 3 PCLKA	2ICLK	section 41
000E 2810h	RSPIA0	RSPI Slave Select Polarity Register	SSLP	8	8	2, 3 PCLKA	2ICLK	section 41
000E 2811h	RSPIA0	RSPI Bit Rate Register	SPBR	8	8	2, 3 PCLKA	2ICLK	section 41
000E 2813h	RSPIA0	RSPI Sequence Control Register	SPSCR	8	8	2, 3 PCLKA	2ICLK	section 41
000E 2814h	RSPIA0	RSPI Command Register 0	SPCMD0	32	32	2, 3 PCLKA	2ICLK	section 41
000E 2818h	RSPIA0	RSPI Command Register 1	SPCMD1	32	32	2, 3 PCLKA	2ICLK	section 41
000E 281Ch	RSPIA0	RSPI Command Register 2	SPCMD2	32	32	2, 3 PCLKA	2ICLK	section 41
000E 2820h	RSPIA0	RSPI Command Register 3	SPCMD3	32	32	2, 3 PCLKA	2ICLK	section 41
000E 2824h	RSPIA0	RSPI Command Register 4	SPCMD4	32	32	2, 3 PCLKA	2ICLK	section 41
000E 2828h	RSPIA0	RSPI Command Register 5	SPCMD5	32	32	2, 3 PCLKA	2ICLK	section 41
000E 282Ch	RSPIA0	RSPI Command Register 6	SPCMD6	32	32	2, 3 PCLKA	2ICLK	section 41
000E 2830h	RSPIA0	RSPI Command Register 7	SPCMD7	32	32	2, 3 PCLKA	2ICLK	section 41
000E 2840h	RSPIA0	RSPI Data Control Register	SPDCR	16	16	2, 3 PCLKA	2ICLK	section 41

Table 5.1 List of I/O Registers (Address Order) (54 / 55)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
000E 2844h	RSPIA0	RSPI FIFO Control Register	SPFCR	16	16	2, 3 PCLKA	2ICLK	section 41
000E 2851h	RSPIA0	RSPI Sequence Status Register	SPSSR	8	8	2, 3 PCLKA	2ICLK	section 41
000E 2852h	RSPIA0	RSPI Status Register	SPSR	16	16	2, 3 PCLKA	2ICLK	section 41
000E 2858h	RSPIA0	RSPI Transmit FIFO Status Register	SPTFSR	8	8	2, 3 PCLKA	2ICLK	section 41
000E 285Ch	RSPIA0	RSPI Receive FIFO Status Register	SPRFSR	8	8	2, 3 PCLKA	2ICLK	section 41
000E 286Ah	RSPIA0	RSPI Status Clear Register	SPSCLR	16	16	2, 3 PCLKA	2ICLK	section 41
000E 286Ch	RSPIA0	RSPI FIFO Clear Register	SPFCLR	8	8	2, 3 PCLKA	2ICLK	section 41
000E C014h	RIICHS0	Control Register	ICCR	32	32	2, 3 PCLKB	2ICLK	section 38
000E C020h	RIICHS0	Reset Control Register	ICRCR	32	32	2, 3 PCLKB	2ICLK	section 38
000E C024h	RIICHS0	Operating Mode Monitor Register	ICMMR	32	32	2, 3 PCLKB	2ICLK	section 38
000E C060h	RIICHS0	Function Enable Register	ICFER	32	32	2, 3 PCLKB	2ICLK	section 38
000E C064h	RIICHS0	Slave Mode Control Register	ICSCR	32	32	2, 3 PCLKB	2ICLK	section 38
000E C070h	RIICHS0	Reference Clock Control Register	ICRCCR	32	32	2, 3 PCLKB	2ICLK	section 38
000E C074h	RIICHS0	F/S-Mode Bitrate Register	ICFBR	32	32	2, 3 PCLKB	2ICLK	section 38
000E C078h	RIICHS0	Hs-Mode Bitrate Register	ICHBR	32	32	2, 3 PCLKB	2ICLK	section 38
000E C07Ch	RIICHS0	Bus Free Time Setting Register	ICBFTR	32	32	2, 3 PCLKB	2ICLK	section 38
000E C088h	RIICHS0	Output Signal Control Register	ICOCR	32	32	2, 3 PCLKB	2ICLK	section 38
000E C08Ch	RIICHS0	Input Signal Control Register	ICICR	32	32	2, 3 PCLKB	2ICLK	section 38
000E C090h	RIICHS0	Timeout Control Register	ICTOR	32	32	2, 3 PCLKB	2ICLK	section 38
000E C0A0h	RIICHS0	Acknowledgment Bit Control Register	ICACKR	32	32	2, 3 PCLKB	2ICLK	section 38
000E C0A4h	RIICHS0	Clock Stretch Control Register	ICCSCR	32	32	2, 3 PCLKB	2ICLK	section 38
000E C140h	RIICHS0	Condition Generation Request Register	ICCGR	32	32	2, 3 PCLKB	2ICLK	section 38
000E C158h	RIICHS0	Transmit/Receive Data Register	ICDR	32	8, 32	2, 3 PCLKB	2ICLK	section 38
000E C1D0h	RIICHS0	Status Register 2	ICSR2	32	32	2, 3 PCLKB	2ICLK	section 38
000E C1D4h	RIICHS0	Status Detection Enable Register	ICSER	32	32	2, 3 PCLKB	2ICLK	section 38
000E C1D8h	RIICHS0	Status Interrupt Enable Register	ICSIER	32	32	2, 3 PCLKB	2ICLK	section 38
000E C1E0h	RIICHS0	Communication Status Register	ICCSR	32	32	2, 3 PCLKB	2ICLK	section 38
000E C1E4h	RIICHS0	Communication Status Detection Enable Register	ICCSER	32	32	2, 3 PCLKB	2ICLK	section 38
000E C1E8h	RIICHS0	Communication Status Interrupt Enable Register	ICCSIER	32	32	2, 3 PCLKB	2ICLK	section 38
000E C210h	RIICHS0	Bus Status Register	ICBSR	32	32	2, 3 PCLKB	2ICLK	section 38
000E C214h	RIICHS0	Slave Mode Status Register	ICSSR	32	32	2, 3 PCLKB	2ICLK	section 38
000E C2B0h	RIICHS0	Slave Address Register 0	SAR0	32	32	2, 3 PCLKB	2ICLK	section 38
000E C2B4h	RIICHS0	Slave Address Register 1	SAR1	32	32	2, 3 PCLKB	2ICLK	section 38
000E C2B8h	RIICHS0	Slave Address Register 2	SAR2	32	32	2, 3 PCLKB	2ICLK	section 38
000E C330h	RIICHS0	Slave Address Monitor Register 0	SAMR0	32	32	2, 3 PCLKB	2ICLK	section 38
000E C334h	RIICHS0	Slave Address Monitor Register 1	SAMR1	32	32	2, 3 PCLKB	2ICLK	section 38
000E C338h	RIICHS0	Slave Address Monitor Register 2	SAMR2	32	32	2, 3 PCLKB	2ICLK	section 38
000E C380h	RIICHS0	Bit Count Register	ICBCR	32	32	2, 3 PCLKB	2ICLK	section 38
000E C3CCh	RIICHS0	Internal Status Monitor Register	ICIMR	32	32	2, 3 PCLKB	2ICLK	section 38
007F B0E0h	SYSTEM	High-Speed On-Chip Oscillator Trimming Registers 0	HOCOTRR0	32	32	2 ICLK		section 9
007F B0E4h	SYSTEM	High-Speed On-Chip Oscillator Trimming Registers 1	HOCOTRR1	32	32	2 ICLK		section 9
007F B0E8h	SYSTEM	High-Speed On-Chip Oscillator Trimming Registers 2	HOCOTRR2	32	32	2 ICLK		section 9
007F B0ECh	CTSU	CTSU Reference Current Calibration Register	CTSUTRMR	32	32	2 ICLK		section 47
007F C040h	FLASH	Data Flash Memory Access Frequency Setting Register	EEPFCLK	8	8	2 FCLK		section 55
007F E010h	FLASH	Flash Access Status Register	FASTAT	8	8	2 FCLK		section 55
007F E014h	FLASH	Flash Access Error Interrupt Enable Register	FAEINT	8	8	2 FCLK		section 55
007F E018h	FLASH	Flash Ready Interrupt Enable Register	FRDYIE	8	8	2 FCLK		section 55
007F E030h	FLASH	FACI Command Start Address Register	FSADDR	32	32	2 FCLK		section 55
007F E034h	FLASH	FACI Command End Address Register	FEADDR	32	32	2 FCLK		section 55

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

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles		Reference Section
						ICLK ≥ PCLK	ICLK < PCLK	
007F E080h	FLASH	Flash Status Register	FSTATR	32	32	2 FCLK		section 55
007F E084h	FLASH	Flash P/E Mode Entry Register	FENTRYR	16	16	2 FCLK		section 55
007F E08Ch	FLASH	Flash Sequencer Set-Up Initialization Register	FSUINITR	16	16	2 FCLK		section 55
007F E0A0h	FLASH	FACI Command Register	FCMDR	16	16	2 FCLK		section 55
007F E0D0h	FLASH	Data Flash Blank Check Control Register	FBCCNT	8	8	2 FCLK		section 55
007F E0D4h	FLASH	Data Flash Blank Check Status Register	FBCSTAT	8	8	2 FCLK		section 55
007F E0D8h	FLASH	Data Flash Programming Start Address Register	FPSADDR	32	32	2 FCLK		section 55
007F E0DCh	FLASH	Flash Access Window Monitor Register	FAWMON	32	32	2 FCLK		section 55
007F E0E0h	FLASH	Flash Sequencer Processing Switching Register	FCPSR	16	16	2 FCLK		section 55
007F E0E4h	FLASH	Flash Sequencer Processing Clock Frequency Notification Register	FPCKAR	16	16	2 FCLK		section 55
007F E0E8h	FLASH	Start-Up Area Control Register	FSUACR	16	16	2 FCLK		section 55
7400 0000h	QSPIX	Mode Register 0	SPMR0	32	32	Write: up to 13 ICLK, read: 2 ICLK		section 42
7400 0004h	QSPIX	Slave Select Signal Control Register	SPSSCR	32	32	Write: up to 13 ICLK, read: 2 ICLK		section 42
7400 0008h	QSPIX	Clock Control Register	SPOCR	32	32	Write: up to 13 ICLK, read: 2 ICLK		section 42
7400 000Ch	QSPIX	Prefetch Status Register	SPPFSR	32	32	Write: up to 13 ICLK, read: 2 ICLK		section 42
7400 0010h	QSPIX	SPI Data Register	SPDR	32	32	Write: up to 7 ICLK, read: up to 24 ICLK		section 42
7400 0014h	QSPIX	Mode Register 1	SPMR1	32	32	Write: up to 13 ICLK, read: 2 ICLK		section 42
7400 0018h	QSPIX	SPI Status Register	SPSR	32	32	Write: up to 13 ICLK, read: 2 ICLK		section 42
7400 0020h	QSPIX	Special Read Instruction Setting Register	SPRIR	32	32	Write: up to 13 ICLK, read: 2 ICLK		section 42
7400 0024h	QSPIX	Address Mode Register	SPAMR	32	32	Write: up to 13 ICLK, read: 2 ICLK		section 42
7400 0028h	QSPIX	Dummy Cycle Control Register	SPDCR	32	32	Write: up to 13 ICLK, read: 2 ICLK		section 42
7400 0030h	QSPIX	Mode Register 2	SPMR2	32	32	Write: up to 13 ICLK, read: 2 ICLK		section 42
7400 0034h	QSPIX	Port Control Register	SPPCR	32	32	Write: up to 13 ICLK, read: 2 ICLK		section 42
7400 0804h	QSPIX	Upper Address Register	SPUAR	32	32	Write: up to 13 ICLK, read: 2 ICLK		section 42
FE7F 5D00h	OFSM	Endian Select Register	MDE	32	32	1 to 3 ICLK		section 7
FE7F 5D04h	OFSM	Option Function Select Register 0	OFS0	32	32	1 to 3 ICLK		section 7
FE7F 5D08h	OFSM	Option Function Select Register 1	OFS1	32	32	1 to 3 ICLK		section 7
FE7F 5D10h	OFSM	TM Identification Data Register	TMINF	32	32	1 to 3 ICLK		section 7
FE7F 5D20h	OFSM	Bank Select Register	BANKSEL	32	32	1 to 3 ICLK		section 7
FE7F 5D40h	OFSM	Serial Programmer Command Control Register	SPCC	32	32	1 to 3 ICLK		section 7
FE7F 5D48h	OFSM	TM Enable Flag Register	TMEF	32	32	1 to 3 ICLK		section 7
FE7F 5D50h	OFSM	OCD/Serial Programmer ID Setting Register	OSIS	128	32	1 to 3 ICLK		section 7
FE7F 5D64h	OFSM	Flash Access Window Setting Register	FAW	32	32	1 to 3 ICLK		section 7
FE7F 5D70h	OFSM	ROM Code Protection Register	ROMCODE	32	32	1 to 3 ICLK		section 7
FE7F 7D7Ch	TEMPSCONST	Temperature Sensor Calibration Data Register	TSCDR	32	32	1 to 3 ICLK		section 51
FE7F 7D90h	FLASHCONST	Unique ID Register 0	UIDR0	32	32	1 to 3 ICLK		section 55
FE7F 7D94h	FLASHCONST	Unique ID Register 1	UIDR1	32	32	1 to 3 ICLK		section 55
FE7F 7D98h	FLASHCONST	Unique ID Register 2	UIDR2	32	32	1 to 3 ICLK		section 55
FE7F 7D9Ch	FLASHCONST	Unique ID Register 3	UIDR3	32	32	1 to 3 ICLK		section 55

Note 1. When the register is accessed while the USB is operating, a delay may be generated in accessing.
Note 2. The address must end with 0h, 4h, 8h, or Ch when access is made in 32-bit units. The address must end with 0h, 2h, 4h, 6h, 8h, Ah, Ch, or Eh when access is made in 16-bit units.

6. Resets

6.1 Overview

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

Table 6.1 lists the reset names and sources.

Table 6.1 Reset Names and Sources

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

Note 1. For details on the voltages to be monitored (VPOR, Vdet0, Vdet1, and Vdet2), refer to section 8, Voltage Detection Circuit (LVDA) and section 56, 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 (1/2)

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

Table 6.2 Targets to be Initialized by Each Reset Source (2/2)

Targets to be Initialized	Reset Source								
	RES# Pin Reset	Power-On Reset	Voltage-Monitoring 0 Reset	Independent Watchdog Timer Reset	Watchdog Timer Reset	Voltage-Monitoring 1 Reset	Voltage-Monitoring 2 Reset	Deep Software Standby Reset	Software Reset
Registers other than the above, CPU, and internal state	✓	✓	✓	✓	✓	✓	✓	✓	✓

✓: Targets to be initialized, —: No change occurs.

Note 1. Some control bits are initialized by all types of resets. For details on the target bits, refer to section 31, Realtime Clock (RTCd).

Note 2. The operating mode is determined by the level of the mode setting pins when the reset is released. For details, refer to section 3, Operating Modes.

When a reset is canceled, the reset exception handling starts. For details on the reset exception handling, refer to section 14, Exception Handling.

Table 6.3 lists the pin related to the reset.

Table 6.3 Pin Related to Reset

Pin Name	I/O	Function
RES#	Input	Reset pin

6.2 Register Descriptions

6.2.1 Reset Status Register 0 (RSTSR0)

Address(es): 0008 C290h

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

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

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

Note 2. Only 0 can be written to clear the flag.

PORF Flag (Power-On Reset Detect Flag)

The PORF flag indicates that a power-on reset has occurred.

[Setting condition]

- When a power-on reset occurs.

[Clearing conditions]

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

LVD0RF Flag (Voltage-Monitoring 0 Reset Detect Flag)

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

- When a voltage-monitoring 0 reset occurs.

[Clearing conditions]

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

LVD1RF Flag (Voltage-Monitoring 1 Reset Detect Flag)

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

- When a voltage-monitoring 1 reset occurs.

[Clearing conditions]

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

LVD2RF Flag (Voltage-Monitoring 2 Reset Detect Flag)

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

- When a voltage-monitoring 2 reset occurs.

[Clearing conditions]

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

DPSRSTF Flag (Deep Software Standby Reset Flag)

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

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

[Clearing conditions]

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

6.2.2 Reset Status Register 1 (RSTSR1)

Address(es): 0008 C291h

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

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

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

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

CWSF Flag (Cold/Warm Start Determination Flag)

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

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

[Clearing condition]

- When a reset listed in Table 6.2 occurs.

6.2.3 Reset Status Register 2 (RSTSR2)

Address(es): 0008 00C0h

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

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

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

IWDTRF Flag (Independent Watchdog Timer Reset Detect Flag)

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

- When an independent watchdog timer reset occurs.

[Clearing conditions]

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

WDTRF Flag (Watchdog Timer Reset Detect Flag)

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

- When a watchdog timer reset occurs.

[Clearing conditions]

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

SWRF Flag (Software Reset Detect Flag)

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

- When a software reset occurs.

[Clearing conditions]

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

6.2.4 Software Reset Register (SWRR)

Address(es): 0008 00C2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after 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	—	Software Reset	Writing A501h resets the MCU. These bits are read as 0000h.	R/W

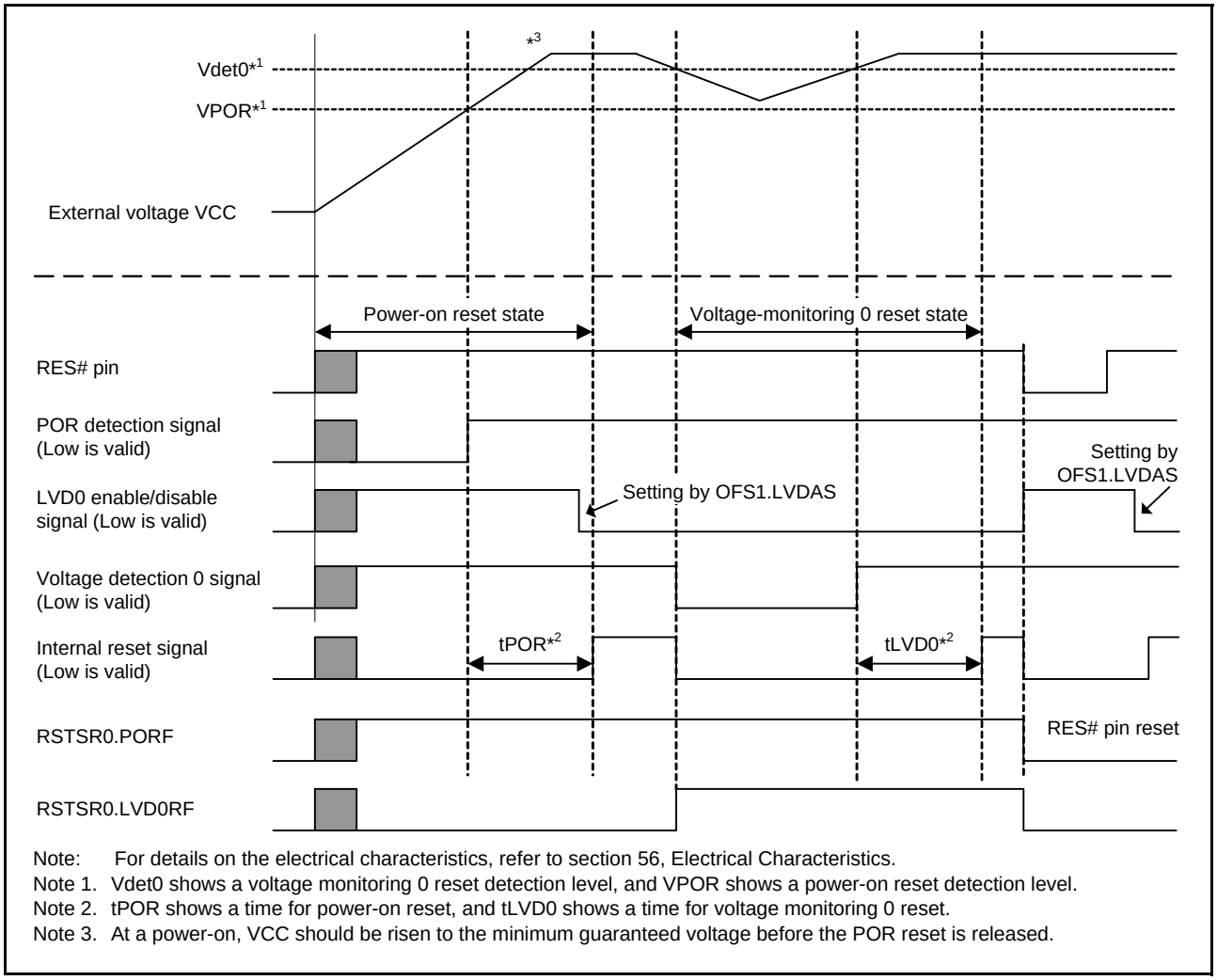
6.3 Operation

6.3.1 RES# Pin Reset

This is a reset generated by the RES# pin.
When the RES# pin is driven low, all the processing in progress is aborted and the MCU enters a reset state.
In order to unfailingly reset the MCU, the RES# pin should be held low for the specified power supply stabilization time at a power-on.
When the RES# pin is driven high from low, the internal reset is canceled after the post-RES# cancelation wait time (tRESWT) has elapsed, and then the CPU starts the reset exception handling.
For details, refer to section 56, Electrical Characteristics.

6.3.2 Power-On Reset and Voltage Monitoring 0 Reset

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



Note: For details on the electrical characteristics, refer to section 56, Electrical Characteristics.
Note 1. Vdet0 shows a voltage monitoring 0 reset detection level, and VPOR shows a power-on reset detection level.
Note 2. tPOR shows a time for power-on reset, and tLVD0 shows a time for voltage monitoring 0 reset.
Note 3. At a power-on, VCC should be risen to the minimum guaranteed voltage before the POR reset is released.

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

6.3.3 Voltage Monitoring 1 Reset and Voltage Monitoring 2 Reset

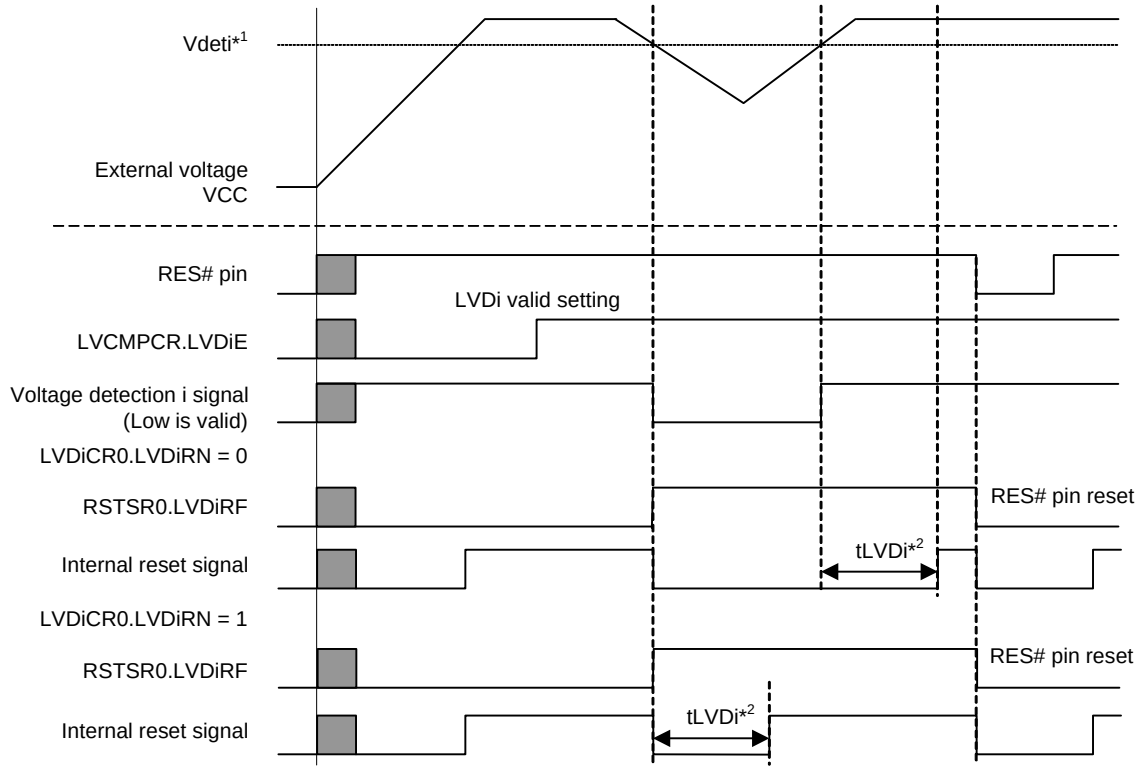
The voltage monitoring 1 reset and voltage monitoring 2 reset are internal resets generated by the voltage monitoring circuit.

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

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

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

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



Note: For details on the electrical characteristics, refer to section 56, Electrical Characteristics.
Note 1. Vdeti shows a detection level of voltage monitoring 1 reset and voltage monitoring 2 reset. (i = 1, 2)
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 released by an interrupt.
When an interrupt for releasing from deep software standby mode is generated, a deep software standby reset is generated. The deep software standby reset is negated after recovery time from deep software standby mode (tDSBY) has elapsed. At the same time, deep software standby mode is also released.
When the wait time after recovery from deep software standby mode (tDSBYWT) has elapsed after deep software standby mode has been released, the internal reset is negated and the CPU starts the reset exception handling.
For details of the deep software standby reset, refer to section 11, Low Power Consumption.

6.3.5 Independent Watchdog Timer Reset

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

6.3.6 Watchdog Timer Reset

The watchdog-timer reset is an internal reset from the watchdog timer.
Output of the independent watchdog timer reset from the independent watchdog timer can be selected by settings in the IWDT reset control register (IWDTRCR) or option function select register 0 (OFS0).
When output of the watchdog timer reset is selected, a watchdog timer reset is generated if the watchdog timer underflows, or if data is written when refresh operation is disabled. When the internal reset time (tRESW2) has elapsed after the watchdog timer reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.
For details on the watchdog timer reset, refer to section 32, 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.

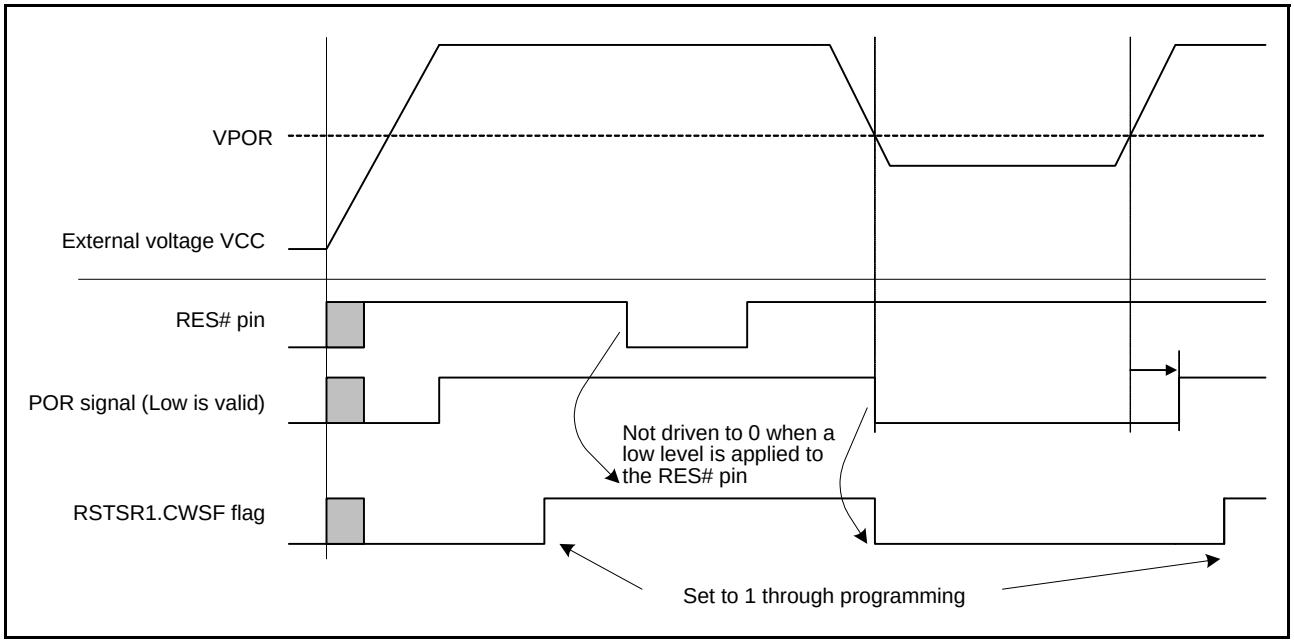


Figure 6.3 Example of Cold/Warm Start Determination Operation

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.

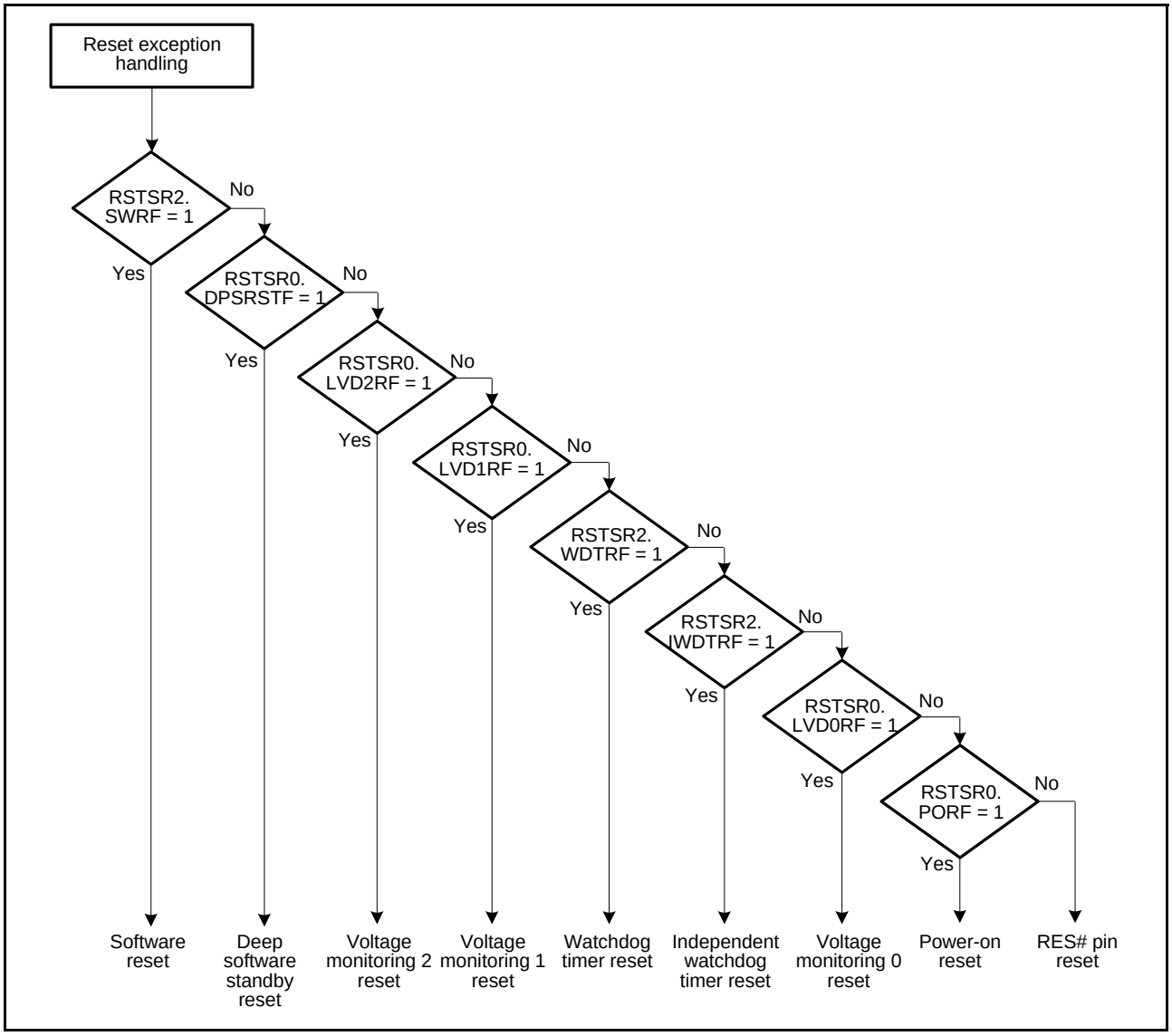


Figure 6.4 Example of Reset Generation Source Determination Flow

7. Option-Setting Memory (OFSM)

7.1 Overview

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

- Serial programmer command control register (SPCC)
- OCD/serial programmer ID setting register (OSIS)
- Option function select register 0 (OFS0)
- Option function select register 1 (OFS1)
- Endian select register (MDE)
- TM enable flag register (TMEF)
- TM identification data register (TMINF)
- Bank select register (BANKSEL)
- Flash access window setting register (FAW)
- ROM Code Protection Register (ROMCODE)

The option-setting memory (configuration setting area) determines the state of this MCU after a reset.
The method of setting the option-setting memory is different from that of the I/O registers. For details, refer to section 7.5, Setting the Option-Setting Memory.
Figure 7.1 shows the option-setting memory.

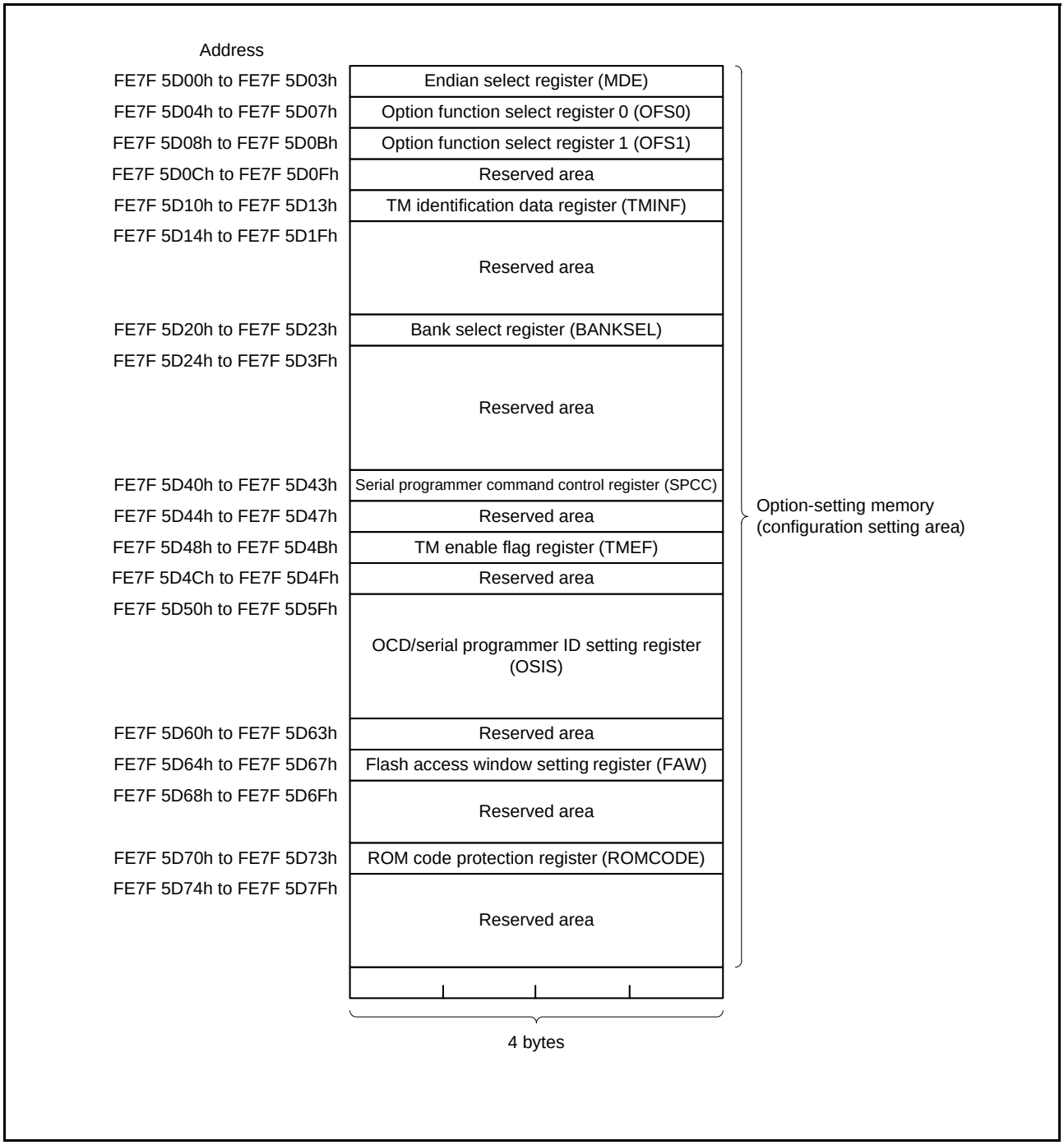


Figure 7.1 Option-Setting Memory

7.2 Register Descriptions

7.2.1 Serial Programmer Command Control Register (SPCC)

Address(es): OFSM.SPCC FE7F 5D40h

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

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

Bit	Symbol	Bit Name	Description	R/W
b16 to b0	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b17	OCDE	On-Chip Debugger Connection Enable	0: Connection of an on-chip debugger is prohibited after a reset. 1: Connection of an on-chip debugger is permitted after a reset.	R
b26 to b18	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b27	SPE	Serial Programmer Connection Enable	0: Connection of a serial programmer is prohibited after a reset. 1: Connection of a serial programmer is permitted after a reset.	R
b31 to b28	—	Reserved	When reading, these bits return to the value written by the user. The write value should be 1.	R

Note 1. For products in which the option-setting memory has not been specifically set, the value is FFFF FFFFh. The value will be set by the user.

This register is used to permit or prohibit the connection of an on-chip debugger or the connection of a serial programmer.

OCDE Bit (On-Chip Debugger Connection Enable)

This bit enables or disables the connection of an on-chip debugger.

SPE Bit (Serial Programmer Connection Enable)

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

7.2.2 OCD/Serial Programmer ID Setting Register (OSIS)

This register is used to store the control code or ID code for ID code protection of the OCD/serial programmer.
After the OCD/serial programmer sends a control code or ID code, it is tested for a match with the value stored in this register.
Connection to the OCD/serial programmer can proceed if the codes match and cannot proceed if they do not.
For the products without the setting of the option-setting memory, the value after a reset for ID code 1/control code to ID code 16 is FFh. The value will be set by the user.

Address	Bit 31			Bit 0
FE7F 5D50h to FE7F 5D53h	ID Code 4	ID Code 3	ID Code 2	ID Code 1/Control Code
FE7F 5D54h to FE7F 5D57h	ID Code 8	ID Code 7	ID Code 6	ID Code 5
FE7F 5D58h to FE7F 5D5Bh	ID Code 12	ID Code 11	ID Code 10	ID Code 9
FE7F 5D5Ch to FE7F 5D5Fh	ID Code 16	ID Code 15	ID Code 14	ID Code 13

ID Code 1/Control Code to ID Code 16

The control code or ID code for ID code protection of an OCD/serial programmer is stored in this register.
ID code 1 is used as a control code for connection to a serial programmer and as an ID code for connection to an OCD.
For details of the control code, refer to section 7.4, Settings of the Option-Setting Memory and Reading, Programming, and Erasure.

7.2.3 Option Function Select Register 0 (OFS0)

Address(es): OFSM.OFS0 FE7F 5D04h

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

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

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

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

Note 1. For products in which the option-setting memory has not been specifically set, the value is FFFF FFFFh. The value will be set by the user.

IWDTSTRT Bit (IWDT Start Mode Select)

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

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

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

For details, refer to section 33, Independent Watchdog Timer (IWDTa).

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

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

For details, refer to section 33, 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 33, 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 33, Independent Watchdog Timer (IWDTa).

IWDTRSTIRQS Bit (IWDT Reset Interrupt Request Select)

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

IWDTSLCSTP Bit (IWDT Sleep Mode Count Stop Control)

This bit selects to stop counting when entering sleep, software standby, deep software standby, or all-module clock stop mode.
For details, refer to section 33, Independent Watchdog Timer (IWDTa).

WDTSTRT Bit (WDT Start Mode Select)

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

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

These bits select the timeout period, i.e. the time it takes for the down-counter to underflow, as 1024, 4096, 8192, or 16384 cycles of the frequency-divided clock set by the WDTCKS[3:0] bits. The time (number of PCLKB cycles) it takes to underflow after a refresh operation is determined by a combination of the WDTCKS[3:0] bits and WDTTOPS[1:0] bits.
For details, refer to section 32, Watchdog Timer (WDTA).

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

These bits select, from 1/4, 1/64, 1/128, 1/512, 1/2048, and 1/8192, the division ratio of the prescaler to divide the frequency of PCLKB. Using the setting of these bits together with the WDTTOPS[1:0] bit setting, the WDT counting period can be set from 4096 to 134217728 PCLKB cycles.
For details, refer to section 32, 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 32, 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 32, Watchdog Timer (WDTA).

WDTRSTIRQS Bit (WDT Reset Interrupt Request Select)

The setting of this bit selects the operation on an underflow of the down-counter or generation of a refresh error. A watchdog timer reset, a non-maskable interrupt request, or an interrupt request is selectable.
For details, refer to section 32, Watchdog Timer (WDTA).

7.2.4 Option Function Select Register 1 (OFS1)

Address(es): OFSM.OFS1 FE7F 5D08h

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

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

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

Note 1. For products in which the option-setting memory has not been specifically set, the value is FFFF FFFFh. The value will be set by the user.

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

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

LVDAS Bit (Voltage Detection 0 Circuit Start)

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

HOCOEN Bit (HOCO Oscillation Enable)

This bit selects whether the HOCO oscillation enable bit is effective or not after a reset.

Setting the HOCOEN bit to 0 allows the HOCO oscillation to be started before the CPU starts operation, and therefore reduces the waiting time for oscillation stabilization.

Note that even if the HOCOEN bit is set to 0, the system clock source is not switched to HOCO. The system clock source is switched to HOCO only by modifying the clock source select bits (SCKCR3.CKSEL[2:0]) from the CPU.

7.2.5 Endian Select Register (MDE)

Address(es): OFSM.MDE FE7F 5D00h

	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
	—	—	—	—	—	—	—	—	—	BANKMD[2:0]			—	MDE[2:0]		
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	MDE[2:0]	Endian Select	b2 b0 0 0 0: Big endian 1 1 1: Little endian Settings other than above are prohibited.	R
b3	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R
b6 to b4	BANKMD[2:0]	Bank Mode Select	b6 b4 0 0 0: Dual mode 1 1 1: Linear mode Settings other than above are prohibited.	R
b31 to b7	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. For products in which the option-setting memory has not been specifically set, the value is FFFF FFFFh. The value will be set by the user.

This register selects the endian for the CPU and bank mode of the dual bank function of the code flash memory.

MDE[2:0] Bits (Endian Select)

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

BANKMD[2:0] Bits (Bank Mode Select)

These bits select bank mode of the dual bank function of the code flash memory. These bits cannot be rewritten while the TM function is enabled. Set the value of these bits while the TM function is disabled.

7.2.6 TM Enable Flag Register (TMEF)

Address(es): OFSM.TMEF FE7F 5D48h

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

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

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b26 to b24	TMEF[2:0]	TM Enable	b26 b24 0 0 0: The TM function for blocks 8 and 9 in the code flash memory is enabled. 1 1 1: The TM function for blocks 8 and 9 in the code flash memory is disabled. Settings other than above are prohibited.	R
b27	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R
b30 to b28	TMEFDB[2:0]	Dual-Bank TM Enable	b30 b28 0 0 0: The TM function for blocks 46 and 47 in the code flash memory is enabled in dual mode. 1 1 1: The TM function for blocks 46 and 47 in the code flash memory is disabled in dual mode. Settings other than above are prohibited.	R
b31	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R

Note 1. For products in which the option-setting memory has not been specifically set, the value is FFFF FFFFh. The value will be set by the user.

This register is used to enable the TM function in the code flash memory.
To enable the TM function, refer to section 55.15.23, Configuration Programming Command. When the TMEF[2:0] bits or TMEFDB[2:0] bits are rewritten with the TM function enabled, rewriting the bits is ignored.
To disable the TM function, refer to section 55.15.21, Configuration Clearing Command.

TMEF[2:0] Bits (TM Enable)

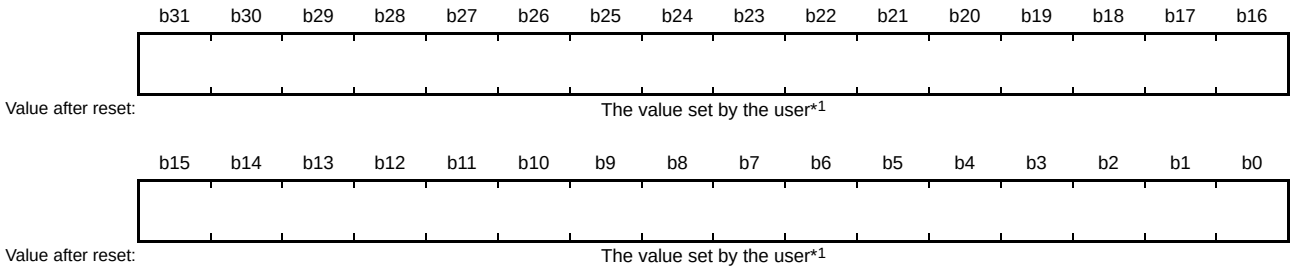
These bits enable or disable the TM function in the code flash memory. When enabling the TMEF[2:0] bits (000b) in dual mode, also enable the TMEFDB[2:0] (000b).

TMEFDB[2:0] Bits (Dual-Bank TM Enable)

These bits enable or disable the TM function of the code flash memory in dual mode. Set these bits to disable dual mode (111b) if linear mode is to be used.
When enabling the TMEFDB[2:0] bits, also enable the TMEF[2:0] bits.
When the TMEF[2:0] bits are 111b (with the TM function disabled), this setting is disabled.

7.2.7 TM Identification Data Register (TMINF)

Address(es): OFSM.TMINF FE7F 5D10h



Note 1. For products in which the option-setting memory has not been specifically set, the value is FFFF FFFFh. The value will be set by the user.

The user can store any desired 32-bit value in this register.

This register is used to store codes that identify the program stored in the TM-target area.

When the TMINF register is rewritten by serial programming while the TM function is enabled, rewriting this register is ignored. To erase the contents of the TMINF register, refer to section 55.15.21, Configuration Clearing Command.

7.2.8 Bank Select Register (BANKSEL)

Address(es): OFSM.BANKSEL FE7F 5D20h

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	BANKSWP[2:0]		
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	BANKSWP[2:0]	Startup Bank Switch	b2 b0 0 0 0: These bits specify the address range of bank 1 from FFF0 0000h to FFFF FFFFh and bank 0 from FFE0 0000h to FFEF FFFFh. 1 1 1: These bits specify the address range of bank 1 from FFE0 0000h to FFEF FFFFh and bank 0 from FFF0 0000h to FFFF FFFFh. Settings other than above are prohibited.	R
b31 to b3	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. For products in which the option-setting memory has not been specifically set, the value is FFFF FFFFh. The value will be set by the user.

This register selects the startup bank of the program when the code flash memory is in dual mode.

BANKSWP[2:0] Bits (Startup Bank Switch)

These bits select the address range of the bank 0 and bank 1 of the code flash memory when the code flash memory is in dual mode.

Selecting the addresses of bank 0 and bank 1 selects the program to be started up.

The setting of these bits in linear mode is invalid.

For details of the startup bank selection, refer to section 55.10.5.2, Selecting the Startup Bank.

7.2.9 Flash Access Window Setting Register (FAW)

Address(es): OFSM.FAW FE7F 5D64h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	BTFLG	—	—	—	FAWE[11:0]											
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FSPR	—	—	—	FAWS[11:0]											
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	FAWS[11:0]	Flash Access Window Start Address	Flash access window start address	R
b14 to b12	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b15	FSPR	Access Window Protection	0: With protection 1: Without protection	R
b27 to b16	FAWE[11:0]	Flash Access Window End Address	Flash access window end address	R
b30 to b28	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b31	BTFLG	Start-up Area Select	0: FFFF C000h to FFFF DFFFh are used as the start-up area 1: FFFF E000h to FFFF FFFFh are used as the start-up area	R

Note 1. For products in which the option-setting memory has not been specifically set, the value is FFFF FFFFh. The value will be set by the user.

This register is used to set the write protection bit and start-up area select bit for setting the flash access window start address, flash access window end address, and access window.

FAWS[11:0] Bits (Flash Access Window Start Address)

These bits are used to set the access window start address.
The access window can be set in block units.
Set the FAWS[10:0] bits to b23 to b13 of the access window start address. Set the FAWS[11] bit to 0.

FSPR Bit (Access Window Protection)

Setting the FSPR bit protects the following operations.

- Setting the areas including the FAW register by using the configuration setting command of the FACI commands.
- Setting the areas including the FAW register by using the configuration program command in boot mode
- Erasing the option-setting memory by using the configuration clearing command in boot mode
- Changing the setting of the start-up area protection by using the FSUACR register.
- Erasing the all data in the flash memory when results of judgment did not match 3 times in a row while the control code is set to 45h in boot mode.

Once 0 is written to this bit, the bit can never be restored to 1.
Therefore, the access window and the start-up area protection never be set again or the TM function never be disabled once it has been enabled.
Exercise extra caution when handling the FSPR bit.

FAWE[11:0] Bits (Flash Access Window End Address)

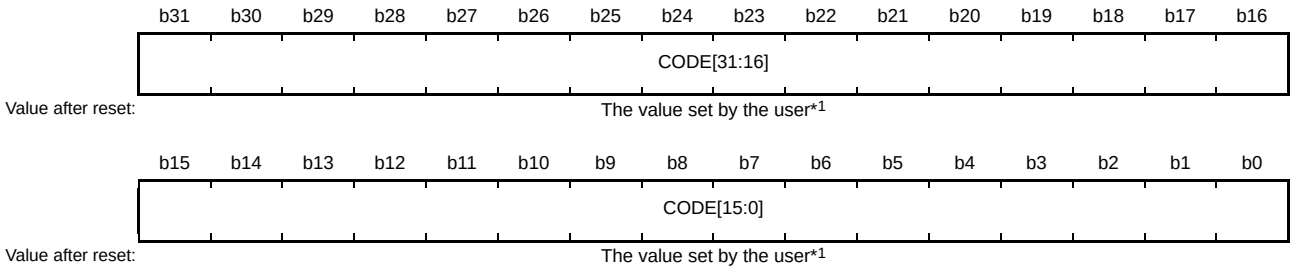
These bits are used to set the access window end address.
The access window can be set in block units.
Set the FAWE[10:0] bits to b23 to b13 of the access window end address. Set the FAWE[11] bit to 0. However, if the access window end address is 0000 0000h, set the FAWE[11:0] bits to 800h.

BTFLG Bit (Start-up Area Select)

This bit is used to set whether the start-up area is switched by using the start-up program protection. In dual mode (the MDE.BANKMD[2:0] bits are 000b), write 1 to this bit.
For details, refer to section 55.10.3, Start-Up Program Protection.

7.2.10 ROM Code Protection Register (ROMCODE)

Address(es): OFSM.ROMCODE FE7F 5D70h



Bit	Symbol	Bit Name	Description	R/W
b31 to b0	CODE[31:0]	ROM Code	0000 0000h: ROM code protection enabled (ROM code protection 1) 0000 0001h: ROM code protection enabled (ROM code protection 2) Other than above: ROM code protection disabled	R

Note 1. For products in which the option-setting memory has not been specifically set, the value is FFFF FFFFh. The value will be set by the user.

This register is used to disable reading, programming, and erasure of the flash memory by a parallel programmer being used for off-board programming.

The ROM code in flash memory is a 32-bit code.

Table 7.1 shows the specifications for ROM code protection.

For release from ROM code protection, write FFFF FFFFh (code protection is disabled) to the ROM code by using the configuration setting command of the self-programming, or by using the configuration setting command of the boot mode, or erase the ROM code by using the configuration clearing command. The user can store any desired 32-bit value in the register.

Table 7.1 Specifications for ROM Code Protection

ROM Code	State of Protection	Operations at the Time of Connection with the Parallel Programmer
0000 0000h	ROM code protection enabled (ROM code protection 1)	Reading, programming, and erasure of the code flash memory is prohibited.
0000 0001h	ROM code protection enabled (ROM code protection 2)	Reading from the code flash memory is prohibited.
Other than above	ROM code protection disabled	Reading, programming, and erasure of the code flash memory is permitted.

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

Table 7.2 shows programming and erasure of the option-setting memory in the individual operating modes.

Table 7.2 Programming and Erasure of the Option-Setting Memory in Individual Operating Modes

Option-Setting Memory	ROMCODE Setting Value	Boot Mode (SCI Interface, USB Interface, FINE Interface)		Self-Programming		Parallel Programmer	
		Programming	Erasure	Programming	Erasure	Programming	Erasure
ROM code protection register (ROMCODE)	0000 0000h	✓*1	✓*1	✓*2	×	×	×
	0000 0001h	✓*1	✓*1	✓*2	×	×*3	✓*4, *5
	Other than above	✓*1	✓*1	✓*2	×	✓*4	✓*4
SPCC register, OSIS register, MDE register, OFS0 register, OFS1 register, TMEF register, TMINF register, BANKSEL register, and FAW register	0000 0000h	✓*1	✓*1	✓*2	×	×	×
	0000 0001h	✓*1	✓*1	✓*2	×	✓*4	✓*4, *5
	Other than above	✓*1	✓*1	✓*2	×	✓*4	✓*4

✓: Possible

×: Not possible

Note 1. The commands for boot mode (SCI, USB, and FINE interfaces) are used for programming or erasure. For details, refer to section 55.11, Boot Mode.

Note 2. The configuration setting command is used for programming. For how to use the configuration setting command, refer to section 55.8.3.14, Configuration Set Command.

Note 3. The ROM code cannot be programmed when the ROM code protection is set.

Note 4. The parallel programmer is used for programming and erasure. For details, refer to the manual of the parallel programmer you are using.

Note 5. It is erasable when the code flash memory is blank.

7.4 Settings of the Option-Setting Memory and Reading, Programming, and Erasure

Table 7.3 shows the settings of the option-setting memory and reading, programming, and erasure when the MCU is connected to a serial programmer.
Table 7.4 shows the settings of the option-setting memory and judgment on ID codes when the MCU is connected to an OCD.

Table 7.3 Settings of the Option-Setting Memory and Reading, Programming, and Erasure When the MCU is Connected to a Serial Programmer

No.	SPCC.SPE	OSIS (Control Code)	OSIS (ID Code 2 to 16)	Connection to a Serial Programmer	Reading, Programming and Erasure after the Connection to a Serial Programmer
1	0	Any value	Any value	Connection prohibited	—
2	1	45h	Any value	Control codes or ID codes matched: Transition to the command waiting phase Control codes or ID codes unmatched: Transition to the state of waiting for a serial programming ID code check command again. However, when results of judgment do not match 3 times in a row, the all data in the flash memory will be erased*1	Reading permitted, programming permitted, erasure permitted
3	1	Other than 45h	Any value	ID codes matched: Transition to the command waiting phase ID codes unmatched: Transition to the state of waiting for a serial programming ID code check command again.	Reading permitted, programming permitted, erasure permitted

Note 1. When the FAW.FSPR bit is 0, the blocks in those areas are not erased.

Table 7.4 Settings of the Option-Setting Memory and Judgment on ID Codes When the MCU is Connected to an OCD

No.	SPCC.SPE	SPCC.OCDE	OSIS (ID Code 1)	OSIS (ID Code 2 to 16)	Connection to an OCD
1	—	1	Any value	Any value	ID codes matched: Connection to an OCD is permitted ID codes unmatched: Waiting for input of ID code
2	—	0	—	—	Connection to an OCD is prohibited (this is independent of the state of ID code matching).

7.5 Setting the Option-Setting Memory

7.5.1 Allocation of Data in the Option-Setting Memory

Data for programming in the option-setting memory should be allocated to the addresses shown in Figure 7.1.
An example of source code for setting the option-setting memory is shown below.

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

Setting F7FFFFFFh in the serial programmer command control register (SPCC)
.ORG 0FE7F5D40H
.LWORD 0F7FFFFFFH

Setting the following ID codes in the OCD/serial programmer ID setting register (OSIS)
ID code 1/control code = FFh, ID code 2 = 02h, ID code 3 = 03h, ID code 4 = 04h,
ID code 5 = 05h, ID code 6 = 06h, ID code 7 = 07h, ID code 8 = 08h,
ID code 9 = 09h, ID code 10 = 0Ah, ID code 11 = 0Bh, ID code 12 = 0Ch,
ID code 13 = 0Dh, ID code 14 = 0Eh, ID code 15 = 0Fh, ID16 = 10h
.ORG 0FE7F5D50H
.LWORD 0040302FFH, 008070605H, 00C0B0A09H, 0100F0E0DH

Setting EF67BA5Dh in the option function select register 0 (OFS0)
.ORG 0FE7F5D04H
.LWORD 0EF67BA5DH

Setting FFFFFEFAh in the option function select register 1 (OFS1)
.ORG 0FE7F5D08H
.LWORD 0FFFFFEFAH

Setting FFFFFFF8h in the endian select register (MDE)
.ORG 0FE7F5D00H
.LWORD 0FFFFFFF8H

Setting F5A5725Ah in the flash access window setting register (FAW)
.ORG 0FE7F5D64H
.LWORD 0F5A5725AH

7.6 Usage Note

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

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

8. Voltage Detection Circuit (LVDA)

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

8.1 Overview

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

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

Table 8.1 Voltage Detection Circuit Specifications

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

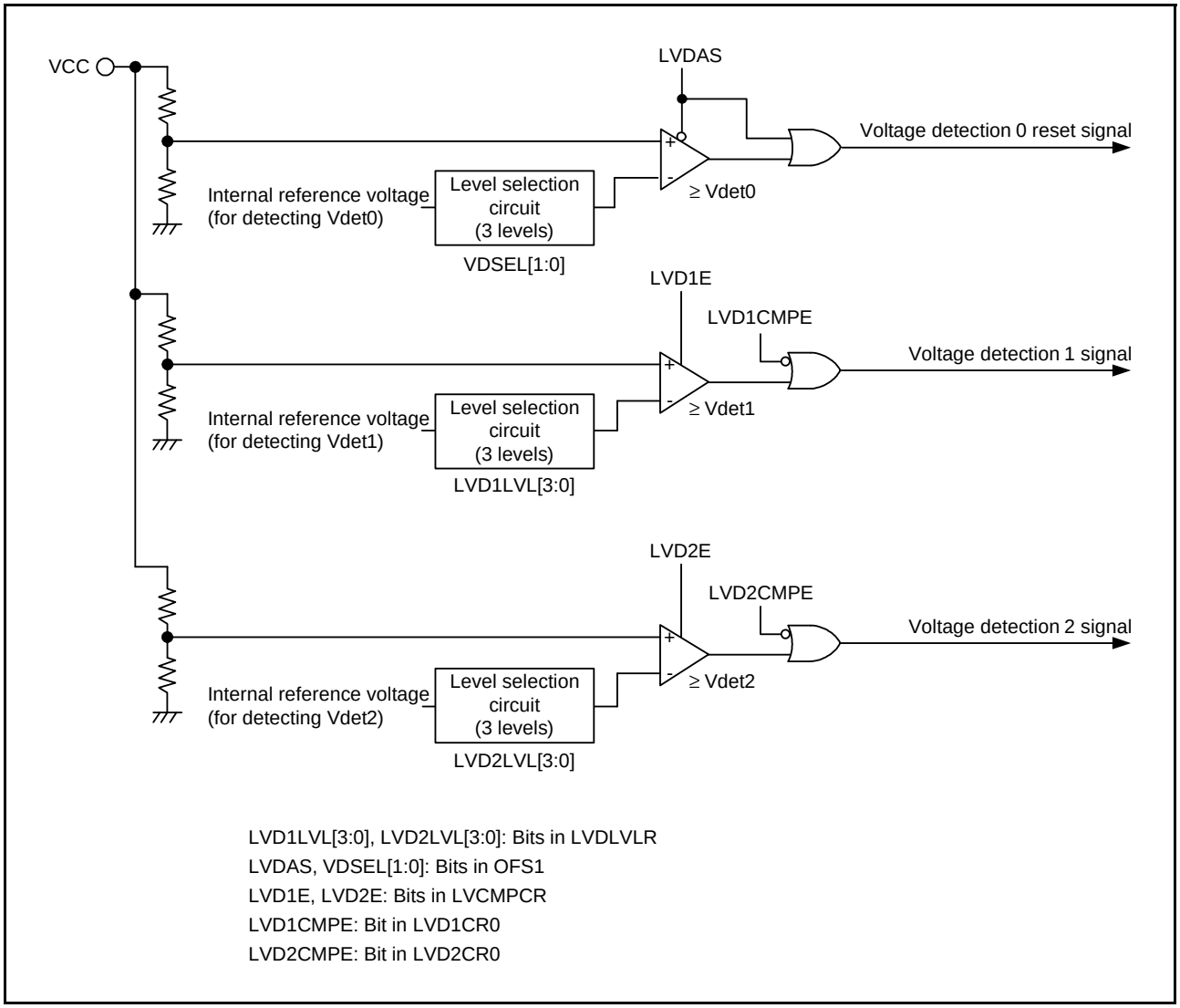


Figure 8.1 Block Diagram of Voltage Detection Circuit

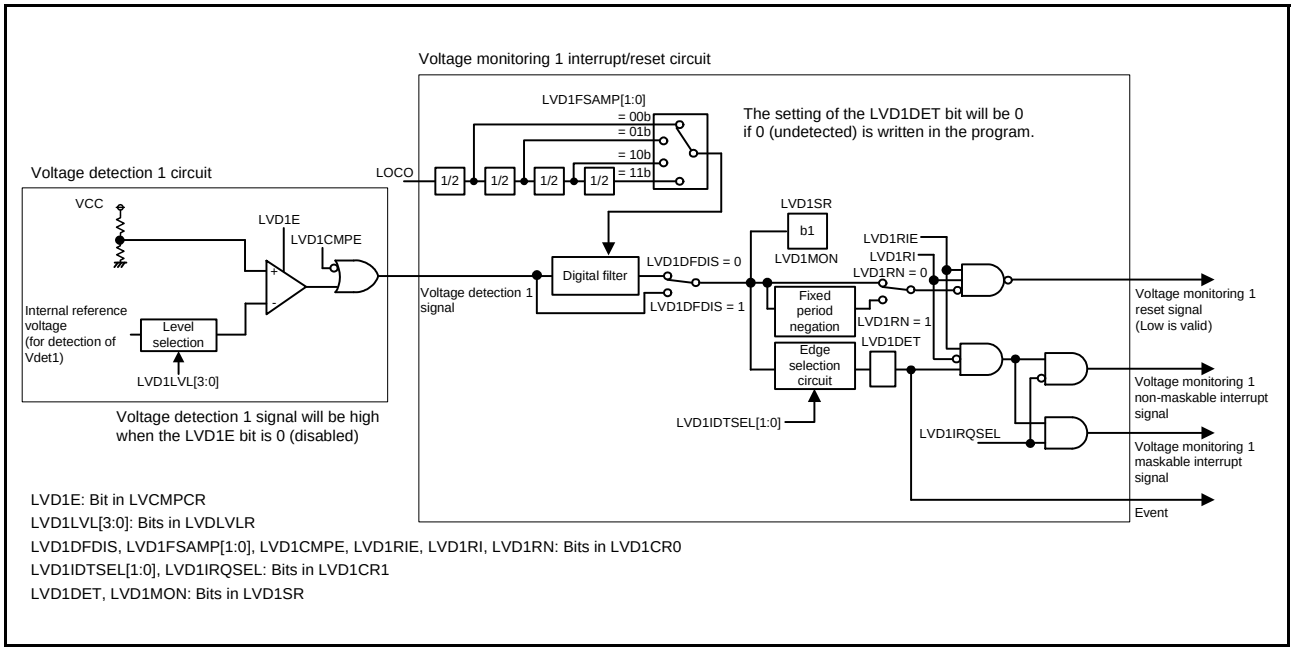


Figure 8.2 Block Diagram of Voltage Monitoring 1 Interrupt/Reset Circuit

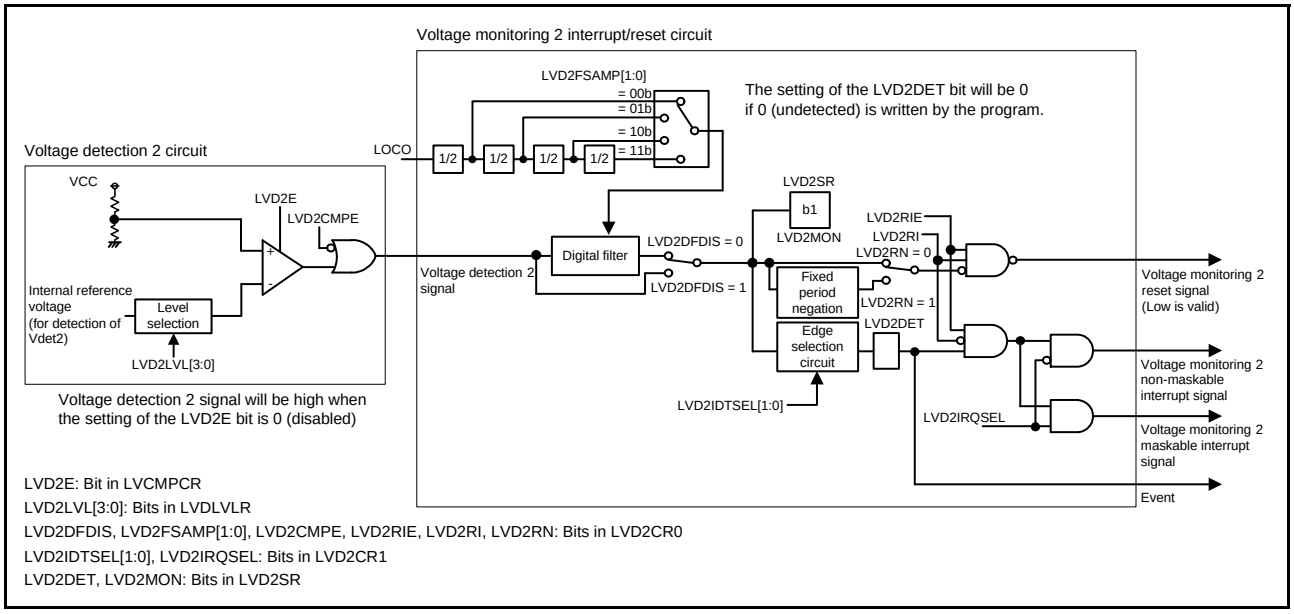


Figure 8.3 Block Diagram of Voltage Monitoring 2 Interrupt/Reset Circuit

8.2 Register Descriptions

8.2.1 Voltage Monitoring 1 Circuit Control Register 1 (LVD1CR1)

Address(es): 0008 00E0h

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

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

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

8.2.2 Voltage Monitoring 1 Circuit Status Register (LVD1SR)

Address(es): 0008 00E1h

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

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

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

LVD1DET Flag (Voltage Monitoring 1 Voltage Change Detection Flag)

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

LVD1MON Flag (Voltage Monitoring 1 Signal Monitor Flag)

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

8.2.3 Voltage Monitoring 2 Circuit Control Register 1 (LVD2CR1)

Address(es): 0008 00E2h

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

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

Note 1. When enabling maskable interrupts, do not change the value of the NMIER.LVD2EN bit on the ICU side from the reset state.

8.2.4 Voltage Monitoring 2 Circuit Status Register (LVD2SR)

Address(es): 0008 00E3h

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

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

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

LVD2DET Flag (Voltage Monitoring 2 Voltage Change Detection Flag)

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

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

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

LVD2MON Flag (Voltage Monitoring 2 Signal Monitor Flag)

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

8.2.5 Voltage Monitoring Circuit Control Register (LVCMPCR)

Address(es): 0008 C297h

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

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

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

LVD1E Bit (Voltage Detection 1 Enable)

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

LVD2E Bit (Voltage Detection 2 Enable)

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

8.2.6 Voltage Detection Level Select Register (LVDLVLR)

Address(es): 0008 C298h

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

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	LVD1LVL[3:0]	Voltage Detection 1 Level Select (Standard voltage during drop in voltage)	b3 b0 1 0 0 1: 2.99 V (Vdet1_1) 1 0 1 0: 2.92 V (Vdet1_2) 1 0 1 1: 2.85 V (Vdet1_3) Settings other than above are prohibited.	R/W
b7 to b4	LVD2LVL[3:0]	Voltage Detection 2 Level Select (Standard voltage during drop in voltage)	b7 b4 1 0 0 1: 2.99 V (Vdet2_1) 1 0 1 0: 2.92 V (Vdet2_2) 1 0 1 1: 2.85 V (Vdet2_3) Settings other than above are prohibited.	R/W

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

8.2.7 Voltage Monitoring 1 Circuit Control Register 0 (LVD1CR0)

Address(es): 0008 C29Ah

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

x: Undefined

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

LVD1RIE Bit (Voltage Monitoring 1 Interrupt/Reset Enable)

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

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

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

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

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

LVD1RI Bit (Voltage Monitoring 1 Circuit Mode Select)

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

LVD1RN Bit (Voltage Monitoring 1 Reset Negate Select)

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

8.2.8 Voltage Monitoring 2 Circuit Control Register 0 (LVD2CR0)

Address(es): 0008 C29Bh

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

x: Undefined

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

LVD2RIE Bit (Voltage Monitoring 2 Interrupt/Reset Enable)

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

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

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

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

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

LVD2RI Bit (Voltage Monitoring 2 Circuit Mode Select)

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

LVD2RN Bit (Voltage Monitoring 2 Reset Negate Select)

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

8.3 VCC Input Voltage Monitor

8.3.1 Monitoring Vdet0

Monitoring Vdet0 is not possible.

8.3.2 Monitoring Vdet1

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

Table 8.2 Procedures for Setting up Monitoring against Vdet1

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

Note 1. Steps 4 to 6 can be performed during the waiting time of step 3. For details of td(E-A), refer to section 56, Electrical Characteristics.
Note 2. Steps 4 to 6 are not required if the digital filter is not in use.

8.3.3 Monitoring Vdet2

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

Table 8.3 Procedures for Setting up Monitoring against Vdet2

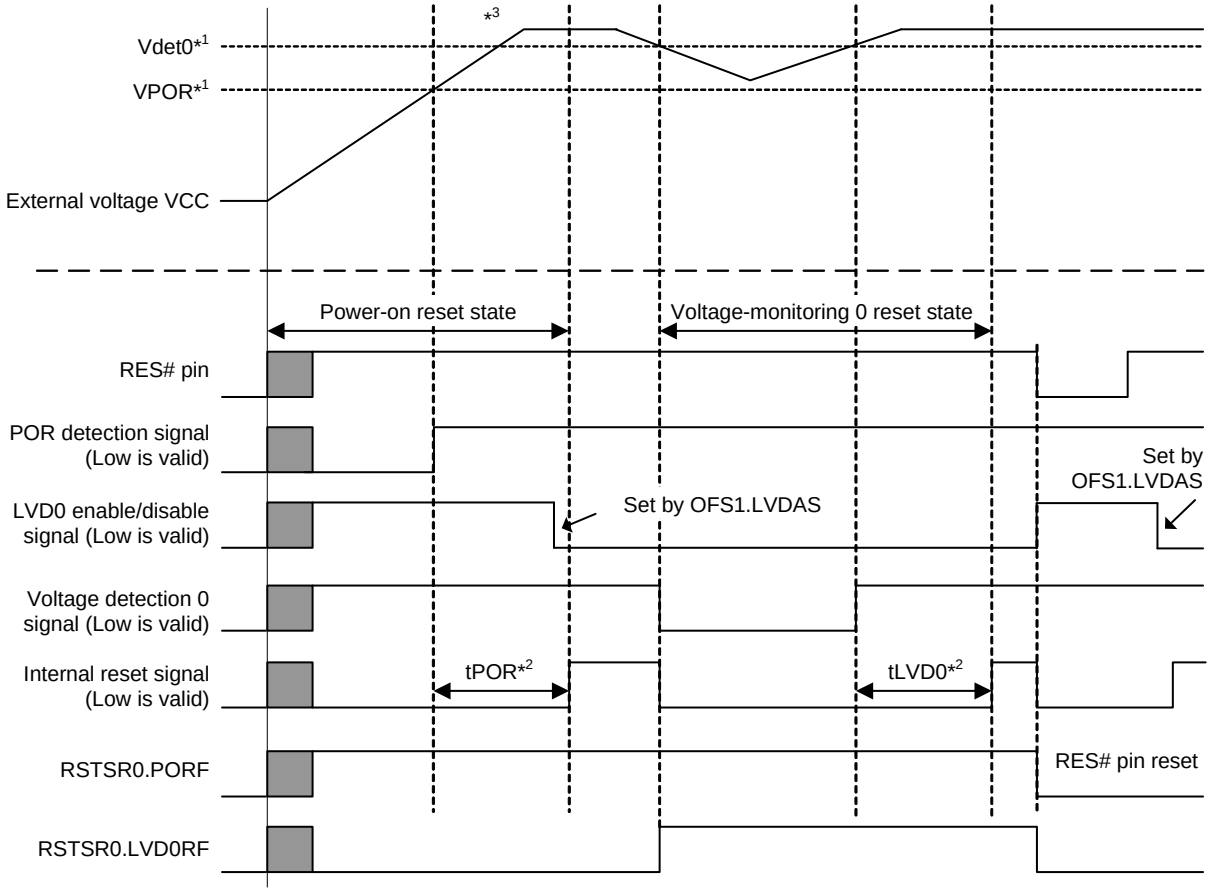
Step		Monitoring the Results of Comparison by Voltage Monitoring 2
Setting the voltage detection 2 circuit	1	Select the detection voltage by setting the LVDLVLR.LVD2LVL[3:0] bits.
	2	Set LVCMPCR.LVD2E = 1 (enabling the voltage detection 2 circuit).
	3	Wait for at least td(E-A) (LVD operation stabilization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

Note 1. Steps 4 to 6 can be performed during the waiting time of step 3. For details of td(E-A), refer to section 56, Electrical Characteristics.
Note 2. Steps 4 to 6 are not required if the digital filter is not in use.

8.4 Reset from Voltage Monitor 0

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

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



Note: For details of the electrical characteristics, refer to section 56, Electrical Characteristics.
Note 1. VPOR indicates the detection level for a power-on reset and Vdet0 indicates the detection level for a voltage monitor 0 reset.
Note 2. tPOR indicates the period of a power-on reset and tLVD0 indicates the period of a voltage monitor 0 reset.
Note 3. At the time the power-supply voltage rises, VCC must rise to at least the minimum guaranteed voltage before release from the POR reset state.

Figure 8.4 Example of Voltage Monitoring 0 Reset Operation

8.5 Interrupt and Reset from Voltage Monitor 1

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

(1) Setting in software standby mode

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

(2) Settings in deep software standby mode

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

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

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

Note 1. Steps 4 to 10 can be performed during the waiting time of step 3. For details of td(E-A), refer to section 56, Electrical Characteristics.
Note 2. Steps 4 to 6 are not required if the digital filter is not in use.
Note 3. Step 10 is not required if only the ELC event signal is to be output.
Note 4. The voltage of VCC = AVCC0 = AVCC1 when LVD1 is enabled must be set to at least 80 mV above the maximum value of the voltage detection 1 level selected by the LVDLVLR.LVD1LVL[3:0] bits.

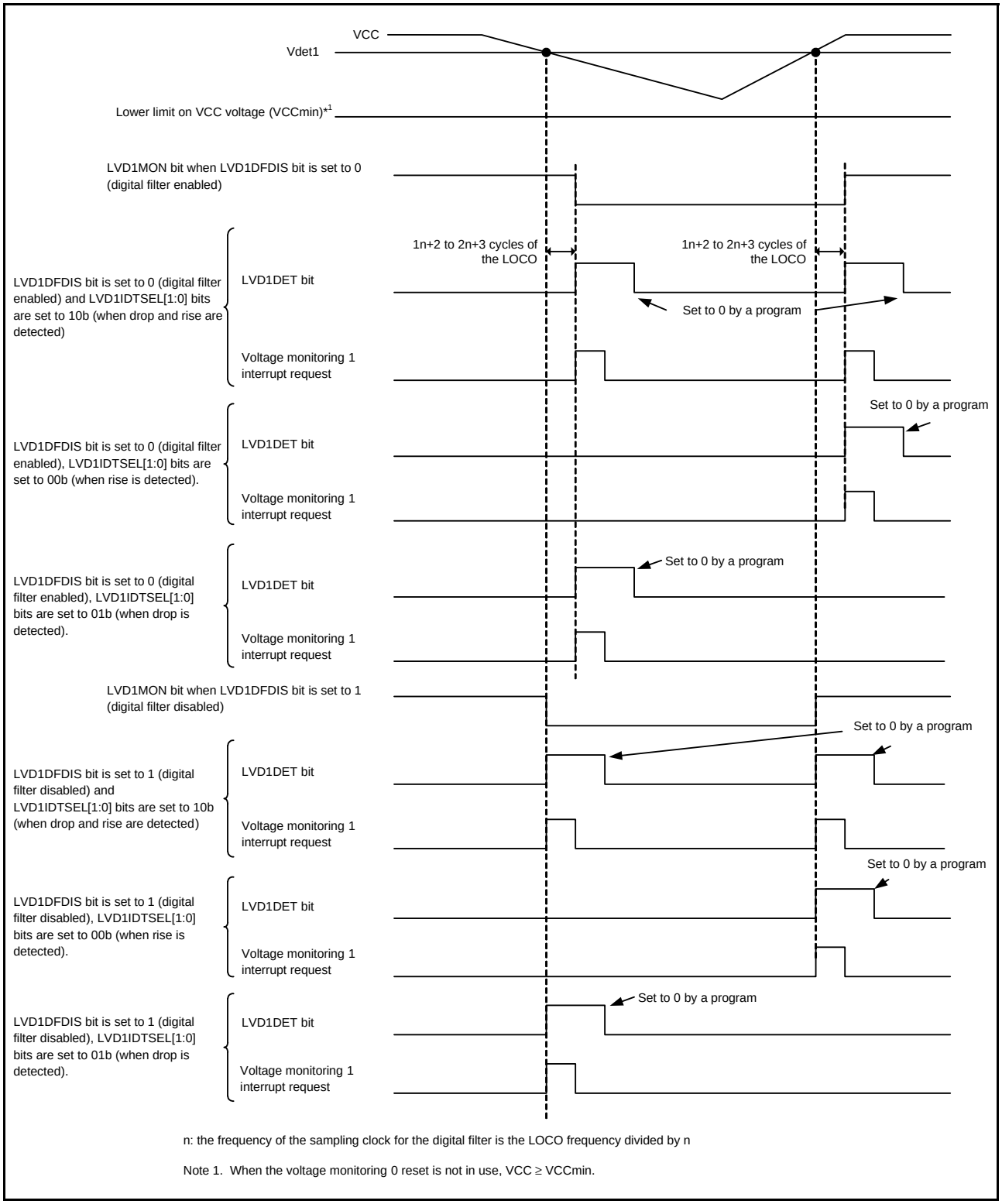
Table 8.5 Procedures for Setting Bits Related to the Voltage Monitoring 1 Interrupt and Voltage Monitoring 1 Reset so that Voltage Monitoring Stops

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

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

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

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



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

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

(1) Setting in software standby mode

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

(2) Settings in deep software standby mode

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

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

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

Note 1. Steps 4 to 10 can be performed during the waiting time of step 3. For details of td(E-A), refer to section 56, Electrical Characteristics.
Note 2. Steps 4 to 6 are not required if the digital filter is not in use.
Note 3. Step 10 is not required if only the ELC event signal is to be output.
Note 4. The voltage of VCC = AVCC0 = AVCC1 when LVD2 is enabled must be set to at least 80 mV above the maximum value of the voltage detection 2 level selected by the LVDLVLR.LVD2LVL[3:0] bits.

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

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

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

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

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

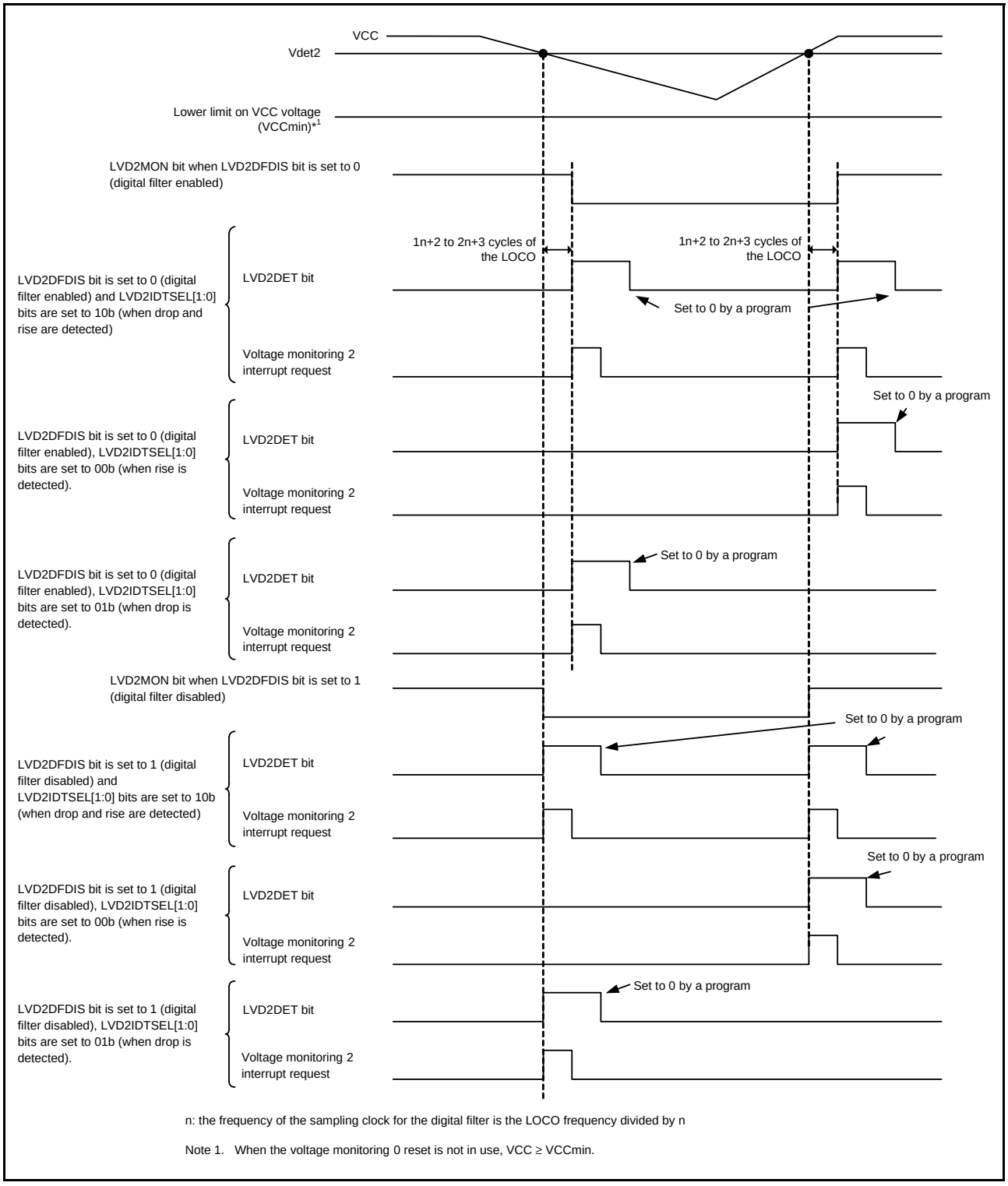


Figure 8.6 Example of Voltage Monitoring 2 Interrupt Operation

8.7 Event Link Output

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

(1) Vdet1 passage detection event

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

(2) Vdet2 passage detection event

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

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

8.7.1 Interrupt Handling and Event Linking

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

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

9. Clock Generation Circuit

9.1 Overview

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

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

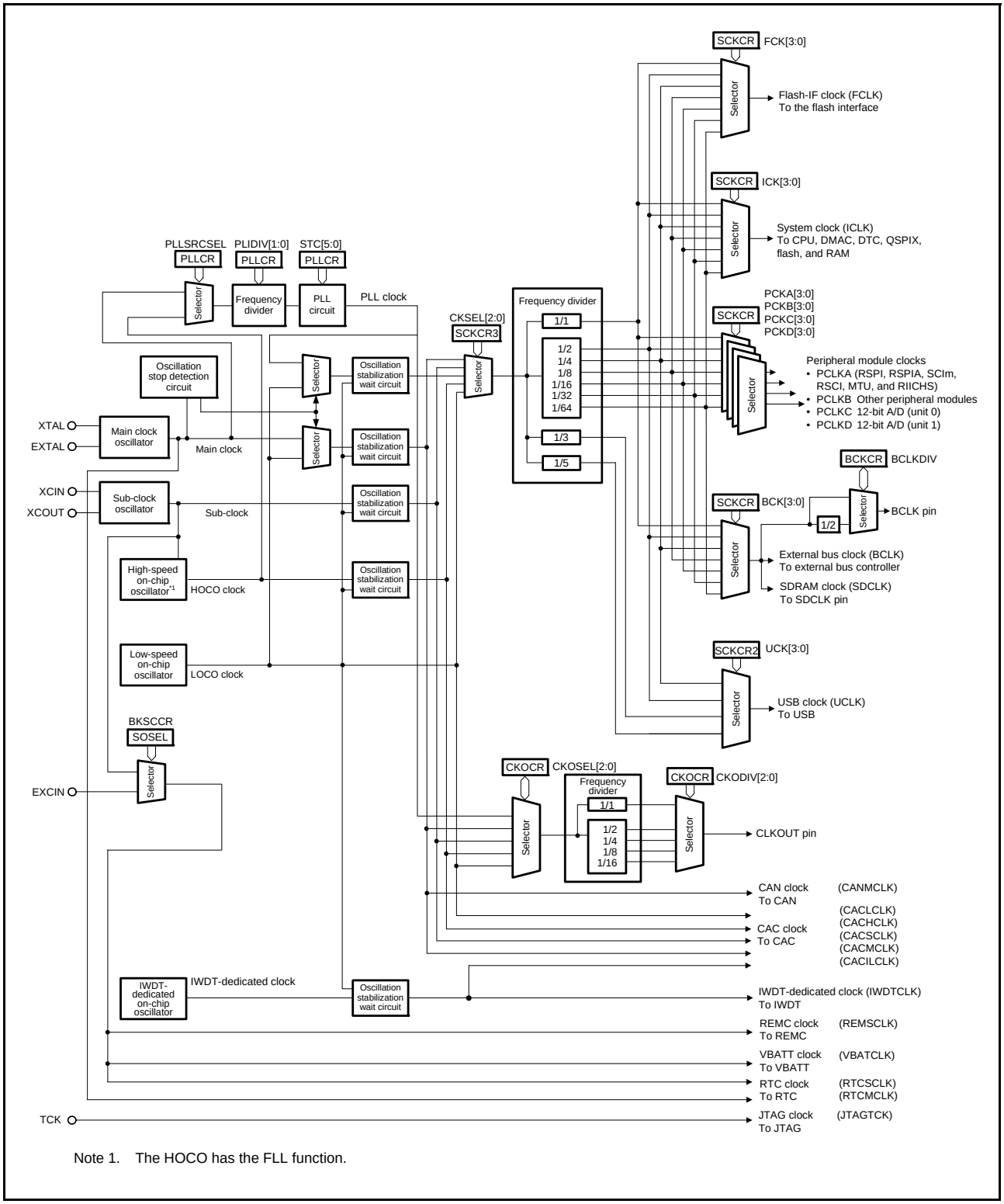
Item	Specification
Use	• Generates the system clock (ICLK) to be supplied to the CPU, DMAC, DTC, QSPIX, code flash memory, and RAM. • Generates the peripheral module clock (PCLKA) to be supplied to the RSPI, RSPIA, SCIm, RSCI, MTU, and RIICHS. • Generates the peripheral module clock (PCLKB) to be supplied to peripheral modules. • Generates the peripheral module clocks (for analog conversion) (ADCLK = PCLKC (unit 0), PCLKD (unit 1)) to be supplied to S12AD. • Generates the flash-IF clock (FCLK) to be supplied to the flash interface. • Generates the external bus clock (BCLK) to be supplied to the external bus. • Generates the SDRAM clock (SDCLK) to be supplied to the SDRAM. • Generates the USB clock (UCLK) to be supplied to the USB. • Generates the CAC clock (CACCLK) to be supplied to the CAC. • Generates the CAN clock (CANMCLK) to be supplied to the CAN. • Generates the RTC sub-clock (RTCSCLK) to be supplied to the RTC. • Generates the RTC main clock (RTCMCLK) to be supplied to the RTC. • Generates the REMC sub-clock (REMSCLK) to be supplied to the REMC. • Generates the VBATT clock (VBATCLK) to be supplied to the VBATT. • Generates the IWDT-dedicated clock (IWDTCLK) to be supplied to the IWDT. • Generates the JTAG clock (JTAGTCK) to be supplied to the JTAG.
Operating frequency*1	• ICLK: 120 MHz (max)*2 • PCLKA: 120 MHz (max) • PCLKB: 60 MHz (max) • PCLKC: 60 MHz (max) • PCLKD: 60 MHz (max) • FCLK: 4 MHz to 60 MHz (for programming and erasing the code flash memory and data flash memory) 60 MHz (max) (for reading from the data flash memory) • BCLK: 120 MHz (max) • BCLK pin output: 60 MHz (max) • SDCLK pin output: 60 MHz (max) • UCLK: 48 MHz • CLKOUT pin output: 40 MHz (max) • CACCLK: Same as the clock from respective oscillators. • CANMCLK: 24 MHz (max) • RTCSCLK: 32.768 kHz • RTCMCLK: 1 kHz to 16 MHz • REMSCLK: 32.768 kHz • VBATCLK: 32.768 kHz • IWDTCLK: 120 kHz • JTAGTCK: 10 MHz (max)
Main clock oscillator	• Resonator frequency: 8 MHz to 24 MHz • External clock input frequency: 24 MHz (max) • Connectable resonator or additional circuit: ceramic resonator, crystal resonator • Connection pin: EXTAL, XTAL • Oscillation stop detection function: When an oscillation stop is detected with the main clock, the system clock source is switched to LOCO, and MTU 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

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

Item	Specification
PLL frequency synthesizer	• Input clock source: Main clock, HOCO • Input pulse frequency division ratio: Selectable from 1, 2, and 3 • Input frequency: 8 MHz to 24 MHz • Frequency multiplication ratio: Selectable from 10 to 30 • Output clock frequency of the PLL frequency synthesizer: 120 MHz to 240 MHz
High-speed on-chip oscillator (HOCO)	• Selectable from 16 MHz, 18 MHz, and 20 MHz • HOCO power supply control • Frequency locked loop (FLL) • User trimming possible
Low-speed on-chip oscillator (LOCO)	Oscillation frequency: 240 kHz
IWDT-dedicated on-chip oscillator	Oscillation frequency: 120 kHz
JTAG external clock input (TCK)	Input clock frequency: 10 MHz (max)
Control of output on the BCLK pin	• BCLK clock output or high output is selectable • BCLK or BCLK/2 is selectable
Control of output on the SDCLK pin	SDCLK clock output or high output is selectable
Event linking (output)	Detection of stopping of the main clock oscillator
Event linking (input)	Switching of the clock source to the low-speed on-chip oscillator

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

Note 2. When the frequency of ICLK is set to faster than 60 MHz, the value of the ROMWT register needs to be modified.



Note 1. The HOCO has the FLL function.

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	
EXCIN	Input	Input pin for a 32.768-kHz external clock
TCK	Input	This pin is used to input the clock for the JTAG.
BCLK	Output	This pin is used to supply external devices with the external bus clock (BCLK).
SDCLK	Output	This pin is used to supply external devices with the SDRAM clock (SDCLK).
CLKOUT	Output	This is the CLKOUT output pin.

9.2 Register Descriptions

9.2.1 System Clock Control Register (SCKCR)

Address(es): 0008 0020h

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

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

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

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

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

SCKCR should not be modified in the following cases:

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

9.2.2 ROM Wait Cycle Setting Register (ROMWT)

Address(es): 0008 101Ch

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	ROMWT[1:0]	ROM Wait Cycle Setting	b1 b0 0 0: No wait 0 1: One wait cycle Settings other than above are prohibited.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register specifies the number of access wait cycles of the Flash memory.
When changing the frequency, modify the ROMWT register in the low speed state according to the procedure below.

- When changing from low speed to high speed:
  Immediately after modifying the value of the ROMWT register, read the register and execute the operation using read value of the register to change the frequency.
- When changing from high speed to low speed:
  Change the frequency. After the frequency is changed, modify the ROMWT register.

Table 9.3 shows the restrictions for setting the ROMWT register. Figure 9.2 and Figure 9.3 show the procedure for modifying the ROMWT[1:0] bits.

Table 9.3 Restrictions on Setting of the ROMWT[1:0] bits

ROMWT[1:0] Bits	ICLK ≤ 60 MHz	60 MHz < ICLK ≤ 120 MHz
00b	✓	×
01b	✓	✓

✓: Setting possible, ×: Setting prohibited

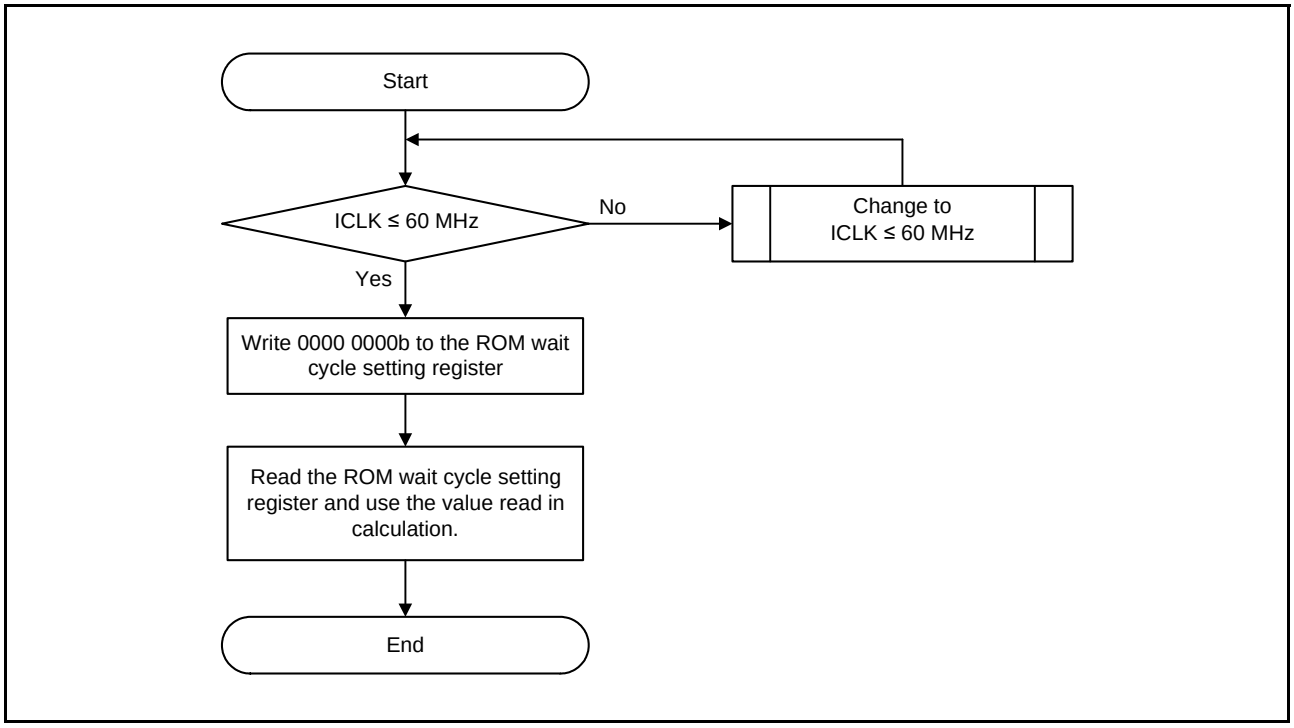


Figure 9.2 Procedure for Changing the ROMWT[1:0] Bits (When Changing the ROMWT[1:0] Bits from 01b to 00b)

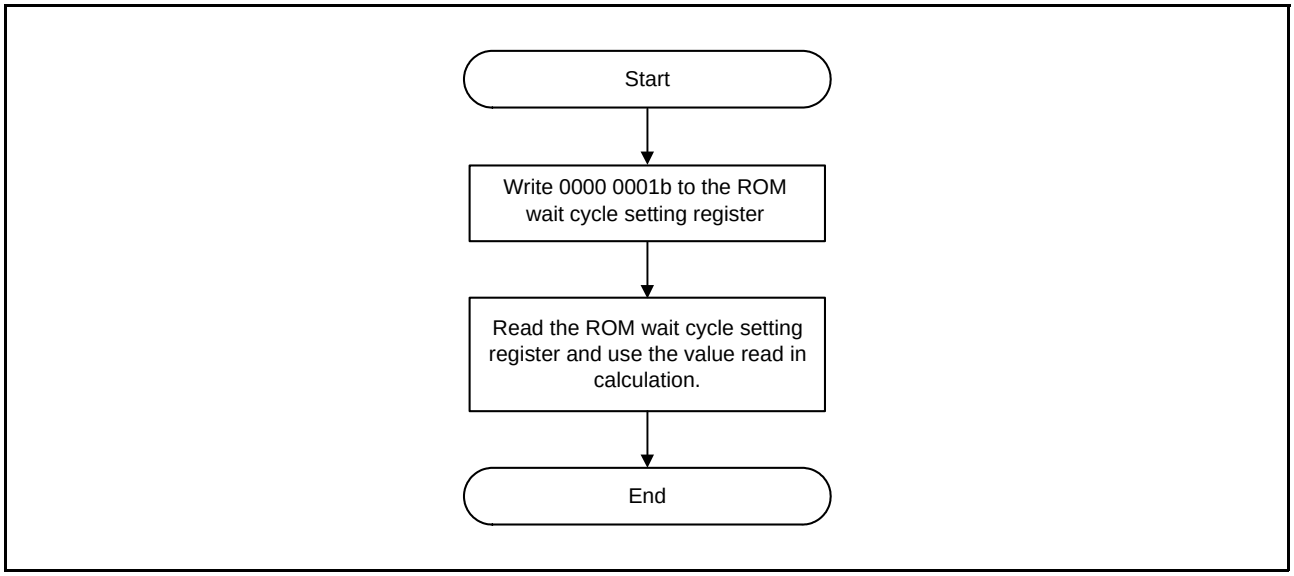


Figure 9.3 Procedure for Changing the ROMWT[1:0] Bits (When Changing the ROMWT[1:0] Bits from 00b to 01b)

9.2.3 System Clock Control Register 2 (SCKCR2)

Address(es): 0008 0024h

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

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

SCKCR2 should not be modified in the following cases:

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

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

These bits select the frequency of the USB clock (UCLK).
The duty ratio is 2:1 when × 1/3 is selected while it is 3:2 when × 1/5 is selected.

9.2.4 System Clock Control Register 3 (SCKCR3)

Address(es): 0008 0026h

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

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

SCKCR3 should not be modified in the following cases:

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

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

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

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

9.2.5 PLL Control Register (PLLCR)

Address(es): 0008 0028h

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

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

Note 1. When using USB, be sure to enable the FLL function.

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

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

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

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

PLLSRCSEL Bit (PLL Clock Source Select)

This bit selects the clock source for the PLL.

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

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

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

9.2.6 PLL Control Register 2 (PLLCR2)

Address(es): 0008 002Ah

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

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

PLLEN Bit (PLL Stop Control)

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

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

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

9.2.7 External Bus Clock Control Register (BCKCR)

Address(es): 0008 0030h

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

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

BCKCR should not be modified in the following cases:

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

BCLKDIV Bit (BCLK Pin Output Select)

This bit selects the clock signal for output from the BCLK pin.
Either the BCLK clock with the frequency selected by the BCK[3:0] bits in SCKCR or the BCLK clock divided by 2 can be selected. To control external bus control signals at the falling edge of the BCLK pin, set this bit to 1.

9.2.8 Main Clock Oscillator Control Register (MOSCCR)

Address(es): 0008 0032h

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

Bit	Symbol	Bit Name	Description	R/W
b0	MOSTP	Main Clock Oscillator Stop	0: Main clock oscillator is operating. 1: Main clock oscillator is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

MOSTP Bit (Main Clock Oscillator Stop)

This bit runs or stops the main clock oscillator.

The main clock oscillator is operated or stopped by the MOSTP bit and main clock oscillator forced oscillation bit in the main clock oscillator forced oscillation control register (MOFCR.MOFXIN). The main clock oscillator can be started by setting the MOSTP bit to operating or by setting the MOFXIN bit to forced oscillation. When the MOFXIN bit is set to forced oscillation, the oscillator operates even in software standby mode and deep software standby mode.

When the main clock is to be used as the source to drive counting by the realtime clock, use the MOFCR.MOFXIN bit to make the main clock run. If the main clock is only to be used as the system clock or PLL clock source, use the MOSCCR.MOSTP bit.

When the main clock is to be used as the clock source for counting and also for another purpose, use both the MOFCR.MOFXIN and MOSCCR.MOSTP bits to make the main clock run.

When changing the value of the MOSTP bit or MOFCR.MOFXIN bit, execute subsequent instructions after reading the bit and checking that its value has actually been updated (refer to (2) Notes on writing to I/O registers, in section 5, I/O Registers).

When the main clock is not to be used as the clock source for counting by the realtime clock, the main clock oscillator wait control register (MOSCWTCR) must be set. In this case, after the setting of the MOSCCR.MOSTP bit has been changed to make the main clock run, only start using the main clock after confirming that the OSCOVFSR.MOOVF flag has been set to 1.

When the main clock is to be used as the clock source for counting by the realtime clock, after the setting of MOFCR.MOFXIN bit has been changed to make the main clock run, only start using the main clock as the clock source for counting by the realtime clock after software has checked that the main clock oscillation stabilization time (crystal) (tMAINOSC) has elapsed. When an external clock signal is to be input for supply to the main clock oscillator, waiting for the main clock oscillation stabilization time (tMAINOSC) is not necessary.

For the main clock oscillator, a fixed time is required for oscillation to become stable after the settings for operation have been made. Furthermore, a fixed time is required for oscillation to actually stop after the settings to stop oscillation have been made. Accordingly, take note of the following limitations when starting and stopping operation.

- Main-clock operation can be selected with the MOSTP bit regardless of the setting of the OSCOVFSR.MOOVF flag. However, the time until deactivation of the main clock is completed (the time until the OSCOVFSR.MOOVF flag is set to 0 after the setting to stop operation) means that writing to the MOSCCR register takes longer than the setting to operate the main clock.
- The main clock can be stopped by the MOSTP bit regardless of the setting of the OSCOVFSR.MOOVF flag. However, the wait for oscillation to become stable, i.e. the oscillation settling time (the time until the OSCOVFSR.MOOVF flag is set to 1 after the setting to operate the main clock), means that writing to the MOSCCR register takes longer than the setting to stop operation.

- Regardless of whether or not it is selected as the system clock, confirm that the OSCOVFSR.MOOVF flag has been set to 1 before executing a WAIT instruction to place the chip to software standby or deep software standby after the setting to operate the main clock oscillator by the MOSTP bit.
- When a transition to software standby or deep software standby is to follow the setting to stop the main clock oscillator, confirm that the OSCOVFSR.MOOVF flag has been set to 0 before executing the WAIT instruction.

Writing of 1 to the MOSTP bit (stopping the main clock oscillator) is prohibited in either of the following cases:

- The main clock oscillator is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).
- 0 (PLL operation) is selected by the PLL stop control bit in PLL control register 2 (PLLCR2.PLLEN), and the main clock oscillator is selected by the PLL clock source select bit in the PLL control register (PLLCR.PLLSRCSEL).

9.2.9 Sub-Clock Oscillator Control Register (SOSCCR)

Address(es): 0008 0033h

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

Bit	Symbol	Bit Name	Description	R/W
b0	SOSTP	Sub-Clock Oscillator Stop	0: Sub-clock oscillator is operating. 1: Sub-clock oscillator is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SOSTP Bit (Sub-Clock Oscillator Stop)

This bit runs or stops the sub-clock oscillator.

The SOSTP bit and the sub-clock control bit in the RTC control register 3 (RCR3.RTCEN) controls whether to operate or stop the sub-clock oscillator. If one of these bits is set so as to enable the operation, the sub-clock oscillator runs.

When changing the value of the SOSTP bit or RCR3.RTCEN bit, execute subsequent instructions after reading the bit and checking that its value has actually been updated (refer to (2) Notes on writing to I/O registers, in section 5, I/O Registers).

When the sub-clock is to be used as the source to drive counting by the realtime clock, use the RCR3.RTCEN and SOSTP bits. When the sub-clock is not to be used as the clock source for counting by the realtime clock but is to be used as the system clock, use the SOSCCR.SOSTP bit.

When the SOSTP bit is set to operate the sub-clock, be sure to set the sub-clock oscillator wait control register (SOSCWTCR) beforehand. Furthermore, after selecting sub-clock operation, only start using the sub-clock after confirming that the OSCOVFSR.SOOVF flag has been set to 1.

A fixed time for stabilization is required for oscillation to become stable after selecting sub-clock operation with the SOSTP bit. Furthermore, a fixed time is required for oscillation to actually stop after the setting to stop operation. Accordingly, take note of the following when starting and stopping operation with the SOSTP bit.

- Sub-clock operation can be selected with the SOSTP bit regardless of the setting of the OSCOVFSR.SOOVF flag. However, the time until deactivation of the sub-clock is completed (the time until the OSCOVFSR.SOOVF flag is set to 0 after the setting to stop operation) means that writing to the SOSCCR register takes longer than the setting to operate the sub-clock.
- The sub-clock can be stopped by the SOSTP bit regardless of the setting of the OSCOVFSR.SOOVF flag. However, the wait for oscillation to become stable, i.e. the oscillation settling time (the time until the OSCOVFSR.SOOVF flag is set to 1 after the setting to operate the sub-clock), means that writing to the SOSCCR register takes longer than the setting to stop operation.
- Regardless of whether or not it is selected as the system clock, confirm that the OSCOVFSR.SOOVF flag has been set to 1 before executing a WAIT instruction to place the chip on software standby or deep software standby after the setting to operate the sub-clock oscillator by the SOSTP bit.
- When a transition to software standby or deep software standby is to follow the setting to stop the sub-clock oscillator, confirm that the OSCOVFSR.SOOVF flag has been set to 0 after the setting to stop the sub-clock oscillator and before executing the WAIT instruction.
- When the SOSCCCR2.SOSTP2 bit is to be set to 1, also set the SOSTP bit to 1.

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]).

9.2.10 Low-Speed On-Chip Oscillator Control Register (LOCOCR)

Address(es): 0008 0034h

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

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

LCSTP Bit (LOCO Stop)

This bit runs or stops the LOCO.
After the setting of the LCSTP bit has been changed so that the LOCO operates, only start using the LOCO after the LOCO clock oscillation stabilization waiting time (tLOCOWT) has elapsed.
That is, a fixed time for stabilization of oscillation is required for oscillation to become stable after setting LOCO operation with the LCSTP bit. A fixed time is also required for oscillation to stop after the setting to stop the oscillator. Accordingly, take note of the following limitations when starting and stopping the oscillator.

- When restarting the LOCO after it has been stopped, allow at least five cycles of the LOCO as an interval over which it is still stopped.
- Ensure that oscillation by the LOCO is stable when making the setting to stop the LOCO.
- Regardless of whether or not it is selected as the system clock, ensure that oscillation by the LOCO is stable before executing a WAIT instruction to place the chip on software standby or deep software standby.
- When a transition to software standby or deep software standby is to follow the setting to stop the LOCO, wait for at least three cycles of the LOCO after the setting to stop the LOCO and before executing the WAIT instruction.

Writing of 1 to the LCSTP bit (stopping the LOCO) is prohibited while the LOCO is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).
Writing of 1 to the LCSTP bit (stopping the LOCO) is prohibited if detection of oscillation stopping is enabled by the oscillation stop detection enable bit in the oscillation stop detection control register (OSTDCR.OSTDE).
Since the LOCO clock is used to measure the waiting time for other oscillators, the LOCO clock oscillates while the waiting time for other oscillators is being measured, regardless of the setting of LCSTP bit. Therefore, the LOCO clock may be unintentionally supplied even if the LCSTP bit is set to be stopped.

9.2.11 IWDT-Dedicated On-Chip Oscillator Control Register (ILOCOCR)

Address(es): 0008 0035h

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

Bit	Symbol	Bit Name	Description	R/W
b0	ILCSTP	IWDT-Dedicated On-Chip Oscillator Stop	0: IWDT-dedicated on-chip oscillator is operating. 1: IWDT-dedicated on-chip oscillator is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

When the IWDT start mode select bit in the option function select register 0 (OFS0.IWDTSTRT) is 0 (IWDT operating), the setting of this register is invalid; it is valid only when the OFS0.IWDTSTRT bit is set to 1 (IWDT stopped). The ILCSTP bit cannot be changed from 0 (IWDT-dedicated on-chip oscillator operating) to 1 (IWDT-dedicated on-chip oscillator stopped) while ILOCOCR is valid.

ILCSTP Bit (IWDT-Dedicated On-Chip Oscillator Stop)

This bit runs or stops the IWDT-dedicated on-chip oscillator.

After the setting of the ILCSTP bit has been changed to make the IWDT-dedicated on-chip oscillator run, confirm that the OSCOVFSR.ILCOVF flag has been set to 1 before starting to use the oscillator.

When a transition to software standby or deep software standby mode is to follow the setting to start the IWDT-dedicated on-chip oscillator, confirm that the OSCOVFSR.ILCOVF flag has been set to 1 before executing the WAIT instruction.

9.2.12 High-Speed On-Chip Oscillator Control Register (HOCOCR)

Address(es): 0008 0036h

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

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

Note 1. The HCSTP bit value after a reset is 0 when the OFS1.HOCOEN bit is 0. It is 1 when the OFS1.HOCOEN bit is 1.

HCSTP Bit (HOCO Stop)

This bit runs or stops the HOCO.
After the setting of the HCSTP bit has been changed to make the HOCO run, confirm that the OSCOVFSR.HCOVF flag has been set to 1 before starting to use the oscillator.
A fixed time for stabilization is required for oscillation to become stable after setting HOCO operation with the HCSTP bit. Furthermore, a fixed time is required for oscillation to actually stop after the setting to stop operation. Accordingly, take note of the following when starting and stopping operation with the HCSTP bit.

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

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

9.2.13 High-Speed On-Chip Oscillator Control Register 2 (HOCOCR2)

Address(es): 0008 0037h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	HCFRQ[1:0]	HOCO Frequency Setting	b1 b0 0 0: 16 MHz 0 1: 18 MHz 1 0: 20 MHz Settings other than above are prohibited.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Writing to the HOCOCR2 register is prohibited when the HOCOCR.HCSTP bit is 0 (making the HOCO run).

9.2.14 FLL Control Register 1 (FLLCR1)

Address(es): 0008 0039h

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

Bit	Symbol	Bit Name	Description	R/W
b0	FLLEN	FLL Function Enable	0: FLL is disabled. 1: FLL is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Be sure to stop the HOCO (HOCOCR.HCSTP = 1) before changing the setting of the FLLCR1.FLLEN bit.
Note: Ensure the operation of the sub-clock oscillator is stable while FLL is enabled (FLLCR1.FLLEN = 1).

The FLLCR1 register enables or disables the frequency correction function of the HOCO.

FLLEN Bit (FLL Function Enable)

This bit enables or disables the FLL function of the HOCO. Enabling FLL improves the precision of the HOCO frequency. Operation of the sub-clock oscillator must be stable when FLL is to be enabled because FLL operates with the clock generated by the sub-clock oscillator. The frequency precision cannot be guaranteed unless the FLL operation is stable even if the setting of the OSCOVFSR.HCOVF bit is 1. FLL must be disabled before the MCU is placed in the software standby mode. Be sure to set this bit to 0 before placing the MCU in the software standby mode.
Figure 9.4 shows the flow of setting FLL after release from the reset state or deep software standby mode. Figure 9.5 shows the flow of setting FLL before transition to and after release from the software standby mode.

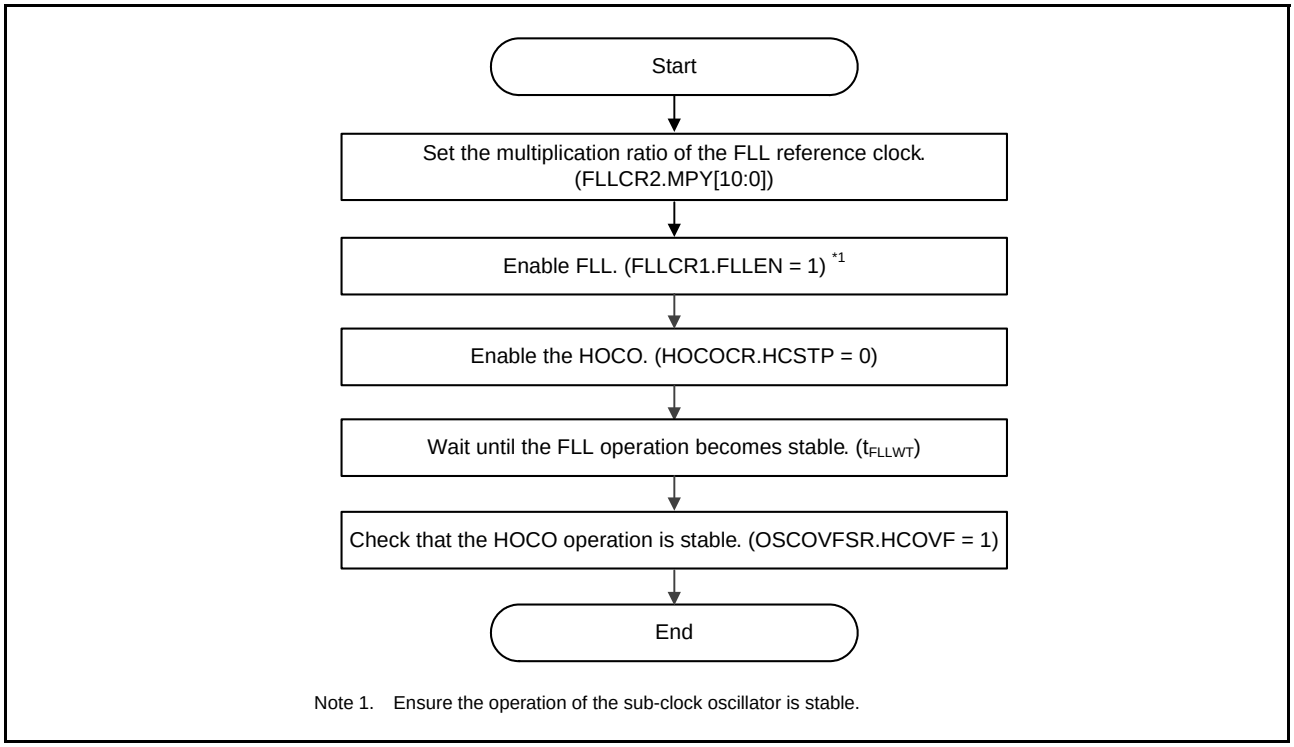
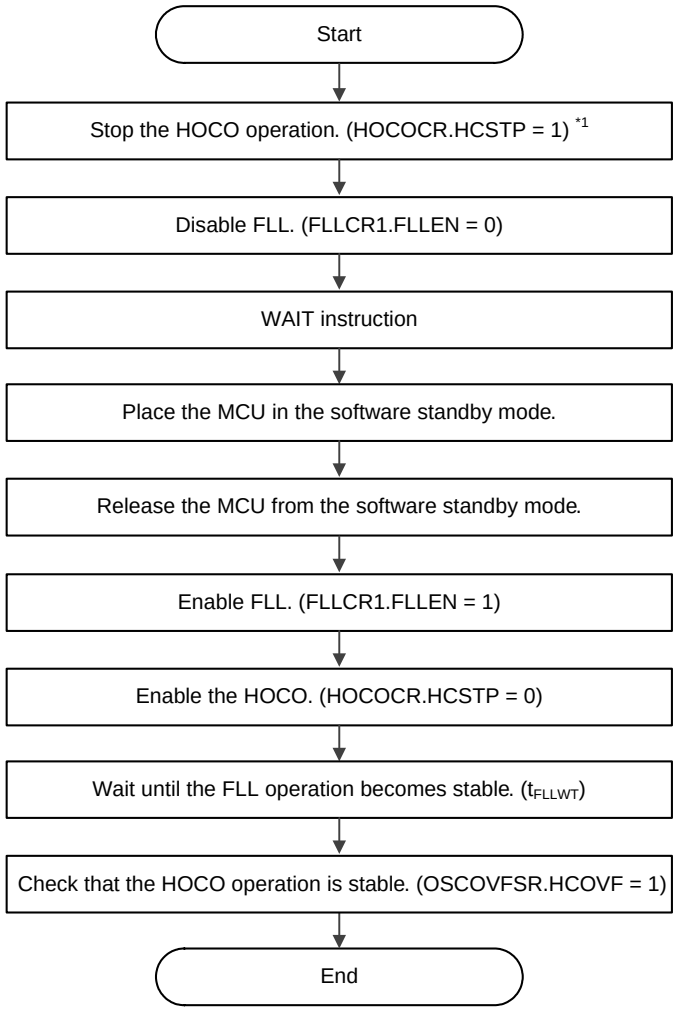


Figure 9.4 Flow of Setting FLL after Release from the Reset State or Deep Software Standby Mode



Note 1. When the HOCO is in use for the system clock or PLL reference clock, be sure to change the clock source to another clock before stopping the HOCO operation.

Figure 9.5 Flow of Setting FLL before Transition to and after Release from the Software Standby Mode

9.2.15 FLL Control Register 2 (FLLCR2)

Address(es): 0008 003Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	MPY[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	MPY[10:0]	Frequency Multiplication Factor Select	• These bits must be set to 1E9h when the setting of the HOCOCR2.HCFRQ[1:0] bits is 00b (16 MHz). • These bits must be set to 226h when the setting of the HOCOCR2.HCFRQ[1:0] bits is 01b (18 MHz). • These bits must be set to 263h when the setting of the HOCOCR2.HCFRQ[1:0] bits is 10b (20 MHz). 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

The FLLCR2 register controls the FLL function of the HOCO.

MPY[10:0] Bit (Frequency Multiplication Factor Select)

These bits select the multiplication ratio of the FLL reference clock. These bits must be set before FLL is enabled (FLLCR1.FLLEN = 1).

9.2.16 Oscillation Stabilization Flag Register (OSCOVFSR)

Address(es): 0008 003Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	ILCOVF	HCOVF	PLOVF	SOOVF	MOOVF
Value after reset:	0	0	0	0/1*1	0/1*2	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MOOVF	Main Clock Oscillation Stabilization Flag	0: MOSTP = 1 (stopping the main clock oscillator) or oscillation of the main clock has not yet become stable.*3, *5 1: Oscillation of the main clock is stable so the clock is available for use as the system clock.*4	R
b1	SOOVF	Sub-Clock Oscillation Stabilization Flag	0: SOSTP = 1 (stopping the sub-clock oscillator) or oscillation of the sub-clock has not yet become stable.*5 1: Oscillation of the sub-clock is stable so the clock is available for use as the system clock.*4	R
b2	PLOVF	PLL Clock Oscillation Stabilization Flag	0: The PLL clock is stopped or oscillation of the PLL clock has not yet become stable. 1: Oscillation of the PLL clock is stable so the clock is available for use as the system clock.	R
b3	HCOVF*2	HOCO Clock Oscillation Stabilization Flag	0: The HOCO clock is stopped or oscillation of the HOCO clock has not yet become stable. 1: Oscillation of the HOCO clock is stable so the clock is available for use as the system clock.	R
b4	ILCOVF*1	IWDT-Dedicated Clock Oscillation Stabilization Flag	0: The IWDT-dedicated on-chip oscillator is stopped or oscillation of the IWDT-dedicated on-chip oscillator has not yet become stable. 1: Oscillation of the IWDT-dedicated on-chip oscillator is stable.	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The ILCOVF flag value after a reset is 1 when the OFS0.IWDTSTRT bit is 0. It is 0 when the OFS0.IWDTSTRT bit is 1.
Note 2. The HCOVF flag value after a reset is 1 when the OFS1.HOCOEN bit is 0. It is 0 when the OFS1.HOCOEN bit is 1.
Note 3. The MOOVF flag does not reflect the control of the main clock oscillator by the MOFCR.MOFXIN bit. Accordingly, if the MOSTP bit is set to 1 while the MOFXIN bit is 1, the main clock oscillator continues to oscillate but the setting of the MOOVF flag becomes 0.
Note 4. The SOOVF flag does not reflect the control of the sub-clock oscillator by the RCR3.RTCEN bit. Accordingly, if the SOSTP bit is set to 1 while the RTCEN bit is 1, the sub-clock oscillator continues to oscillate but the setting of the SOOVF flag becomes 0.
Note 5. If the value set in the wait control register of the main clock oscillator and sub-clock oscillator is not sufficient for the given oscillation stabilization time, the oscillation stabilization flag is set to 1 and supply of the clock signal to the internal circuits starts before oscillation is stable. This may cause malfunction of this MCU, so ensure that the setting of the wait control register is at least the oscillation settling time for the oscillator considering the maximum frequency of the LOCO clock.

OSCOVFSR contains flags to indicate the states of operation of the counters within the oscillation stabilization wait circuits for the individual oscillators.
The counters measure the waiting times until each oscillator output clock is supplied to the internal circuits after oscillation starts, and an overflow of a counter indicates the start of clock supply from the corresponding oscillator to the internal circuits.

MOOVF Flag (Main Clock Oscillation Stabilization Flag)

This flag indicates the state of operation of the counter that measures the waiting time for the main clock oscillator.
[Setting condition]

- After the main clock oscillator has stopped and the MOSCCR.MOSTP bit is set to 0, the number of LOCO clock cycles corresponding to the setting of the MOSCWTCR register being counted and supply of the main clock within the MCU starting.

[Clearing condition]

- After the main clock oscillator has started to operate and the MOSCCR.MOSTP bit has been set to 1, deactivation of the main clock oscillator being completed.

SOOVF Flag (Sub-Clock Oscillation Stabilization Flag)

This flag indicates the state of operation of the counter that measures the waiting time for the sub-clock oscillator.
[Setting condition]

- After the sub-clock oscillator has stopped and the SOSCCR.SOSTP bit is set to 0, the number of LOCO clock cycles corresponding to the setting of the SOSCWTCR register being counted and supply of the sub-clock within the MCU starting.

[Clearing condition]

- After the sub-clock oscillator has started to operate and the SOSCCR.SOSTP bit has been set to 1, deactivation of the sub-clock oscillator being completed.

PLOVF Flag (PLL Clock Oscillation Stabilization Flag)

This flag indicates the state of operation of the counter that measures the waiting time for the PLL.
[Setting condition]

- After the PLL has stopped and the PLLCR2.PLLEN bit is set to 0, 62 cycles of the LOCO clock being counted and supply of the PLL clock within the MCU starting.

If oscillation by the PLL clock source selected by the PLLCR.PLLSRCSEL bit is not stable when the PLLEN bit is set to 0, counting of LOCO clock cycles proceeds after the oscillation of the PLL clock source has been stabilized.

[Clearing condition]

- After the PLL has started to operate and the PLLCR2.PLLEN bit has been set to 1, deactivation of the PLL being completed.

HCOVF Flag (HOCO Clock Oscillation Stabilization Flag)

This flag indicates the state of operation of the counter that measures the waiting time for the high-speed on-chip oscillator.
[Setting condition]

- After the high-speed on-chip oscillator has stopped and the HOCOCR.HCSTP bit is set to 0, 25 cycles of the LOCO clock being counted and supply of the HOCO clock within the MCU starting.

[Clearing condition]

- After the high-speed on-chip oscillator has started to operate and the HOCOCR.HCSTP bit has been set to 1, deactivation of the high-speed on-chip oscillator being completed.

ILCOVF Flag (IWDT-Dedicated Clock Oscillation Stabilization Flag)

This flag indicates the state of operation of the counter that measures the waiting time for the IWDT-dedicated on-chip oscillator.
[Setting condition]

- After the IWDT-dedicated on-chip oscillator has stopped and the ILOCOCR.ILCSTP bit is set to 0, 34 cycles of the LOCO clock being counted and supply of the IWDT-dedicated clock within the MCU starting.

9.2.17 Oscillation Stop Detection Control Register (OSTDCR)

Address(es): 0008 0040h

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

Bit	Symbol	Bit Name	Description	R/W
b0	OSTDIE	Oscillation Stop Detection Interrupt Enable	0: The oscillation stop detection interrupt is disabled. Oscillation stop detection is not notified to the POE. 1: The oscillation stop detection interrupt is enabled. Oscillation stop detection is notified to the POE.	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	OSTDE	Oscillation Stop Detection Function Enable	0: Oscillation stop detection function is disabled. 1: Oscillation stop detection function is enabled.	R/W

OSTDCR register is used to enable the oscillation stop detection function for the main clock oscillator and conveying of interrupts in response.

OSTDIE Bit (Oscillation Stop Detection Interrupt Enable)

If the oscillation-stop detection flag in the oscillation-stop detection status register (OSTDSR.OSTDF) requires clearing, do this after clearing the OSTDIE bit to 0. Wait for at least two cycles of PCLKB before again setting the OSTDIE bit to 1. According to the number of cycles for access to read out a given I/O register, waiting time longer than two cycles of PCLKB may have to be secured.

OSTDE Bit (Oscillation Stop Detection Function Enable)

This bit enables or disables the oscillation stop detection function.

When the OSTDE bit is 1 (oscillation stop detection function enabled), the LOCO stop bit (LOCOCR.LCSTP) is set to 0 and the LOCO operation is started. The LOCO cannot be stopped while the oscillation stop detection function is enabled; writing 1 to the LOCOCR.LCSTP bit (LOCO stopped) is invalid.

When the oscillation stop detection flag in the oscillation stop detection status register (OSTDSR.OSTDF) is 1 (main clock oscillation stop detected), writing 0 to the OSTDE bit is invalid.

When the OSTDE bit is 1, a transition cannot be made to software standby mode or deep software standby mode. To make a transition to software standby mode or deep software standby mode, execute the WAIT instruction with the OSTDE bit being 0.

To check the oscillation stop detection flag in the oscillation stop detection status register (OSTDSR.OSTDF), wait for at least three cycles of ICLK after the OSTDE bit has been set to 1 (oscillation stop detection enabled).

9.2.18 Oscillation Stop Detection Status Register (OSTDSR)

Address(es): 0008 0041h

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

Bit	Symbol	Bit Name	Description	R/W
b0	OSTDF	Oscillation Stop Detection Flag	0: Stopping of the main clock oscillator has not been detected. 1: Stopping of the main clock oscillator has been detected.	R(/W) *1
b7 to b1	—	Reserved	These bits are read as 0 and cannot be modified.	R

Note 1. This bit can only be set to 0.

The OSTDSR register indicates the detection of stopping of the main clock oscillator.

OSTDF Flag (Oscillation Stop Detection Flag)

This bit is a flag to indicate the main clock status. When the OSTDF flag is 1, it indicates that the main clock oscillation stop has been detected.

Once the main clock oscillation stop is detected, the OSTDF flag is not set to 0 even though the main clock oscillation is restarted. The OSTDF flag is set to 0 by reading 1 from the bit and then writing 0. At least 3 ICLK cycles of wait time is necessary between writing 0 to OSTDF and reading OSTDF as 0. If the OSTDF flag is set to 0 from 1 while the main clock oscillation is stopped, the OSTDF flag becomes 0 and then returns to 1.

The OSTDF flag cannot be modified to 0 while the main clock oscillator or PLL is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]) (010b or 100b). The OSTDF flag should be set to 0 after switching the clock source to other sources than the main clock oscillator and PLL.

[Setting condition]

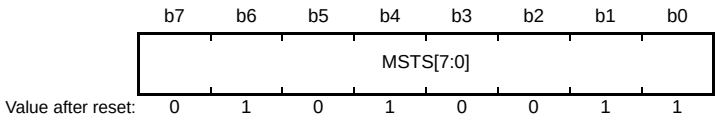
- The main clock oscillation is stopped with the OSTDCR.OSTDE being 1 (oscillation stop detection function enabled).

[Clearing condition]

- When 1 is read and then 0 is written when the value of the SCKCR3.CKSEL[2:0] bits are neither 010b or 100b and the PLLCR.PLLSRCSEL bit is set to a value other than 0.

9.2.19 Main Clock Oscillator Wait Control Register (MOSCWTCR)

Address(es): 0008 00A2h



MOSCWTCR is used to control the waiting time until output of the signal from the main clock oscillator to the internal circuits starts. The oscillation stabilization wait circuit for the main clock oscillator measures the waiting time by counting the number of LOCO clock cycles corresponding to the setting of the MOSCWTCR register.
The oscillation stabilization wait circuit measures the waiting time and controls the clock supply within the MCU. When the main clock oscillator starts by setting the MOSCCR.MOSTP bit, the oscillation stabilization wait circuit starts counting the waiting time with the LOCO clock. The clock supply within the MCU is disabled over the period until counting of the set number of cycles is completed. After counting is completed, supply of the clock signal within the MCU starts and the OSCOVFSR.MOOVF flag is set to 1.
Counting of LOCO clock cycles by the oscillation stabilization wait circuit proceeds regardless of the setting of the LOCOCR.LCSTP bit. Hardware automatically controls running and stopping of the LOCO clock for measurement of the waiting time.
Values can only be written to MOSCWTCR while the MOSCCR.MOSTP bit is 1 or the OSCOVFSR.MOOVF flag is 1; do not attempt writing to MOSCWTCR if neither is the case.
The waiting time is not required when an external clock signal is input for the main clock oscillator. Set the MSTS[7:0] bits to 00h.

The value of the MSTS[7:0] bits required for correspondence with the waiting time required to secure stable oscillation by the main clock oscillator is obtained by using the maximum frequency for fLOCO in the formula below.

MSTS[7:0] > (tMAINOSC × (fLOCO_max) + 16)/32
(tMAINOSC: main clock oscillation stabilization time; fLOCO_max: maximum frequency for fLOCO)

If tMAINOSC is 1 ms and fLOCO_max is 264 kHz (the period is 1/3.78 µs), the formula gives MSTS[7:0] > (1 ms × (264 kHz) + 16)/32 = 8.75, so set the MSTS[7:0] bits to 9.

Waiting time:
When LOCO is at its highest frequency: (9 × 32 – 16) × (1/264 kHz = 3.78 µs) = 1.028 ms
When LOCO is at its normal frequency: (9 × 32 + 3) × (1/240 kHz = 4.18 µs) = 1.216 ms
When LOCO is at its lowest frequency: (9 × 32 + 10) × (1/216 kHz = 4.63 µs) = 1.380 ms

9.2.20 Sub-Clock Oscillator Wait Control Register (SOSCWTCR)

Address(es): 0008 00A3h

	b7	b6	b5	b4	b3	b2	b1	b0
	SSTS[7:0]							
Value after reset:	0	0	1	0	0	0	0	1

SOSCWTCR is used to control the waiting time until output of the signal from the sub-clock oscillator to the internal circuits starts. The oscillation stabilization wait circuit for the sub-clock oscillator measures the waiting time by counting the number of LOCO clock cycles corresponding to the setting of the SOSCWTCR register.
The oscillation stabilization wait circuit measures the waiting time and controls the clock supply within the MCU. When the sub-clock oscillator starts by setting the SOSCCR.SOSTP bit, the oscillation stabilization wait circuit starts counting the waiting time with the LOCO clock. The clock supply within the MCU is disabled over the period until counting of the set number of cycles is completed. After counting is completed, supply of the clock signal within the MCU starts and the OSCOVFSR.SOOVF flag is set to 1.
Counting of LOCO clock cycles by the oscillation stabilization wait circuit proceeds regardless of the setting of the LOCOCR.LCSTP bit. Hardware automatically controls running and stopping of the LOCO clock for measurement of the waiting time.
Values can only be written to SOSCWTCR while the SOSCCR.SOSTP bit is 1 or the OSCOVFSR.SOOVF flag is 1; do not attempt writing to SOSCWTCR if neither is the case.

The value of the SSTS[7:0] bits required for correspondence with the expected time to secure settling of oscillation by the sub-clock oscillator is obtained by using the maximum frequency for fLOCO in the formula below.

SSTS[7:0] > (tSUBOSC × (fLOCO_max) + 16)/16384
(tSUBOSC: sub-clock oscillation stabilization time; fLOCO_max: maximum frequency for fLOCO)

If tSUBOSC is 2 s and fLOCO is 264 kHz (the period is 1/3.78 µs), the formula gives SSTS[7:0] > (2 s × (264 kHz) + 16)/16384 = 32.22, so set the SSTS[7:0] bits to 33.

Waiting time:
When LOCO is at its highest frequency: (33 × 16384 – 16) × (1/264 kHz = 3.78 µs) = 2.044 s
When LOCO is at its normal frequency: (33 × 16384 + 3) × (1/240 kHz = 4.18 µs) = 2.260 s
When LOCO is at its lowest frequency: (33 × 16384 + 10) × (1/216 kHz = 4.63 µs) = 2.503 s

9.2.21 Main Clock Oscillator Forced Oscillation Control Register (MOFCR)

Address(es): 0008 C293h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	MOSEL	MODRV2[1:0]		—	—	—	MOFXIN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MOFXIN	Main Clock Oscillator Forced Oscillation	0: Oscillator is not controlled by this bit. 1: The main clock oscillator is forcedly oscillated.	R/W
b3 to b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	MODRV2[1:0]	Main Clock Oscillator Driving Ability 2 Switching	b5 b4 0 0: 20.1 to 24 MHz 0 1: 16.1 to 20 MHz 1 0: 8.1 to 16 MHz 1 1: 8 MHz	R/W
b6	MOSEL	Main Clock Oscillator Switching	0: Resonator 1: External clock input	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

As well as selecting and de-selecting forcible starting of the main clock oscillator, the setting of the MOFCR also selects the driving ability, and the oscillator or an external clock signal.

MOFXIN Bit (Main Clock Oscillator Forced Oscillation)

This bit controls forced oscillation of the main clock oscillator. Although transitions to software standby or deep software standby normally stop oscillation of the main clock oscillator, setting this bit to 1 forces oscillation by the main clock oscillator so that it can still be used as the clock source for the RTC in either standby mode.
When changing the value of the MOSCCR.MOSTP bit or MOFXIN bit, execute subsequent instructions after reading the bit and checking that its value has actually been updated (refer to (2) Notes on writing to I/O registers, in section 5, I/O Registers).

MODRV2[1:0] Bits (Main Clock Oscillator Driving Ability 2 Switching)

These bits switch the driving ability of the main clock oscillator.
Specify the driving ability according to the frequency of a crystal connected to the main clock oscillator.
The frequency ranges specified in the bit description of the MODRV2[1:0] bits are the reference values of the crystal with capacitive load of 8 pF. A setting value may not fit within the frequency range depending on a crystal. Use values recommended by the resonator manufacturer.
In case of a ceramic resonator, it may be better to select lower frequency range than the frequency of the resonator (for example, specify 10b instead of 01b when a ceramic resonator with the frequency range from 16.1 to 20 MHz is used). Use values recommended by the resonator manufacturer.

MOSEL Bit (Main Clock Oscillator Switching)

This bit switches the source for the main clock oscillator.

9.2.22 High-Speed On-Chip Oscillator Power Supply Control Register (HOCOPCR)

Address(es): 0008 C294h

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

Bit	Symbol	Bit Name	Description	R/W
b0	HOCOPCNT	High-Speed On-Chip Oscillator Power Supply Control	0: Turns the power supply of the HOCO on. 1: Turns the power supply of the HOCO off.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

HOCOPCNT Bit (High-Speed On-Chip Oscillator Power Supply Control)

This bit controls the power supply for the HOCO.

When this bit is set to 0, the power supply of the HOCO is turned on, enabling oscillation.

When this bit is set to 1, the power supply of the HOCO is turned off, reducing power consumption.

When setting the HOCOPCNT bit to 1, set the HOCO stop bit in the high-speed on-chip oscillator control register (HOCOCR.HCSTP) to 1 (HOCO stopped) beforehand.

After the HOCOPCNT bit is changed from 1 to 0, oscillation settling time is required before the HOCOCR.HCSTP bit is set to 0. For details, refer to section 56, Electrical Characteristics.

Do not change the value of the HOCOPCNT bit in the following cases:

- When the HOCO is selected as the clock source by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).
- When the setting of the operating power control mode select bits in the operating power control register (OPCCR.OPCM[2:0]) is for low-speed operating mode 1 or 2.

9.2.23 CLKOUT Output Control Register (CKOCR)

Address(es): 0008 003Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CKOSTP	CKODIV[2:0]			—	CKOSEL[2:0]			—	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	CKOSEL[2:0]	CLKOUT Output 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
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14 to b12	CKODIV[2:0]	CLKOUT Output Divisor Select	b14 b12 0 0 0: ×1/1 0 0 1: ×1/2 0 1 0: ×1/4 0 1 1: ×1/8 1 0 0: ×1/16 Settings other than above are prohibited.	R/W
b15	CKOSTP	CLKOUT Output Stop Control	0: CLKOUT pin output enabled*1 1: CLKOUT pin output stopped (fixed to the low level)	R/W

Note 1. Setting of the pin function control register and port mode register for the corresponding pin is also required.

The CKOCR register sets the clock signal for output from the CLKOUT pin.

CKOSEL[2:0] Bits (CLKOUT Output Source Select)

The CKOSEL[2:0] Bits select the clock source for output from the CLKOUT pin from among the LOCO clock, HOCO clock, main clock, sub-clock, and PLL clock.

To change the clock source, start by setting the CKOSTP bit to 1. Selecting a clock source that is stopped is prohibited.

CKODIV[2:0] Bits (CLKOUT Output Divisor Select)

The CKODIV[2:0] bits select the divisor for the clock selected in the CKOSEL[2:0] bits.

To change the ratio, start by setting the CKOSTP bit to 1.

CKOSTP Bit (CLKOUT Output Stop Control)

The CKOSTP bit controls the output from the CLKOUT pin.

Setting the bit to 0 leads to output of the selected clock. Setting the bit to 1 drives the output to the low level.

Changing the value of the CKOSTP bit while the clock is oscillating may generate a glitch in the output.

9.2.24 Sub-Clock Oscillator Control Register 2 (SOSCCR2)

Address(es): 0008 CC00h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SOSTP 2
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SOSTP2	Sub-Clock Oscillator Stop 2*1	0: The sub-clock oscillator is controlled by the SOSCCR.SOSTP and RTC.RCR3.RTCEN bits. 1: The 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

Note: This register can only be initialized only by the backup domain reset. It is not affected by the MCU reset.
Note 1. Do not set this bit to 0 again after having set it to 1.

SOSTP2 Bit (Sub-Clock Oscillator Stop 2)

The SOSTP 2 bit is the bit that stops the subclock oscillator.

For the relationship between the settings of the SOSTP2 bit, SOSCCR.SOSTP bit, and RCR3.RTCEN bit and the operation of the subclock oscillator, refer to section 12.3.2, Sub-Clock Oscillator for the relations between bit settings and the state of the sub-clock oscillator.

If you want to stop the sub clock oscillator by setting the SOSTP2 bit to 1 , follow the flowchart example in Figure 9.17.

9.2.25 Backup Domain Sub-Clock Control Register (BKSCCR)

Address(es): 0008 CC01h

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

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	SOSEL	Sub-Clock Oscillator Select	0: The internal sub-clock oscillator is selected (use the XCIN and XCOUT pins). 1: The external sub-clock oscillator is selected (input through the EXCIN pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: This register can only be initialized only by the backup domain reset. It is not affected by the MCU reset.

The BKSCCR register is used to select the source of the sub-clock provided to each module in the backup domain and the remote-control signal receiver (REMC).

SOSEL Bit (Sub-Clock Oscillator Select)

The SOSEL bit is used to select the source of the sub-clock provided to each module in the backup domain and the remote-control signal receiver (REMC).

9.2.26 High-Speed On-Chip Oscillator Trimming Registers n (HOCOTRRn) (n = 0 to 2)

Address(es): HOCOTRR0 007F B0E0h, HOCOTRR1 007F B0E4h, HOCOTRR2 007F B0E8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	HOCOTRD[8:0]								
Value after reset:	1	1	1	1	1	1	1	*1	*1	*1	*1	*1	*1	*1	*1	*1

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	HOCOTRD[8:0]	High-Speed On-Chip Oscillator Frequency Adjustment	b8 b0 0 0 0 0 0 0 0 0 0: 0 (Frequency: Low) 0 0 0 0 0 0 0 0 1: 1 : : 1 1 1 1 1 1 1 1 0: 510 1 1 1 1 1 1 1 1 1: 511 (Frequency: High)	R/W
b31 to b9	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note: When FLL is to be used, Do not rewrite this register.
Note 1. Unique value for each chip

HOCOTRD[8:0] Bit (High-Speed On-Chip Oscillator Frequency Adjustment)

Set the frequency trimming value for the high-speed on-chip oscillator. The larger the setting, the higher the frequency becomes. The trimming value is adjusted at shipment on the specified conditions and the value after a reset varies with the chips.

The value of these bits is initialized to the value at shipment by resetting the MCU. When re-writing the value of these bits, start by setting the HOCOCR2.HCFRQ[1:0] bits and then set the corresponding register as follows. After having re-written this register, wait for 65 µs until oscillation becomes stable.

- When the setting of the HOCOCR2.HCFRQ[1:0] bits is 00b (16 MHz), re-write the value of the HOCOTRR0.HOCOTRD[8:0] bits to trim the HOCO frequency.
- When the setting of the HOCOCR2.HCFRQ[1:0] bits is 01b (18 MHz), re-write the value of the HOCOTRR1.HOCOTRD[8:0] bits to trim the HOCO frequency.
- When the setting of the HOCOCR2.HCFRQ[1:0] bits is 10b (20 MHz), re-write the value of the HOCOTRR2.HOCOTRD[8:0] bits to trim the HOCO frequency.

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 9.6 shows an example of connecting a crystal.
Connect capacitors referring to the capacitive load of the crystal to be used. In addition, a damping resistor Rd should be added, if necessary. The values of capacitors and resistor vary depending on the resonator and the oscillator driving ability. Use values recommended by the resonator manufacturer. If use of an external feedback resistor (Rf) is directed by the resonator manufacturer, insert an Rf between EXTAL and XTAL by following the instruction.
When connecting a resonator to supply the clock, the frequency of the crystal must be in the frequency range of the resonator for the main clock oscillator described in Table 9.1.
When a resonator is connected, setting the MOFCR.MODRV2[1:0] bits (Main Clock Oscillator Driving Ability 2 Switching) is required.
The frequency ranges that are specified in the bit description of the MODRV2[1:0] bits are the reference values of the crystal with capacitive load of C_L = 8 pF. The setting value may not fit within the frequency range depending on a crystal. Use values recommended by the resonator manufacturer.
In case of a ceramic resonator, it may be better to select lower frequency range than the frequency of the resonator. (For example, specify 10b instead of 01b when a ceramic resonator with the frequency range of 16.1 to 20 MHz is used). Use values recommended by the resonator manufacturer.

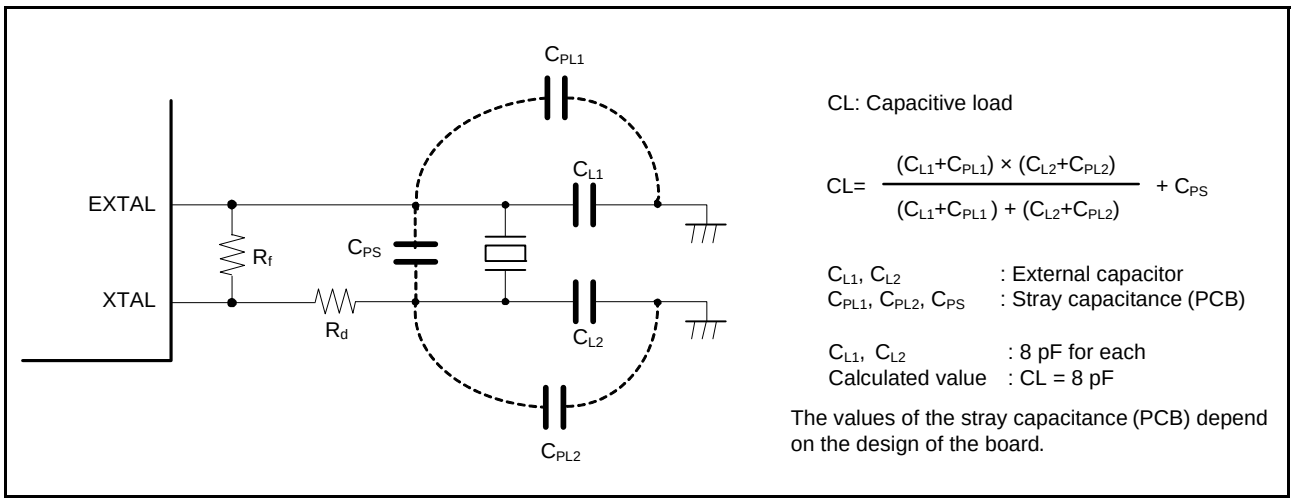


Figure 9.6 Example of Crystal Connection

Table 9.4 Damping Resistance (Reference Values)

Frequency (MHz)	8	12	16	20	24
Rd (Ω)	0	0	0	0	0

Figure 9.7 shows an equivalent circuit of the crystal resonator. Use a crystal resonator that has the characteristics shown in Table 9.5.

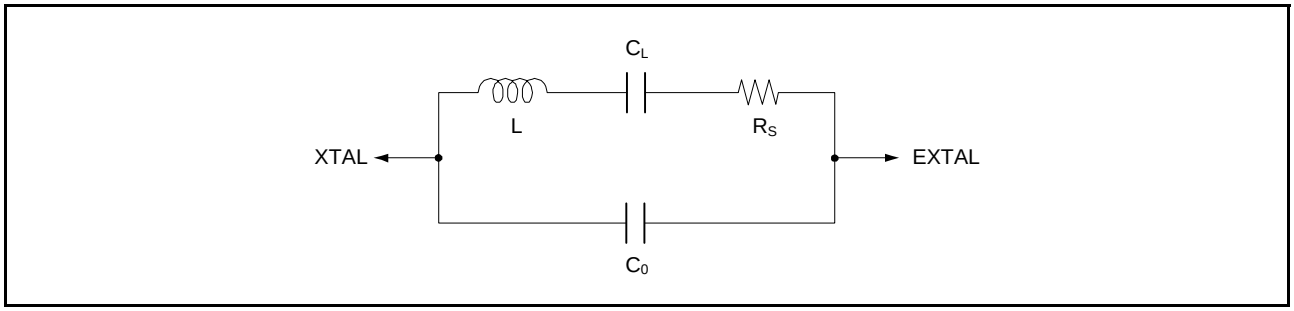


Figure 9.7 Equivalent Circuit of Crystal Resonator

Table 9.5 Crystal Characteristics (Reference Values)

Frequency (MHz)	8	12	16	20	24
R_S max (Ω)	300	100	80	50	50

9.3.2 External Clock Input

Figure 9.8 shows examples of connection of external clock input. Set the MOFCR.MOSEL bit to 1 and open the XTAL pin to operate the oscillator by inputting an external clock signal.

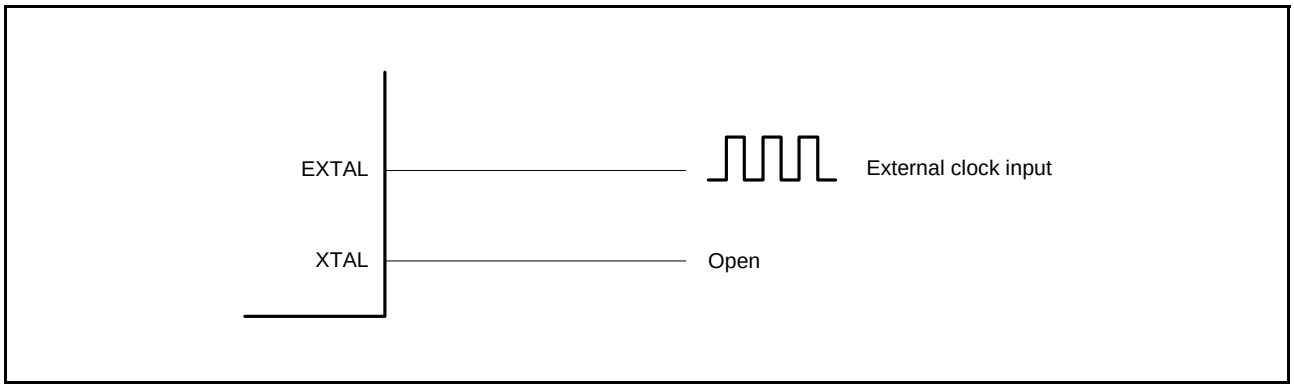


Figure 9.8 Example of Connection of External Clock

9.3.3 Notes on the External Clock Input

The frequency of the external clock input can only be changed while the main clock oscillator is stopped. Do not change the frequency of the external clock input while the setting of the main clock oscillator stop bit (MOSCCR.MOSTP) is 0 (making the main clock oscillator run) or that of the main clock oscillator forced oscillation bit (MOFCR.MOFXIN) is 1 (forcing the main clock oscillator to run).

9.4 Sub-Clock Oscillator

To supply a clock to the sub-clock oscillator, connect a crystal resonator.

9.4.1 Connecting 32.768-kHz Crystal Resonator

To supply a clock to the sub-clock oscillator, connect a 32.768-kHz crystal resonator, as shown in Figure 9.9.
A damping resistor Rd should be added, if necessary. Since the resistor values vary depending on the resonator and the oscillation driving ability, use values recommended by the resonator manufacturer. If use of an external feedback resistor (Rf) is directed by the resonator manufacturer, insert an Rf between XCIN and XCOUT by following the instruction.
When connecting a resonator to supply the clock, the frequency of the resonator should be in the frequency range of the resonator for the sub-clock oscillator described in Table 9.1.

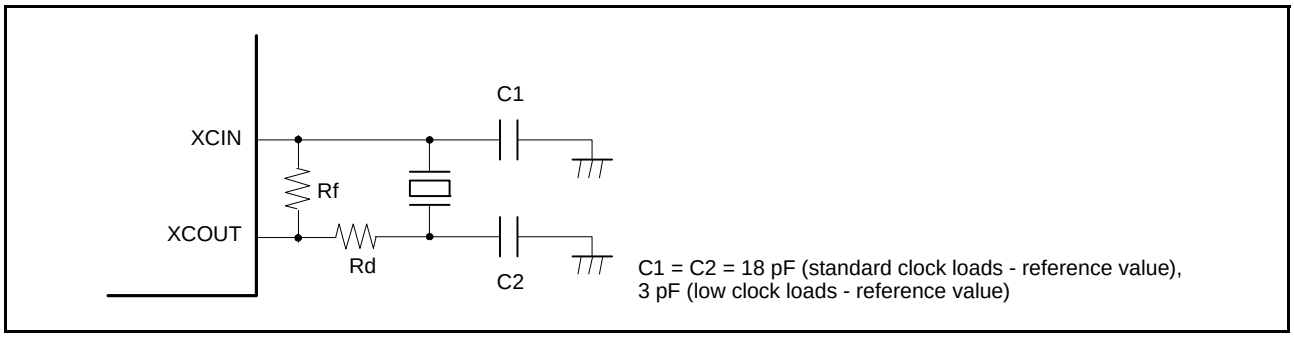


Figure 9.9 Connection Example of 32.768-kHz Crystal Resonator

Figure 9.10 shows an equivalent circuit for the 32.768-kHz crystal resonator. Use a crystal resonator that has the characteristics listed in Table 9.6.

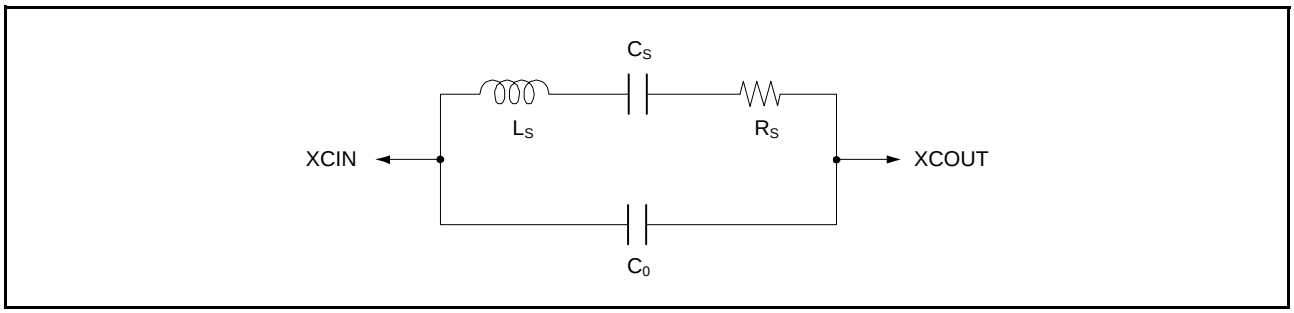


Figure 9.10 Equivalent Circuit for Crystal Resonator

Table 9.6 Crystal Resonator Characteristics (Reference Values)

Frequency (kHz)	32.768 (Low Clock Loads)	32.768 (Standard Clock Loads)
R_S max (kΩ)	60	60

Standard Clock Loads: 7 to 12.5 pF
Low Clock Loads: 5 pF or less

9.4.2 Handling of Pins when Sub-Clock is Not Used

If the sub-clock is not in use, connect the XCIN pin to VSS via a resistor (to pull VSS down) and leave the XCOUT pin open-circuit as shown in Figure 9.11.
In addition, if an oscillator is not connected, set the SOSCCR2.SOSTP2 bit to 1 after setting the SOSCCR.SOSTP bit to 1 (stopping the oscillator) and the RCR3.RTCEN bit to 0 (stopping the sub-clock oscillator). The value of some RTC registers related to the sub-clock will be undefined after a cold start. Accordingly, be sure to set these bits after a cold start.

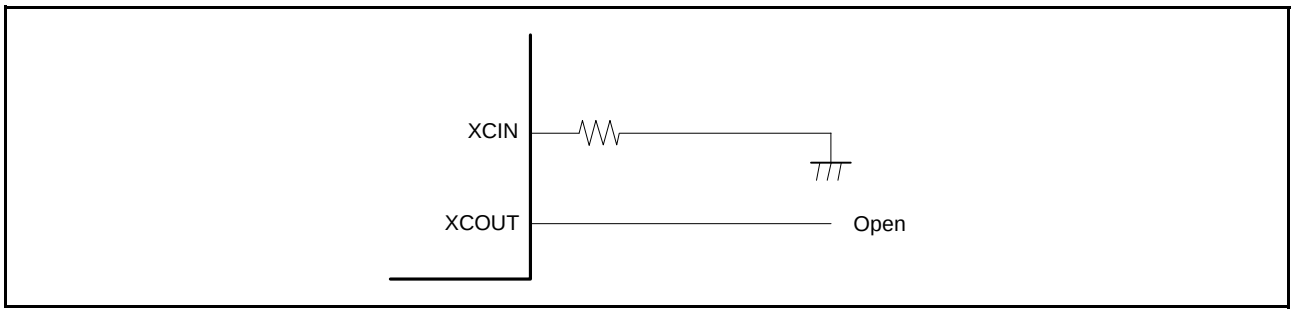


Figure 9.11 Pin Handling when Sub-Clock is Not Used

9.5 Oscillation Stop Detection Function

9.5.1 Oscillation Stop Detection and Operation after Detection

The oscillation stop detection function is used to detect the main clock oscillator stop and to supply LOCO clock pulses from the low-speed on-chip oscillator as the system clock source instead of the main clock or PLL clock. When the HOCO is selected as the clock source for the PLL and the PLL clock is selected as the system clock, the system clock is not switched to the LOCO even if stopping of the main clock is detected.

An oscillation stop detection interrupt request can be generated when an oscillation stop is detected. In addition, the MTU output can be forcedly driven to the high-impedance on the detection. For details, refer to section 24, Multi-Function Timer Pulse Unit 3 (MTU3a) and section 25, Port Output Enable 3 (POE3a).

In the MCU, the input clock remaining at a given level over a certain period due, for example, to a malfunction of the main clock oscillator, is the criterion to detect stopping of the main clock. For details of the detection period, refer to Table 56.59, Oscillation Stop Detection Circuit Characteristics.

When an oscillation stop is detected, the main clock or PLL clock selected by the clock source select bits (SCKCR3.CKSEL[2:0]) is switched to the LOCO clock by the corresponding selectors in the former stage.

Therefore, on detection of oscillation stopping while the PLL clock or system clock is selected as the source of the main clock, the system clock source is switched to the LOCO clock without a change of CKSEL[2:0].

Switching between the main clock and LOCO clock or between the PLL clock and LOCO clock is controlled by the oscillation stop detection flag (OSTDSR.OSTDF). The clock source is switched to the LOCO clock when the OSTDF flag is 1, and is switched to the main clock or PLL clock again when the OSTDF flag is set to 0. At this time, if the main clock or PLL clock is selected with the CKSEL[2:0] bits, the OSTDF flag cannot be set to 0. To switch the clock source to the main clock or PLL clock again after the oscillation stop detection, set the CKSEL[2:0] bits to a clock source other than the main clock or PLL clock and clear the OSTDF flag to 0. After that, check that the OSTDF flag is not 1, and then set the CKSEL[2:0] bits to the main clock or PLL clock after the specified oscillation settling time has elapsed.

After a reset is released, the main clock oscillator is stopped and the oscillation stop detection function is disabled. To enable the oscillation stop detection function, activate the main clock oscillator and write 1 to the oscillation stop detection function enable bit (OSTDCR.OSTDE) after confirming that the OSCOVFSR.MOOVF flag or OSCOVFSR.PLOVF flag have been set to 1.

The oscillation stop detection function is provided against the main clock stop by an external cause. Therefore, the oscillation stop detection function should be disabled before the main clock oscillator is stopped by the software or a transition is made to software standby mode or deep software standby mode.

The clocks that are switched to the LOCO clock by the oscillation stop detection are: the main clock, PLL clock, CAC main clock (CACMCLK), CAN clock (CANMCLK), which are provided as the system clock sources. The main clock as the RTC main clock source (RTCMCLK) is not switched to the LOCO clock.

Note that the frequencies of the derived clock signals after switching to the LOCO depends on the settings of the system clock control registers (SCKCR, SCKCR2, or SCKCR3).

Figure 9.12 shows an example of a flowchart for initialization of the oscillation stop detection function.

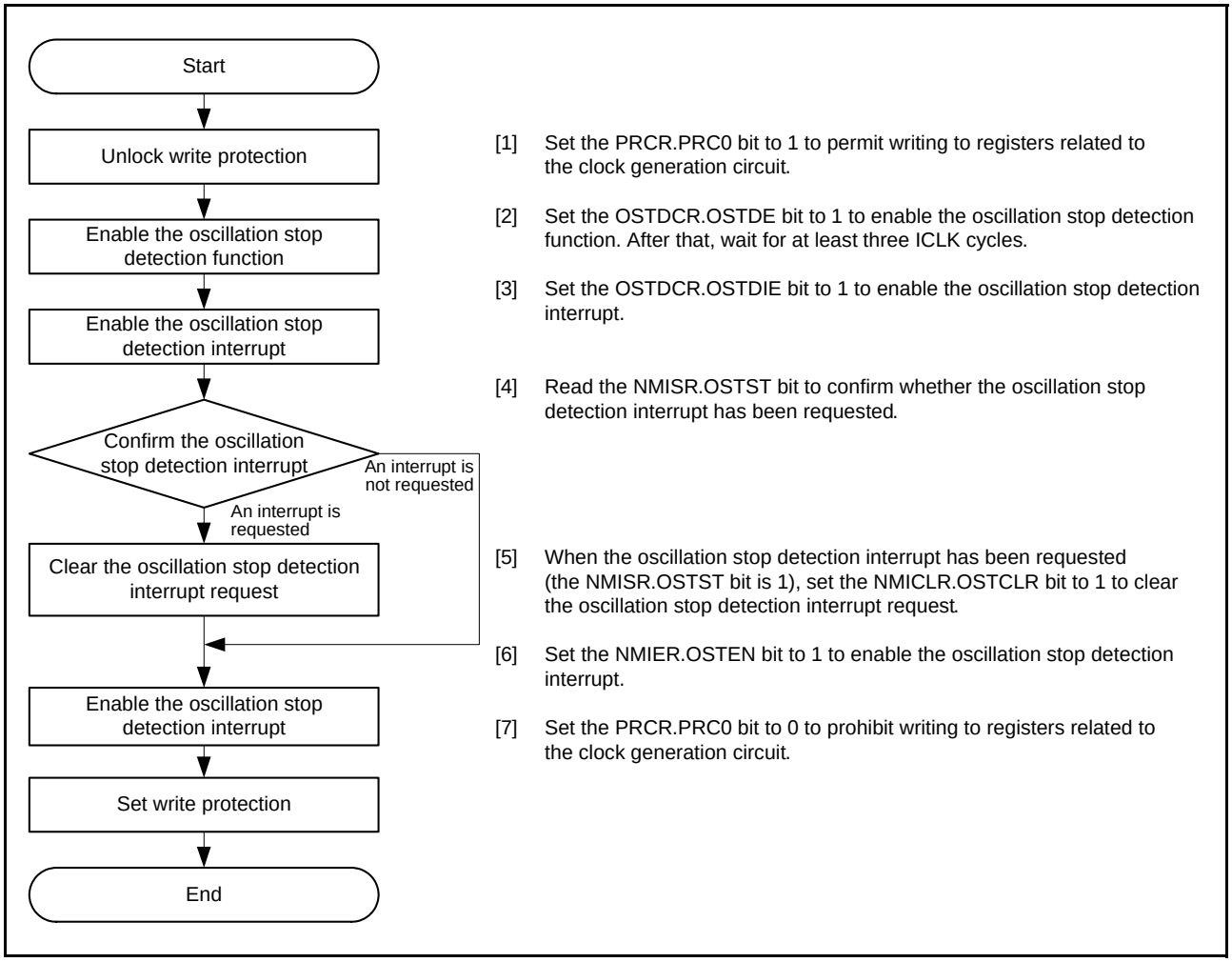
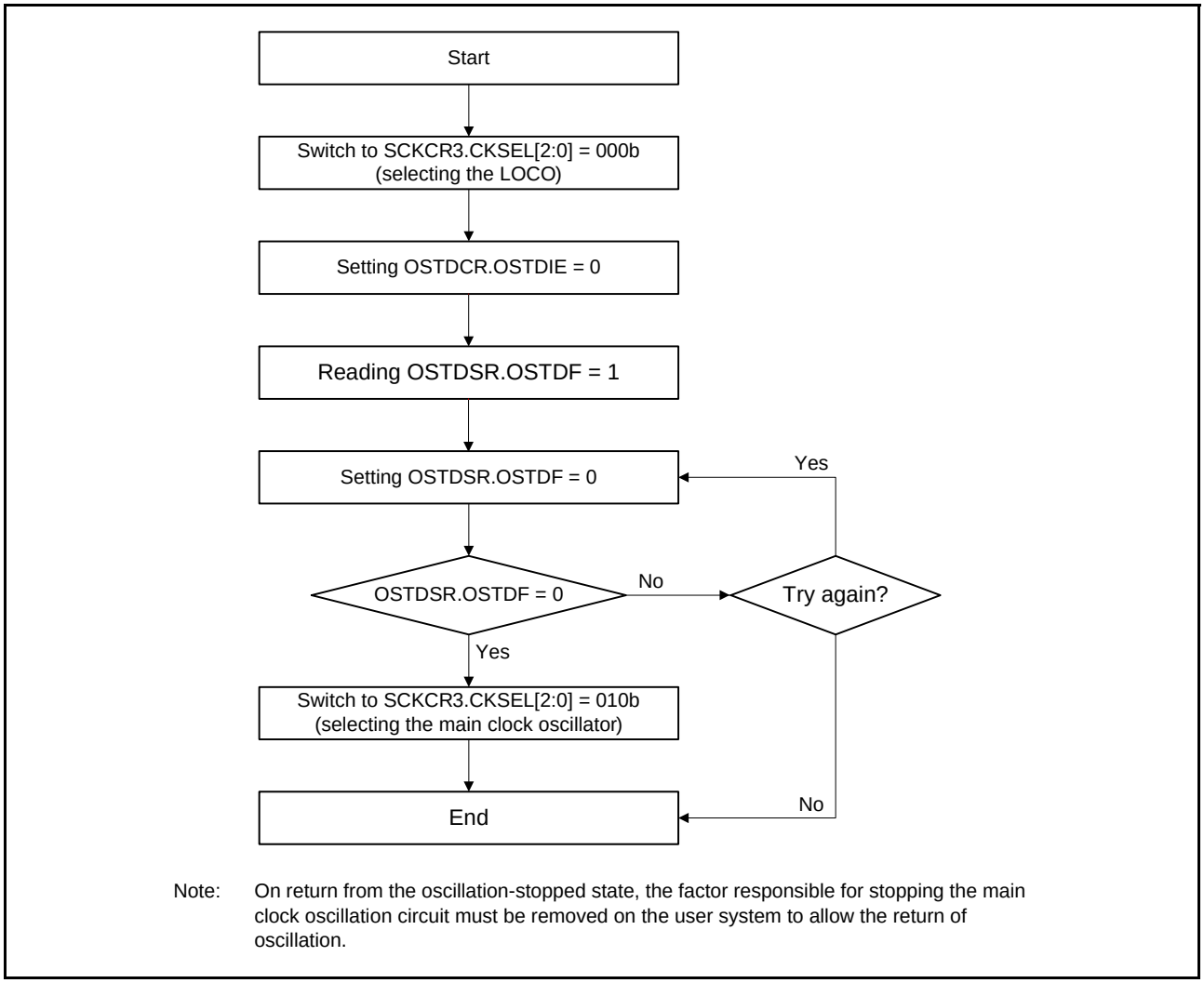


Figure 9.12 Flowchart Example for Initialization of Oscillation Stop Detection Function



Note: On return from the oscillation-stopped state, the factor responsible for stopping the main clock oscillation circuit must be removed on the user system to allow the return of oscillation.

Figure 9.13 Flowchart Example for Recovery from Detection of Oscillator Stop

9.5.2 Oscillation Stop Detection Interrupts

An oscillation-stop detection interrupt (OSTDI) will be generated if the oscillation-stop detection flag (OSTDSR.OSTDF) becomes 1 while the oscillation-stop detection interrupt enable bit in the oscillation stop detection control register (OSTDCR.OSTDIE) is 1 (enabling interrupt generation on oscillation stop detection). At this time, the main clock oscillator stop is notified to the port output enable 3 (POE). On accepting the notification of the oscillation stop, the POE sets the OSTST high-impedance flag in input level control/status register 6 (ICSR6.OSTSTF) to 1. After the oscillation stop is detected, wait for at least 10 cycles of PCLKB before writing to this ICSR6.OSTSTF flag. When the OSTDSR.OSTDF flag requires clearing, do so after clearing the oscillation-stop detection interrupt enable bit in the oscillation stop detection control register (OSTDCR.OSTDIE). Wait for at least two cycles of PCLKB clock before again setting the OSTDCR.OSTDIE bit to 1. According to the number of cycles for access to read out a given I/O register, waiting time longer than two cycles of PCLKB may have to be secured.

If the oscillation stop detection interrupt is to be used as a non-maskable interrupt, since non-maskable interrupts are disabled in the initial state after a reset release, set the corresponding bit in the NMIER register to 1 by software to enable non-maskable interrupts. If it is to be used as a maskable interrupt, do not change the value of the NMIER register from the value after a reset. For details, refer to section 15, Interrupt Controller (ICUE).

9.6 PLL Circuit

The PLL circuit has a function to multiply the frequency from the oscillator.

9.7 Internal Clock

Clock sources of internal clock signals are the main clock, sub-clock, HOCO clock, LOCO clock, PLL clock, dedicated clock for the IWDT, and the JTAG external clock. The internal clocks listed in the table below are produced from these sources.

Frequencies of the internal clocks are set by the combination of the divisors selected by the FCK[3:0], ICK[3:0], BCK[3:0], PCKA[3:0], PCKB[3:0], PCKC[3:0], and PCKD[3:0] bits in SCKCR, the BCLKDIV bit in BCKCR, the UCK[3:0] bits in SCKCR2, the clock source selected by the CKSEL[2:0] bits in SCKCR3, the bits that select the frequency of the PLL circuit (STC[5:0] and PLIDIV[1:0] in PLLCR), and the HCFRQ[1:0] bits in HOCOCR2. If the value of any of these bits is changed, subsequent operation will be at the frequency determined by the new value.

Table 9.7 Internal Clocks and Supply Destination Modules

	Type of Internal Clock	Clock Name	Supply Destination Module
1	System clock	ICLK	CPU, code flash memory, RAM, ICU, BSC, DMAC, DTC, EXDMAC, MPU, QSPIX
2	Peripheral module clocks	PCLKA	MTU, SCIm, RSCI, RSPI, RSPIA, RIICHS
		PCLKB	TPU, PPG, TMR, CMT, CMTW, RTC, WDT, IWDT, POE3, SCIk, SCIh, RIIC, SSIE, CAN, USB, SDHI, CTSU, REMC, S12AD, temperature sensor, CRC, DOC, CAC, Trusted Secure IP, standby RAM, I/O, MPC, ICU
		PCLKC	S12AD (unit 0)
		PCLKD	S12AD (unit 1)
3	Flash-IF clock	FCLK	Data flash memory, code flash memory
4	External bus clock	BCLK	BSC, I/O
5	SDRAM clock	SDCLK	I/O
6	USB clock	UCLK	USB
7	CAN clock	CANMCLK	CAN
8	CLKOUT clock	CLKOUT	I/O
9	CAC clocks	CACMCLK (Main clock)	CAC
		CACSCLK (Sub clock)	
		CACHCLK (HOCO clock)	
		CACLCLK (LOCO clock)	
		CACILCLK (IWDT-dedicated clock)	
10	RTC clocks	RTCMCLK (Main clock)	RTC
		RTCSCLK (Sub clock)	
11	REMC clock	REMSCLK (Sub clock)	REMC
12	VBATT clock	VBATCLK	VBATT
13	IWDT-dedicated clock	IWDTCLK	IWDT, CAC
14	JTAG clock	JTAGTCK	Boundary scan

9.7.1 System Clock

The system clock (ICLK) is used as the operating clock of the CPU, DMAC, DTC, QSPIX, code flash memory, and RAM.
The ICLK frequency is specified by the ICK[3:0] bits in SCKCR, the CKSEL[2:0] bits in SCKCR3, the STC[5:0], and PLIDIV[1:0] bits in PLLCR, and the HCFRQ[1:0] bits in HOCOCR2.

9.7.2 Peripheral Module Clock

The peripheral module clocks (PCLKA, PCLKB, PCLKC, and PCLKD) are the operating clocks for use by peripheral modules.
The frequency of the given clock is specified by the PCKA[3:0], PCKB[3:0], PCKC[3:0], and PCKD[3:0] bits in SCKCR, the CKSEL[2:0] bits in SCKCR3, the STC[5:0], and PLIDIV[1:0] bits in PLLCR, and the HCFRQ[1:0] bits in HOCOCR2. The peripheral module clocks can be set to frequencies above that of the system clock.

9.7.3 Flash-IF Clock

The flash-interface clock (FCLK) is used as the operating clock for the flash-memory interfaces. That is, FCLK is used for programming and erasure of the code flash memory and data flash memory, and reading from the data flash memory.
The FCLK frequency is specified by the FCK[3:0] bits in SCKCR, the CKSEL[2:0] bits in SCKCR3, and the STC[5:0] and PLIDIV[1:0] bits in PLLCR, and the HCFRQ[1:0] bits in HOCOCR2.

9.7.4 External Bus Clock

The external bus clock (BCLK) is an operating clock for the external bus controller and EXDMAC. It is also output externally from the BCLK pin to the external bus. When the external bus is enabled, P53 that is function-multiplexed with the BCLK pin cannot be used as an I/O port.
BCLK can be output from the BCLK pin by setting the SCKCR.PSTOP1 bit to 0 and setting the external bus enable bit in the system control register 0 (SYSCR0.EXBE) to 1. Make sure that modification of the SYSCR0.EXBE bit to 1 must always be performed while the PSTOP1 bit in SCKCR is 1.
When the BCKCR.BCLKDIV bit is set to 1, the BCLK clock divided by 2 is output from the BCLK pin.
The BCLK frequency is specified by the BCK[3:0] bits in SCKCR, the CKSEL[2:0] bits in SCKCR3, and the STC[5:0] and PLIDIV[1:0] bits in PLLCR, and the HCFRQ[1:0] bits in HOCOCR2.
A frequency higher than the system clock (ICLK) should not be set for the BCLK.

9.7.5 SDRAM Clock

The SDRAM clock (SDCLK) is an operating clock for the external bus controller. It is output externally from the SDCLK pin for the SDRAM that is connected to the external bus.
When the SCKCR.PSTOP0 bit is set to 0 and the SDCLK enable bit (SDCLKE) in the external bus control register 1 (PFBCR1) is set to 1 (enabling SDCLK output), it selects output of SDCLK on the SDCLK pin. When changing the value of the PFBCR1.SDCLKE bit, make sure that the value of the SCKCR.PSTOP0 bit is 1.
The SDCLK clock frequency is specified by the SCKCR.BCK[3:0], SCKCR3.CKSEL[2:0], PLLCR.STC[5:0] and PLIDIV[1:0], and HOCOCR2.HCFRQ[1:0] bits.
A frequency higher than the system clock (ICLK) should not be set for the SDCLK.

9.7.6 USB Clock

The USB clock (UCLK) is used as the operating clock for USB.
Use the SCKCR2.UCK[3:0], SCKCR3.CKSEL[2:0], PLLCR.STC[5:0], or PLLCR.PLIDIV[1:0] bits to set the UCLK frequency. The UCLK frequency must be set to 48 MHz.

9.7.7 CLKOUT Clock

The CLKOUT clock (CLKOUT) signal can be externally output from the CLKOUT pin.
The settings of the CKOCR.CKOSEL[2:0] and CKODIV[2:0] bits determine the frequency of the CLKOUT signal.

9.7.8 CAN Clock

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

9.7.9 CAC Clock (CACCLK)

The CAC clock (CACCLK) is an operating clock for the CAC.
CACCLK includes CACMCLK generated by the main clock oscillator, CACSCLK generated by the sub-clock oscillator, CACHCLK generated by the high-speed on-chip oscillator, CACLCLK generated by the low-speed on-chip oscillator, CACILCLK generated by the IWDT-dedicated on-chip oscillator, and PCLKB supplied to peripheral modules.

9.7.10 RTC Clock

The RTC clock (RTCSCLK, RTCMCLK) is the operating clock for the RTC.
Two clock signals are usable as RTCSCLK: that generated by the sub-clock oscillator, and the externally input clock.

9.7.11 REMC Clock

The REMC clock (REMSCLK) is the operating clock for the REMC module.
Two clock signals are usable as REMSCLK: that generated by the sub-clock oscillator, and the externally input clock.

9.7.12 VBATT Clock

The VBATT clock (VBATCLK) is for use by the battery backup circuit (VBATT).
Two clock signals are usable as VBATCLK: that generated by the sub-clock oscillator, and the externally input clock.

9.7.13 IWDT-Dedicated Clock

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

9.7.14 JTAG Clock

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

9.8 Clock Source Switching

In this MCU, the clock signal from the LOCO, which oscillates during release from the reset state, is used to start the fetching of CPU instructions after the internal reset time (tRESWT) has elapsed. After that, set up the clock to which the CPU will be switched while it is still driven by the LOCO, and read the oscillation stabilization flag register to confirm that oscillation of the given clock signal is stable before switching to the selected clock source.

(1) Example of procedure for settings to switch the system clock source from the LOCO to the PLL (clock source for the PLL: main clock) after release from an internal reset

1. After release from the internal reset state, set the driving ability by writing to the MODRV2[1:0] bits in the MOFCR register.
2. Set the oscillation settling time for the main clock oscillator by writing to the MSTS[7:0] bits in the MOSCWTCR register.
3. Make the main clock oscillator run by setting the MOSTP bit in the MOSCCR register.
4. When setting ICLK to [60 MHz < ICLK ≤ 120 MHz], set the ROMWT.ROMWT[1:0] bits to 01b.
5. Set the frequency multiplication factor in the PLLCR register (the initial setting for the PLL clock source selects the main clock oscillator).
6. Select PLL operation by writing to the PLL stop control bit in the PLLCR2 register.
7. Confirm that the PLL clock has become stable by reading the PLOVF flag in the OSCOVFSR register.
8. Change the clock signal from the LOCO clock to the PLL clock by writing to the CKSEL[2:0] bits in the SCKCR3 register.

(2) Example of procedure for settings to switch the system clock source from the LOCO to the PLL (clock source for the PLL: HOCO) after release from an internal reset

1. After release from the internal reset state, set the frequency by writing to the HCFRQ[1:0] bits in the HOCOCR2 register.
2. Make the HOCO clock run by setting the HCSTP bit in the HOCOCR register (the initial value depends on the value of the OFS1.HOCOEN bit; if the OFS1.HOCOEN bit is 0, an explicit setting to make the HOCO clock run is not required. The oscillation settling time for the HOCO is 25 cycles of the LOCO).
3. When setting ICLK to [60 MHz < ICLK ≤ 120 MHz], set the ROMWT.ROMWT[1:0] bits to 01b.
4. Set the frequency multiplication factor and set the HOCO clock as the PLL clock source by writing to the PLLCR register.
5. Select PLL operation by writing to the PLL stop control bit in the PLLCR2 register.
6. Confirm that the PLL clock has become stable by reading the PLOVF flag in the OSCOVFSR register.
7. Change the clock signal from the LOCO clock to the PLL clock by writing to the CKSEL[2:0] bits in the SCKCR3 register.

(3) Example of procedure for settings to switch the system clock source from the LOCO to the main clock after release from an internal reset

1. After release from the internal reset state, set the driving ability by writing to the MODRV2[1:0] bits in the MOFCR register.
2. Set the oscillation settling time for the main clock oscillator by writing to the MSTS[7:0] bits in the MOSCWTCR register.
3. Make the main clock oscillator run by setting the MOSTP bit in the MOSCCR register.
4. Confirm that the main clock has become stable by reading the MOOVF flag in the OSCOVFSR register.
5. Change the clock signal from the LOCO clock to the main clock by writing to the CKSEL[2:0] bits in the SCKCR3 register.

(4) Example of procedure for settings to switch the system clock source from the LOCO to the HOCO after release from an internal reset

1. After release from the internal reset state, set the frequency by writing to the HCFRQ[1:0] bits in the HOCOCR2 register.
2. Make the HOCO clock run by setting the HCSTP bit in the HOCOCR register (the initial value depends on the value of the OFS1.HOCOEN bit; if the OFS1.HOCOEN bit is 0, an explicit setting to make the HOCO clock run is not required. The oscillation settling time for the HOCO is 25 cycles of the LOCO).
3. Confirm that the HOCO clock has become stable by reading the HCOVF flag in the OSCOVFSR register.
4. Change the clock signal from the LOCO clock to the HOCO clock by writing to the CKSEL[2:0] bits in the SCKCR3 register.

9.9 Operations Linked by the ELC

9.9.1 Event Signal Output to the ELC

The clock generation circuit is capable of operation linked with another module set in advance when its interrupt request signal is used as an even signal by the event link controller (ELC) on detection of stopping of the main clock oscillation. The clock generation circuit outputs the event signal regardless of the setting of the corresponding oscillation stop detection interrupt enable bit (OSTDCR.OSTDIE). For details, refer to section 21, Event Link Controller (ELC).

9.9.2 Clock Source Switching on Reception of the Event Signal from the ELC

The clock generation circuit is capable of switching the clock source to the low-speed on-chip oscillator in response to the event set in advance when the event specified in the ELSRn register of the ELC occurs.
While this function is in use, clock source switching on return from sleep mode cannot be used. For details, refer to section 11.2.7, Sleep Mode Return Clock Source Switching Register (RSTCKCR).

9.10 Usage Notes

9.10.1 Notes on Clock Generation Circuit

(1) The frequencies of the system clock (ICLK), peripheral module clock (PCLKA to PCLKD), flash-IF clock (FCLK), external bus clock (BCLK), and SDRAM clock (SDCLK) supplied to each module change according to the settings of SCKCR. Each frequency should meet the following:
Select each frequency that is within the operation guaranteed range of clock cycle time (tcyc) specified in AC characteristics of electrical characteristics.
The frequencies must not exceed the ranges listed in Table 9.1.
The peripheral modules operate on the PCLKB and PCLKA. Note therefore that the operating speed of modules such as the timer and SCI varies before and after the frequency is changed.
Furthermore, PCLKC (unit 0) and PCLKD (unit 1) are available as clocks for the A/D converter.
Do not set a higher frequency than PCLKB, which serves as the operating clock of the A/D converter.

(2) The following relation is required between the frequencies of the system clock (ICLK) and external bus clock (BCLK).
ICLK ≥ BCLK
Also, the following relation is required between the frequencies of the peripheral module clocks.
PCLKA ≥ PCLKB; PCLKB ≥ PCLKC; PCLKB ≥ PCLKD

(3) Do not change the clock frequency during external bus access. Furthermore, when access via the external bus is to start after a change to the clock frequency, only start access via the bus after confirming that the change to the frequencies has been completed.

(4) If the clock frequency is to be changed by modifying the value of SCKCR, SCKCR2, SCKCR3, or BCKCR register, wait for writing of the value to the register to be complete and the new frequency to be stable before starting subsequent processing. For the procedure to confirm the completion of writing to I/O registers, refer to (2) Notes on writing to I/O registers, in section 5, I/O Registers.

9.10.2 Note on Rewriting the SCKCR3 Register

When the SCKCR3.CKSEL[2:0] bits have been rewritten, the clock output is temporarily stopped to prevent the switch of the clock source from generating a clock pulse of short duration (glitch). The interrupt controller and event link controller may not detect the following signals that were input during this period.

(1) An external pin interrupt or NMI pin interrupt with a pulse width shorter than 4 cycles of the PCLKB after the switch while the PCLKB frequency is the 1/1 of the clock source (the SCKCR.PCKB[3:0] bits are 0000b).

(2) An external pin interrupt or NMI pin interrupt with a pulse width shorter than 2.5 cycles of the PCLKB after the switch while the PCLKB frequency is the 1/2 of the clock source (the SCKCR.PCKB[3:0] bits are 0001b).

(3) An RTC periodic interrupt or RTC periodic event output when the SCKCR3.CKSEL[2:0] bits are changed to “011b” (sub-clock) while the count source of the RTC is set to the sub-clock (RCR4.RCKSEL bit is 0).

(4) An RTC periodic interrupt or RTC periodic event output when the SCKCR3.CKSEL[2:0] bits are changed to the value other than “100b” (PLL circuit) while the count source of the RTC is set to the main clock (RCR4.RCKSEL bit is 1).

When the external pin interrupt or NMI pin interrupt is in use, input the signal with enough pulse width to exceed the time condition described in (1) and (2). When the RTC periodic interrupt or RTC periodic event output is in use, switch the clock source after a periodic interrupt is generated and before the next periodic interrupt is generated.

9.10.3 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.

9.10.4 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 9.14 to prevent electromagnetic induction from interfering with correct oscillation.

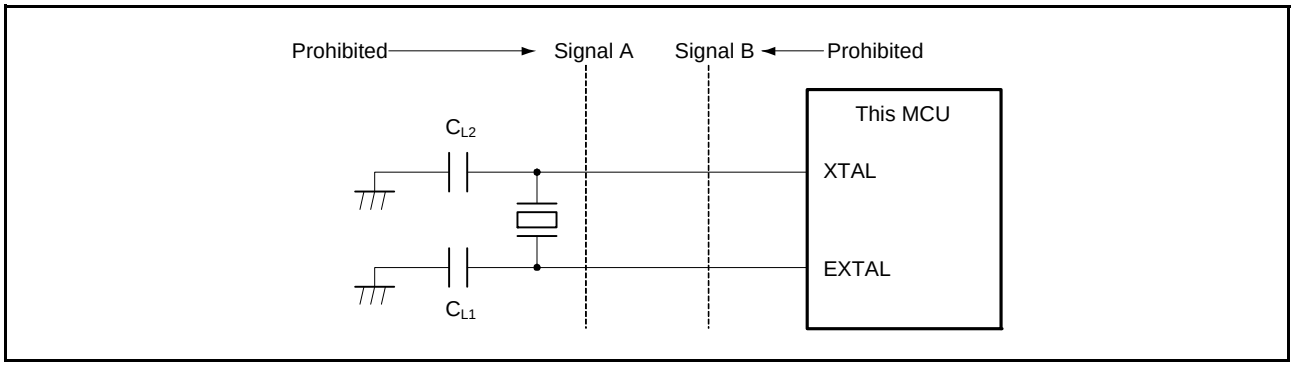


Figure 9.14 Notes on Board Design for Oscillation Circuit (Applies to the Sub-Clock Oscillator, in Case of the Main Clock Oscillator)

9.10.5 Notes on Resonator Connect Pin

When the main clock is not used, the EXTAL and XTAL pins can be used as general ports P36 and P37. When they are used as the general ports, the main clock should be stopped (MOSCCR.MOSTP should be set to 1 and MOFCR.MOFXIN should be set to 0). However, with the system using the main clock, the EXTAL (P36) and XTAL (P37) pins should not be used as output ports.
For the values of registers related to port settings, refer to Table 23.36, 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.

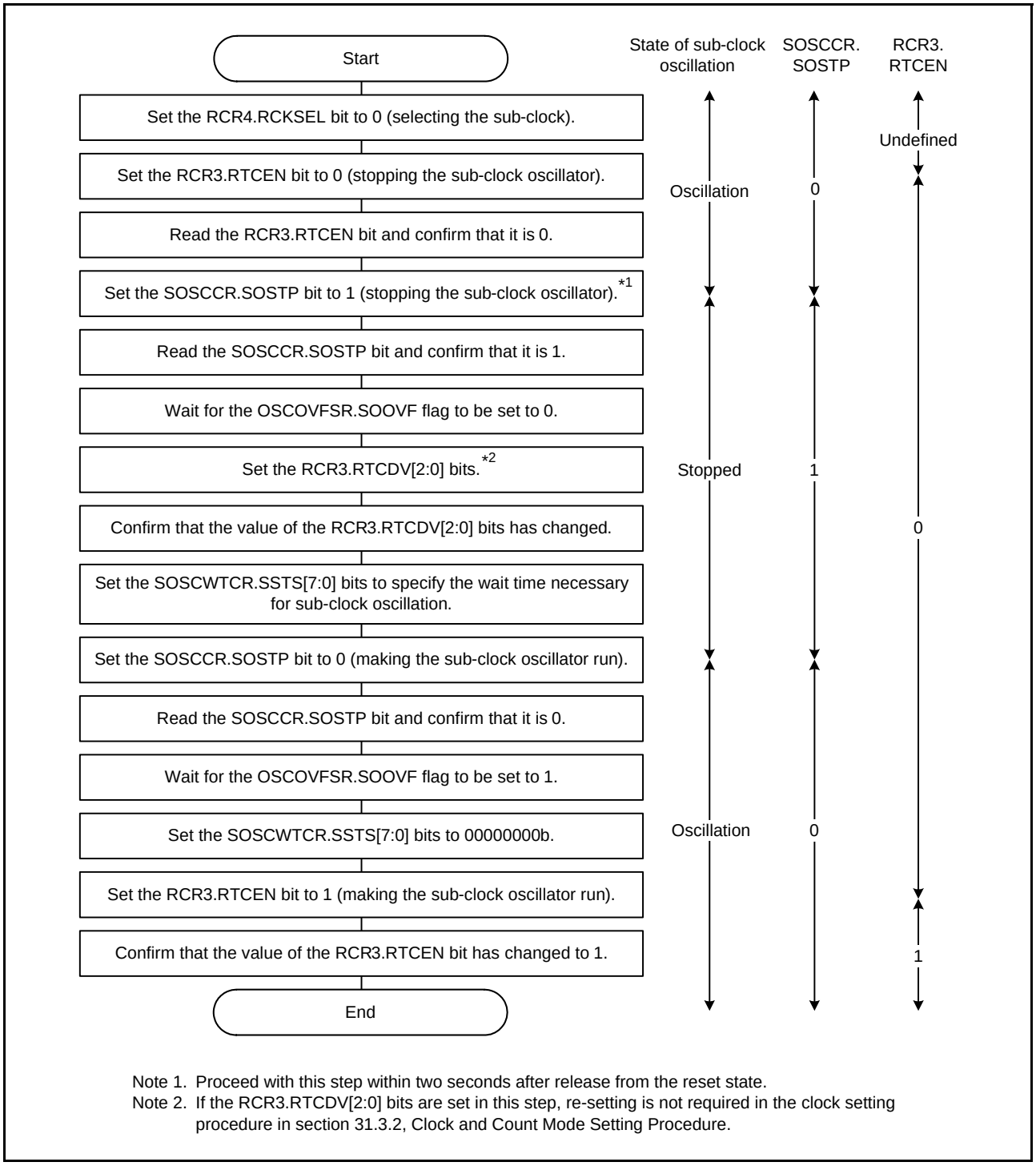
- Programming of flash memory in boot mode (USB interface)*1

Note 1. For the conditions on the oscillator in the various modes, refer to (7) in section 55.20, Usage Notes.

9.10.6 Notes on Sub-Clock Oscillator

The sub-clock can be used as the system clock, as the source to drive counting by the realtime clock, or as both. Accordingly, take note of the following limitations and points for caution regarding the settings, including when the sub-clock is not in use.

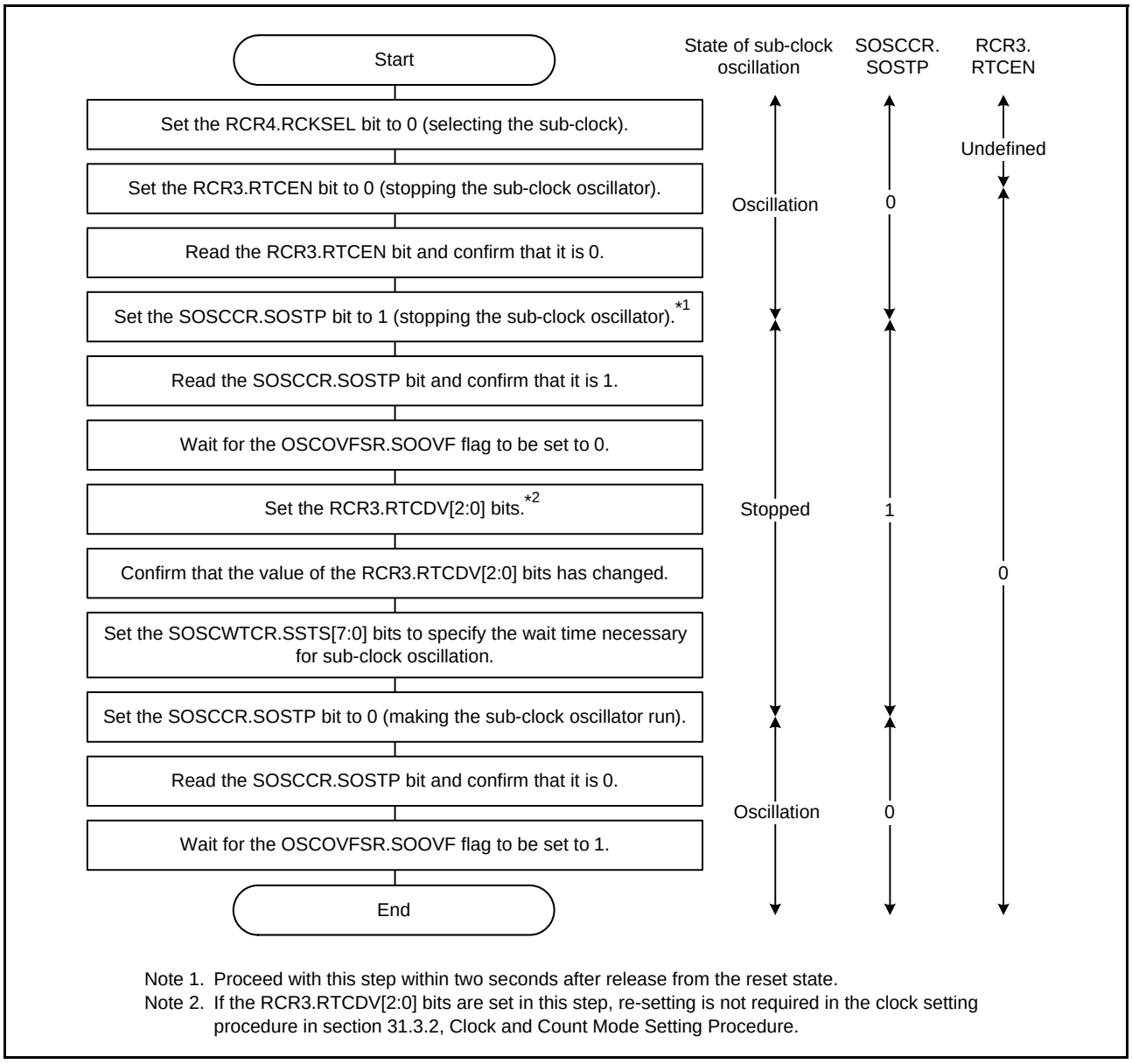
- To select the sub-clock as the system clock, set the SOSCCR.SOSTP bit; to set the sub-clock as the source to drive counting by the realtime clock, use the SOSCCR.SOSTP and RCR3.RTCEN bits. Furthermore, when the sub-clock is to be used as the clock source for counting by the realtime clock, the SOSCWTCR.SSTS[7:0] bits must be set to 00000000b after the oscillation stabilization waiting time has elapsed once the sub-clock has started oscillating.
- Make initial settings according to the example flowchart shown in Figure 9.15 when the sub-clock is to be used as the system clock as well as the source to drive counting by the realtime clock or when the sub-clock is only to be used as the source to drive counting by the realtime clock. After that, make settings by following the clock setting procedure described in section 31.3.2, Clock and Count Mode Setting Procedure.



Note 1. Proceed with this step within two seconds after release from the reset state.
Note 2. If the RCR3.RTCDV[2:0] bits are set in this step, re-setting is not required in the clock setting procedure in section 31.3.2, Clock and Count Mode Setting Procedure.

Figure 9.15 Example Flowchart of Initialization when the Sub-Clock is to be Used as the Source to Drive Counting by the Realtime Clock

- Make initial settings according to the example flowchart shown in Figure 9.16 when the sub-clock is only to be used as the system clock.



Note 1. Proceed with this step within two seconds after release from the reset state.
Note 2. If the RCR3.RTCDV[2:0] bits are set in this step, re-setting is not required in the clock setting procedure in section 31.3.2, Clock and Count Mode Setting Procedure.

Figure 9.16 Example Flowchart for Initialization when the Sub-Clock is to be Used only as the System Clock

- Make initial settings according to the example flowchart shown in Figure 9.17 when the sub-clock is not to be used.

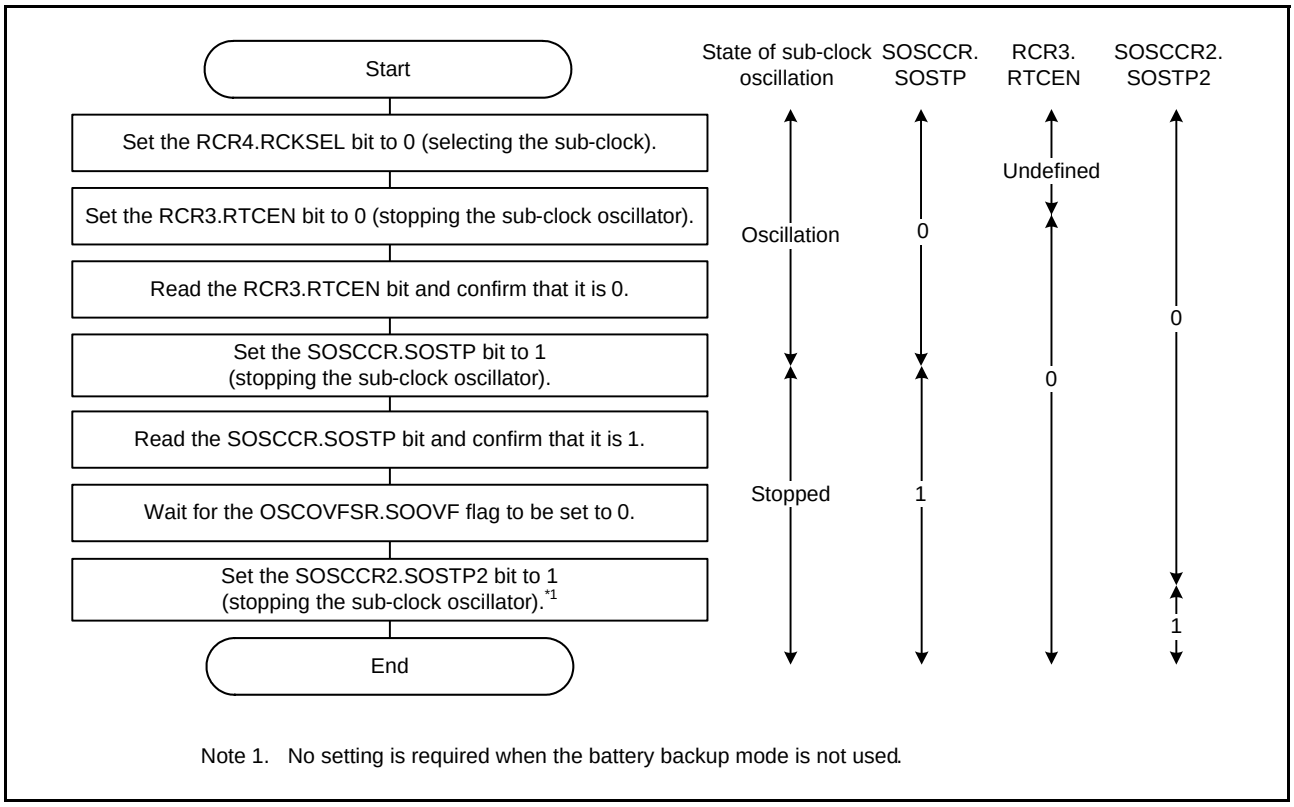


Figure 9.17 Example Flowchart for when the Sub-Clock is not to be Used

- When the sub-clock is used as the system clock, even if the RCR3.RTCEN bit is set to 1 and the sub-clock is already oscillating, wait for the OSCOVFSR.SOOVF flag to be set to 1 after the SOSCCR.SOSTP bit changes from 1 (stopped) to 0 (operating) before starting to use the sub-clock as the system clock.
- The state of the sub-clock control circuit is undefined after a cold start so the sub-clock must be initialized whether or not it is to be used. Setting both the SOSCCR.SOSTP bit and the RCR3.RTCEN bit to the “stopped” setting will initialize the control circuit. For initialization of the RCR3.RTCEN bit, refer to section 31, Realtime Clock (RTCd).
- For the sub-clock oscillator to operate, the RCR3.RTCDV[2:0] bits must also be set. This setting must be made while the sub-clock oscillator is stopped. Writing to these bits during operation is prohibited.
- If the RCR3.RTCEN bit is modified after the SOSCCR.SOSTP bit is modified, or if the SOSCCR.SOSTP bit is modified after the RCR3.RTCEN bit is modified, make sure that first modified bit has been rewritten before the subsequent bit is modified.

9.10.7 Notes on Using a Low CL Crystal Unit

When the RCR3.RTCDV[2:0] bits are 001b (drive capacity for low CL), the oscillator is susceptible to noise. Especially when the signal level of any pin near the XCIN or XCOUT pin is changed, the oscillation accuracy of the sub-clock oscillator may be affected. The accuracy is affected differently depending on the board traces and how the signal level of any pin near the XCIN or XCOUT pin is changed. When designing a board using a low CL crystal unit, refer to the application note “Design Guide for Low CL Sub-clock Circuits” (R01AN1187EJ) to reduce the influence from noise. The following are examples that may significantly affect oscillation accuracy:

(1) When connecting an on-chip debugging emulator to the FINED pin

Since the FINED pin (FINE interface pin) is near the XCIN and XCOUT pins, the oscillation accuracy of the sub-clock oscillator is affected when using the FINED pin in debugging. When using the FINED pin in debugging, set the RCR3.RTCDV[2:0] bits to 001b (drive capacity for low CL) and debug at the room temperature.

(2) When supplying an external clock to the main clock oscillator

When inputting an external clock to the EXTAL pin, the oscillation accuracy of the sub-clock oscillator may be affected. When inputting an inverted external clock to the XTAL pin, the oscillation accuracy will be affected more significantly.

9.10.8 Notes on Products in 48-Pin Packages

The product in the 48-pin package has neither a sub-clock nor an RTC.
For initialization of this product, follow the procedure shown in Figure 9.17, Example Flowchart for when the Sub-Clock is not to be Used.
If a cold start is applied, the states of the bits related to the sub-clock control circuit become undefined. Make the settings to stop the sub-clock control circuit in these bits after a cold start.

10. Clock Frequency Accuracy Measurement Circuit (CAC)

10.1 Overview

The clock frequency accuracy measurement circuit (CAC) counts pulses of the clock to be measured (measurement target clock) within the time generated by the clock to be used as a measurement reference (measurement reference clock), and determines the accuracy depending on whether the number of pulses is within the allowable range.
When measurement is completed or the number of pulses within the time generated by the measurement reference clock is not within the allowable range, an interrupt request is generated.
Table 10.1 lists the specifications of the CAC and Figure 10.1 shows a block diagram of the CAC.

Table 10.1 CAC Specifications

Item	Description
Measurement target clocks	The frequency of the following clocks can be measured. • Main clock • Sub-clock • HOCO clock • LOCO clock • IWDT-dedicated clock (IWDTCLK) • Peripheral module clock B (PCLKB)
Measurement reference clocks	• External clock input to the CACREF pin • Main clock • Sub-clock • HOCO clock • LOCO clock • IWDT-dedicated clock (IWDTCLK) • Peripheral module clock B (PCLKB)
Selectable function	Digital filter function
Interrupt sources	• Measurement end interrupt • Frequency error interrupt • Overflow interrupt
Low power consumption function	Module stop state can be set.

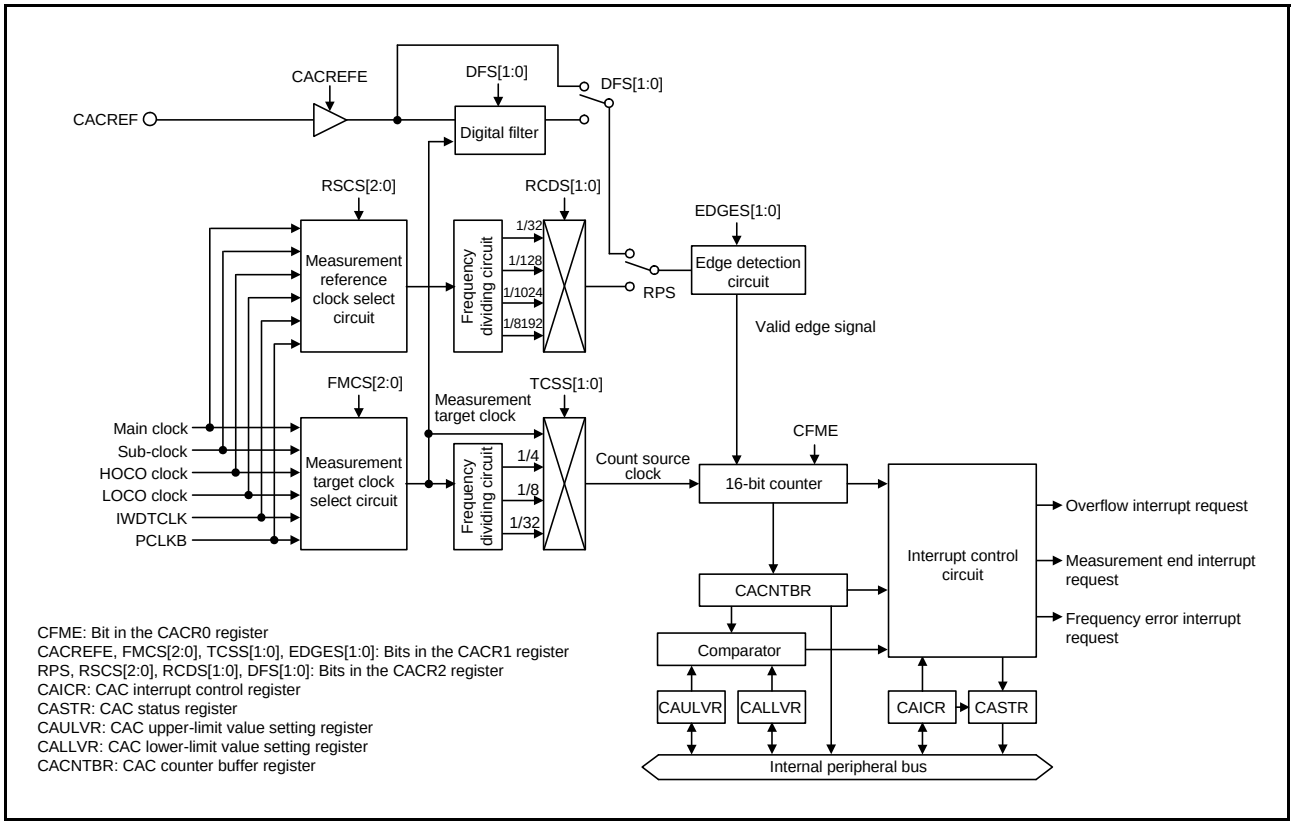


Figure 10.1 CAC Block Diagram

Table 10.2 shows the pin configuration of the CAC.

Table 10.2 Pin Configuration of CAC

Pin Name	I/O	Function
CACREF	Input	Measurement reference clock input pin

10.2 Register Descriptions

10.2.1 CAC Control Register 0 (CACR0)

Address(es): CAC.CACR0 0008 B000h

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

Bit	Symbol	Bit Name	Description	R/W
b0	CFME	Clock Frequency Measurement Enable	0: Clock frequency measurement is disabled. 1: Clock frequency measurement is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CFME Bit (Clock Frequency Measurement Enable)

This bit specifies whether clock frequency measurement is enabled or disabled.

When rewriting this bit, more time is required than other bits for the new value to be reflected in the register. Further write access to this bit are ignored until the current write access is reflected in the register. Read the bit to confirm that the rewrite has been reflected in the register.

10.2.2 CAC Control Register 1 (CACR1)

Address(es): CAC.CACR1 0008 B001h

	b7	b6	b5	b4	b3	b2	b1	b0
	EDGES[1:0]		TCSS[1:0]		FMCS[2:0]			CACREFE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CACREFE	CACREF Pin Input Enable	0: CACREF pin input is disabled. 1: CACREF pin input is enabled.	R/W
b3 to b1	FMCS[2:0]	Measurement Target Clock Select	b3 b1 0 0 0: Main clock 0 0 1: Sub-clock 0 1 0: HOCO clock 0 1 1: LOCO clock 1 0 0: IWDT-dedicated clock (IWDTCLK) 1 0 1: Peripheral module clock B (PCLKB) Settings other than above are prohibited.	R/W
b5, b4	TCSS[1:0]	Timer Count Clock Source Select	b5 b4 0 0: No division 0 1: ×1/4 clock 1 0: ×1/8 clock 1 1: ×1/32 clock	R/W
b7, b6	EDGES[1:0]	Valid Edge Select	b7 b6 0 0: Rising edge 0 1: Falling edge 1 0: Both rising and falling edges 1 1: Setting prohibited	R/W

Note: Set the CACR1 register when the CACR0.CFME bit is 0.

CACREFE Bit (CACREF Pin Input Enable)

This bit specifies whether the CACREF pin input is enabled or disabled.

FMCS[2:0]Bits (Measurement Target Clock Select)

These bits select the measurement target clock whose frequency is to be measured.

TCSS[1:0] Bits (Timer Count Clock Source Select)

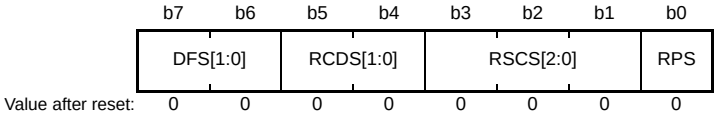
These bits select the count clock source for the clock frequency accuracy measurement circuit.

EDGES[1:0]Bits (Valid Edge Select)

These bits select the valid edge for the reference signal.

10.2.3 CAC Control Register 2 (CACR2)

Address(es): CAC.CACR2 0008 B002h



Bit	Symbol	Bit Name	Description	R/W
b0	RPS	Reference Signal Select	0: CACREF pin input 1: Internal clock (internally generated signal)	R/W
b3 to b1	RSCS[2:0]	Measurement Reference Clock Select	b3 b1 0 0 0: Main clock 0 0 1: Sub-clock 0 1 0: HOCO clock 0 1 1: LOCO clock 1 0 0: IWDT-dedicated clock (IWDTCLK) 1 0 1: Peripheral module clock B (PCLKB) Settings other than above are prohibited.	R/W
b5, b4	RCDS[1:0]	Measurement Reference Clock Frequency Division Ration Select	b5 b4 0 0: ×1/32 clock 0 1: ×1/128 clock 1 0: ×1/1024 clock 1 1: ×1/8192 clock	R/W
b7, b6	DFS[1:0]	Digital Filter Select	b7 b6 0 0: Digital filtering is disabled. 0 1: The sampling clock for the digital filter is the measurement target clock. 1 0: The sampling clock for the digital filter is the measurement target clock divided by 4. 1 1: The sampling clock for the digital filter is the measurement target clock divided by 16.	R/W

Note: Set the CACR2 register when the CACR0.CFME bit is 0.

RPS Bit (Reference Signal Select)

This bit selects whether to use the CACREF pin input or an internal clock (internally generated signal) as the reference signal.

RSCS[2:0]Bits (Measurement Reference Clock Select)

These bits select the clock source for generating the measurement reference clock.

RCDS[1:0]Bits (Measurement Reference Clock Frequency Division Ration Select)

These bits select the frequency division ratio of the measurement reference clock.

DFS[1:0]Bits (Digital Filter Select)

The setting of these bits enables or disables the digital filter and selects its sampling clock.

10.2.4 CAC Interrupt Request Enable Register (CAICR)

Address(es): CAC.CAICR 0008 B003h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	OVFFCL	MENDFCL	FERRFCL	—	OVFIE	MENDIE	FERRIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FERRIE	Frequency Error Interrupt Request Enable	0: Frequency error interrupt request is disabled. 1: Frequency error interrupt request is enabled.	R/W
b1	MENDIE	Measurement End Interrupt Request Enable	0: Measurement end interrupt request is disabled. 1: Measurement end interrupt request is enabled.	R/W
b2	OVFIE	Overflow Interrupt Request Enable	0: Overflow interrupt request is disabled. 1: Overflow interrupt request is enabled.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	FERRFCL	FERRF Clear	When 1 is written to this bit, the CASTR.FERRF flag is cleared. This bit is read as 0.	R/W
b5	MENDFCL	MENDF Clear	When 1 is written to this bit, the CASTR.MENDF flag is cleared. This bit is read as 0.	R/W
b6	OVFFCL	OVFF Clear	When 1 is written to this bit, the CASTR.OVFF flag is cleared. This bit is read as 0.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

FERRIE Bit (Frequency Error Interrupt Request Enable)

This bit specifies whether the frequency error interrupt request is enabled or disabled.

MENDIE Bit (Measurement End Interrupt Request Enable)

This bit specifies whether the measurement end interrupt request is enabled or disabled.

OVFIE Bit (Overflow Interrupt Request Enable)

This bit specifies whether the overflow interrupt request is enabled or disabled.

FERRFCL Bit (FERRF Clear)

Setting this bit to 1 clears the CASTR.FERRF flag.

MENDFCL Bit (MENDF Clear)

Setting this bit to 1 clears the CASTR.MENDF flag.

OVFFCL Bit (OVFF Clear)

Setting this bit to 1 clears the CASTR.OVFF flag.

10.2.5 CAC Status Register (CASTR)

Address(es): CAC.CASTR 0008 B004h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	OVFF	MENDF	FERRF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FERRF	Frequency Error Flag	0: The clock frequency is within the range corresponding to the settings. 1: The clock frequency has deviated beyond the range corresponding to the settings (frequency error).	R
b1	MENDF	Measurement End Flag	0: Measurement is in progress. 1: Measurement has ended.	R
b2	OVFF	Overflow Flag	0: The counter has not overflowed. 1: The counter has overflowed.	R
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FERRF Flag (Frequency Error Flag)

This flag indicates deviation of the clock frequency from the set value (frequency error).

[Setting condition]

- The clock frequency is outside of the setting range.

[Clearing condition]

- 1 is written to the CAICR.FERRFCL bit.

MENDF Flag (Measurement End Flag)

This flag indicates the end of measurement.

[Setting condition]

- Measurement has finished.

[Clearing condition]

- 1 is written to the CAICR.MENDFCL bit.

OVFF Flag (Overflow Flag)

This flag indicates that the counter has overflowed.

[Setting condition]

- The counter has overflowed.

[Clearing condition]

- 1 is written to the CAICR.OVFFCL bit.

10.2.6 CAC Upper-Limit Value Setting Register (CAULVR)

Address(es): CAC.CAULVR 0008 B006h

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

The CAULVR register is a 16-bit readable/writable register that specifies the upper-limit value of the counter used for measuring the frequency. When the frequency rises above the value specified in this register, a frequency error is detected.
Write to this register when the CACR0.CFME bit is 0.
The counter value held in the CACNTBR register can vary with the difference between the phases of the digital filter and edge-detection circuit on the one hand and the signal on the CACREF pin on the other, so ensure that this setting allows an adequate margin.

10.2.7 CAC Lower-Limit Value Setting Register (CALLVR)

Address(es): CAC.CALLVR 0008 B008h

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

The CALLVR register is a 16-bit readable/writable register that specifies the lower-limit value of the counter used for measuring the frequency. When the frequency falls below the value specified in this register, a frequency error is detected.
Write to this register when the CACR0.CFME bit is 0.
The counter value held in the CACNTBR register can vary with the difference between the phases of the digital filter and edge-detection circuit on the one hand and the signal on the CACREF pin on the other, so ensure that this setting allows an adequate margin.

10.2.8 CAC Counter Buffer Register (CACNTBR)

Address(es): CAC.CACNTBR 0008 B00Ah

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

The CACNTBR register is a 16-bit read-only register that retains the counter value at the time a valid reference signal edge is input.

10.3 Operation

10.3.1 Measuring Clock Frequency

The clock frequency accuracy measurement circuit measures the clock frequency using the CACREF pin input or the internal clock as a reference. Figure 10.2 shows an operating example of the clock frequency accuracy measurement circuit.

The clock frequency accuracy measurement circuit operates as shown below when measuring the clock frequency.

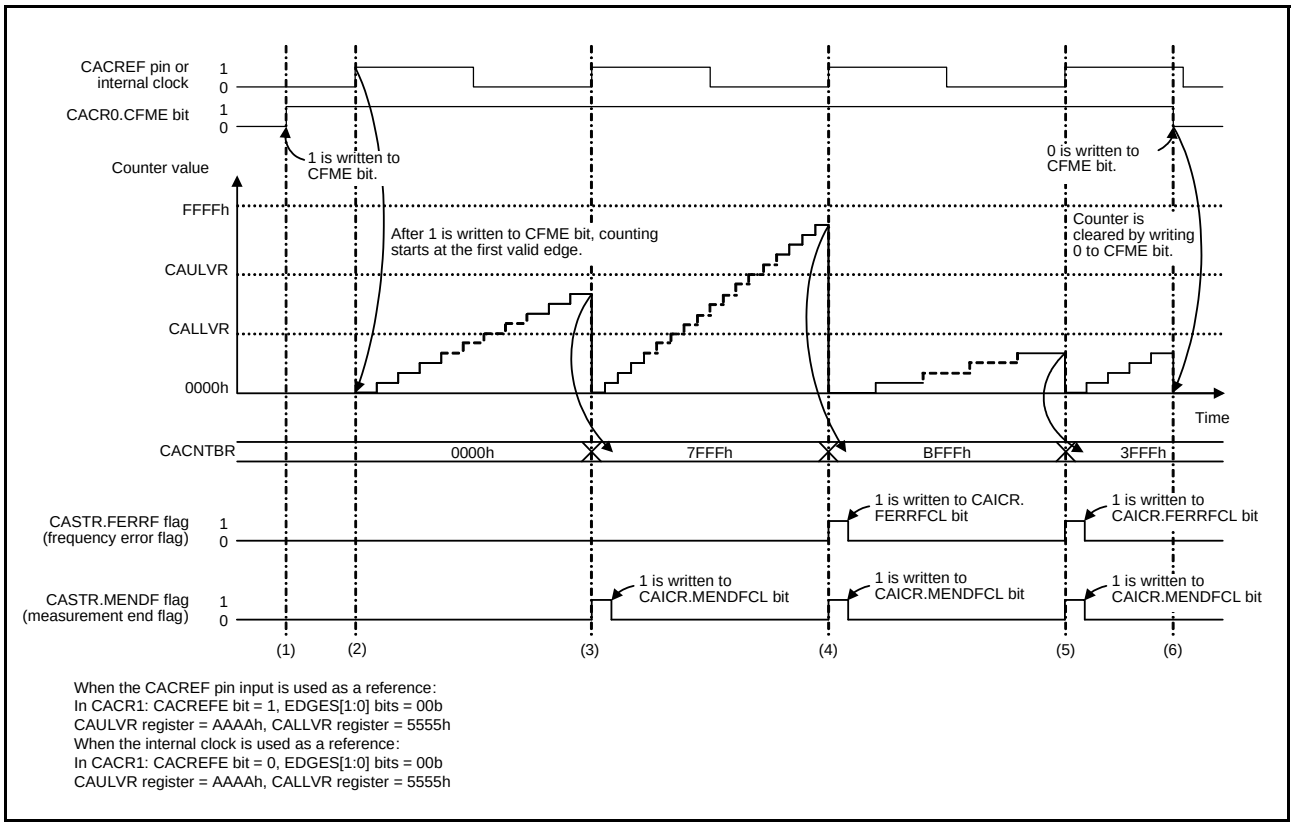


Figure 10.2 Operating Example of Clock Frequency Accuracy Measurement Circuit

(1) When the CACREF pin input is used as a reference (the CACR1.CACREFE bit = 1), clock frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is 0 and the CACR1.CACREFE bit is 1. On the other hand, when the internal clock is used as a reference (the CACR1.CACREFE bit = 0), clock frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is 1.

(2) When the CACREF pin input is used as a reference, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits is input from the CACREF pin after 1 is written to the CFME bit. The valid edge is a rising edge (the CACR1.EDGES[1:0] bits = 00b) in Figure 10.2.
When the internal clock is used as a reference, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits is input based on the clock source selected by the CACR2.RSCS[2:0] bits after 1 is written to the CFME bit. The valid edge is a rising edge (the CACR1.EDGES[1:0] bits = 00b) in Figure 10.2.

(3) When the next valid edge is input, the counter value is transferred in the CACNTBR register and compared with the values of the CAULVR and CALLVR registers. If the formula CALLVR ≤ CACNTBR ≤ CAULVR is satisfied, only the CASTR.MENDF flag is set to 1 because the clock frequency is correct. If the CAICR.MENDIE bit is 1, a measurement end interrupt is generated.

(4) When the next valid edge is input, the counter value is transferred in the CACNTBR register and compared with the values of the CAULVR and CALLVR registers. In the case of CACNTBR > CAULVR, the CASTR.FERRF flag is set to 1 because the clock frequency is erroneous. If the CAICR.FERRIE bit is 1, a frequency error interrupt is

generated. Also, the CASTR.MENDF flag is set to 1. If the CAICR.MENDIE bit is 1, a measurement end interrupt is generated.

(5) When the next valid edge is input, the counter value is transferred in the CACNTBR register and compared with the values of the CAULVR and CALLVR registers. In the case of CACNTBR < CALLVR, the CASTR.FERRF flag is set to 1 because the clock frequency is erroneous. If the CAICR.FERRIE bit is 1, a frequency error interrupt is generated. Also, the CASTR.MENDF flag is set to 1. If the CAICR.MENDIE bit is 1, a measurement end interrupt is generated.

(6) While the CACR0.CFME bit is 1, the counter value is transferred in the CACNTBR register and compared with the values of the CAULVR and CALLVR registers every time a valid edge is input. Writing 0 to the CACR0.CFME bit clears the counter and stops up-counting.

10.3.2 Digital Filtering of Signals on the CACREF Pin

The CACREF pin has a digital filter. Levels on the target pin for sampling are conveyed to the internal circuitry after matching three consecutive times at the selected sampling interval and the same level continues to be conveyed internally until the level on the pin again matches three consecutive times.

Enabling and disabling of the digital filter and its sampling clock are selectable.

The counter value transferred in the CACNTBR register may be in error by up to one cycle of the sampling clock due to the difference between the phases of the digital filter and the signal input to the CACREF pin.

When a frequency dividing clock is selected as a count source clock, the counter value error is obtained by the following formula:

Counter value error = (One cycle of the count source clock) / (One cycle of the sampling clock)

10.4 Interrupt Requests

The CAC generates three types of interrupt request: frequency error interrupt, measurement end interrupt, and overflow interrupt. When an interrupt source is generated, the corresponding status flag becomes 1. Table 10.3 lists details on the interrupt requests of the clock frequency accuracy measurement circuit.

Table 10.3 Interrupt Requests of Clock Frequency Accuracy Measurement Circuit

Interrupt Request	Interrupt Enable Bit	Status Flag	Interrupt Source
Frequency error interrupt	CAICR.FERRIE	CASTR.FERRF	The result of comparing CACNTBR to CAULVR and CALLVR is either CACNTBR > CAULVR or CACNTBR < CALLVR.
Measurement end interrupt	CAICR.MENDIE	CASTR.MENDF	A valid edge is input from the CACREF pin. Note however that a measurement end interrupt does not occur at the first valid edge after writing 1 to the CACR0.CFME bit.
Overflow interrupt	CAICR.OVFIE	CASTR.OVFF	The counter has overflowed.

10.5 Usage Notes

10.5.1 Module Stop Function Setting

CAC operation can be disabled or enabled using module stop control register C (MSTPCRC). The initial setting is for the CAC to be halted. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

11. Low Power Consumption

11.1 Overview

This MCU has several functions for reducing power consumption, including switching of clock signals to reduce power consumption, BCLK output control, SDCLK output control, stopping modules, functions for low power consumption in normal operation, and transitions to low power consumption states.

Table 11.1 lists the specifications of low power consumption functions, and Table 11.2 lists the conditions to shift to low power consumption modes, states of the CPU and peripheral modules, and the method for release from each mode. After a reset, this MCU enters the normal program execution state, but modules except for the DMAC, DTC, and RAM do not operate.

Table 11.1 Specifications of Low Power Consumption Functions

Item	Specification
Reducing power consumption by switching clock signals	The frequency division ratio is settable independently for the system clock (ICLK), peripheral module clock (PCLKA, PCLKB, PCLKC, PCLKD), external bus clock (BCLK), and flash interface clock (FCLK).*1
BCLK output control function	BCLK output or high-level output can be selected.*1
SDCLK output control function	SDCLK output or high-level output can be selected.*1
Module-stop function	Functions can be stopped independently for each peripheral module.
Function for transition to low power consumption mode	Transition to a low power consumption mode in which the CPU, peripheral modules, or oscillators are stopped is enabled.
Low power consumption modes	• Sleep mode • All-module clock stop mode • Software standby mode • Deep software standby mode
Function for lower operating power consumption	• Power consumption can be reduced in normal operation, sleep mode, and all-module clock stop mode by selecting an appropriate operating power consumption control mode according to the operating frequency and operating voltage range. • Three operating power control modes High-speed operating mode Low-speed operating mode 1 Low-speed operating mode 2 There is no difference in power consumption when the same conditions (frequency and voltage) are set in low-speed operating modes 1 and 2.

Note 1. For details, refer to section 9, Clock Generation Circuit.

Table 11.2 Entering and Exiting Low Power Consumption Modes and Operating States in Each Mode

Entering and Exiting Low Power Consumption Modes and Operating States	Sleep Mode	All-Module Clock Stop Mode	Software Standby Mode	Deep Software Standby Mode
Transition condition	Control register + instruction	Control register + instruction	Control register + instruction	Control register + instruction
Method of release other than reset	Interrupt	Interrupt*1	Interrupt*2	Interrupt*3
State after release*4	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (reset processing)
Main clock oscillator	Operating possible	Operating possible	Operating possible*5	Operating possible*5
Sub-clock oscillator	Operating possible	Operating possible	Operating possible*6	Operating possible*6
High-speed on-chip oscillator	Operating possible	Operating possible	Stopped	Stopped
Low-speed on-chip oscillator	Operating possible	Operating possible	Stopped	Stopped
IWDT-dedicated on-chip oscillator	Operating possible*7	Operating possible*7	Operating possible*7	Stopped (Undefined)*7
PLL	Operating possible	Operating possible	Stopped	Stopped
CPU	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Undefined)
RAM	Operating possible (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Undefined)
Standby RAM	Operating possible (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Retained/Undefined)*8
Flash memory	Operating	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)
USBFS host/function module (USB0, USB1)	Operating possible	Stopped*9	Stopped*9	Stopped (Retained*10/Undefined)
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)
Port Output Enable (POE)	Operating possible	Operating possible*12	Stopped (Retained)	Stopped (Undefined)
Remote control signal receiver (REMC)	Operating possible	Stopped*13	Operating possible	Operating possible
Voltage detection circuit (LVD)	Operating possible	Operating possible	Operating possible	Operating possible *14, *15
Power-on reset circuit	Operating	Operating	Operating	Operating*15
Other peripheral modules	Operating possible	Stopped (Retained)	Stopped (Retained)	Stopped (Undefined)
I/O ports	Operating	Retained*16	Retained*17	Retained*17

“Operating possible” means that operating or stopped can be controlled by the control register setting.
“Stopped (Retained)” means that internal register values are retained and internal operations are suspended.
“Stopped (Undefined)” means that internal register values are undefined and power is not supplied to the internal circuit.

Note 1. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ15) or any of peripheral interrupts (the 8-bit timer, RTC alarm, RTC periodic, IWDT, USB0 suspend/resume, REMC reception, VBATT tampering detection, voltage monitoring 1, voltage monitoring 2, and main-clock oscillation stop detection).
Note 2. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ15) or any of peripheral interrupts (the RTC alarm, RTC periodic, IWDT, USB0 suspend/resume, REMC reception, VBATT tampering detection, voltage monitoring 1, and voltage monitoring 2 interrupts).
Note 3. “Interrupts” here indicates those of the pin functions that are sources for the generation of external pin interrupts (the NMI, IRQ0-DS to IRQ15-DS, SCL2-DS, SDA2-DS, or CRX1-DS) or any of peripheral interrupts (the RTC alarm, RTC periodic, USB0 suspend/resume, REMC reception, VBATT tampering detection, 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. When the pin functions have “-DS” appended to their names, they can also be used as triggers for release from deep software standby. Also, USB0 is not released from deep software standby mode using USB0_OVRCURB assigned to pin P22.
Note 4. This does not include release initiated by the RES# pin reset, power-on reset, voltage monitoring reset, or independent watchdog-timer reset. The transition is to the reset state when release is initiated by one of these reset sources.
Note 5. Operation or stopping can be selected by the main clock oscillator forced oscillation bit (MOFXIN) in the main clock oscillator forced oscillation control register (MOFCR).
Note 6. Operation or stopping is selected by the sub-clock oscillator control bit (RTCEN) in the RTC control register 3 (RCR3).
Note 7. Operation or stopping is selected by the setting of the IWDT sleep mode count stop control bit (IWDTSLCSTP) in the option function select register 0 (OFS0) in IWDT auto start mode. If the OFS0.IWDTSLCSTP bit is 0 (disabling stopping of the counter when a transition to low power consumption mode is made), the transition is to software standby mode rather than deep

software standby mode. In any mode other than IWDT auto start mode, operation or stopping is selected by the setting of the sleep mode counter stop control bit (SLCSTP) in the IWDT counter stop control register (IWDTCSTPR). If the IWDTCSTPR.SLCSTP bit is 0 (disabling stopping of the counter when a transition to low power consumption mode is made), the transition is to software standby mode rather than deep software standby mode.

Note 8. Retention or undefined is selectable by the setting of the deep cut bits in the deep standby control register (DPSBYCR.DEEPCUT[1:0]).

Note 9. The detection of resumption by USB0 is possible.

Note 10. Disabling or enabling of detection of resumption by USB0 is controllable by the deep cut bits in the deep standby control register (DPSBYCR.DEEPCUT[1:0]). When detection of resumption is enabled, the values of the registers in the USB0 resume detecting unit are only retained even in deep software standby mode. Note that USB0 is not released from deep software standby mode by using USB0_OVRCURB assigned to pin P22.

Note 11. Stopping or operation is controlled by the module-stop setting bits (MSTPA4 and MSTPA5, respectively) in module-stop control register A (MSTPCRA) for 8-bit timers 0 and 1 (unit 0) and 2 and 3 (unit 1).

Note 12. When a source condition for POE interrupts is satisfied while POE interrupts are enabled and the chip is in all-module clock stop mode, the flag for the source condition is retained but return from all-module clock stop mode does not proceed. If a different source initiates return from all-module clock stop mode in this situation, the POE interrupt is generated after that.

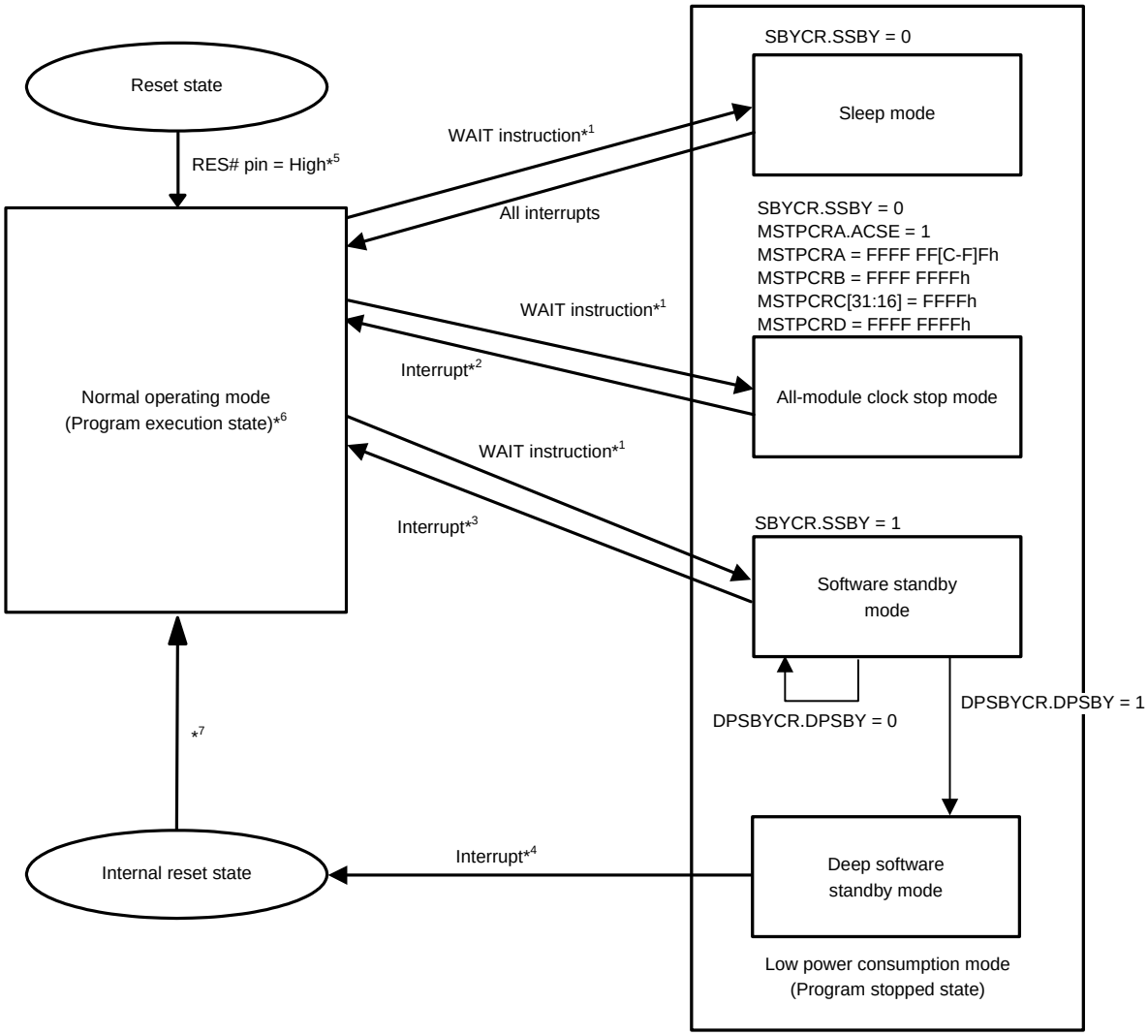
Note 13. Operation of the REMC is possible when the operating clock is the sub-clock or the TMR compare match output. Note that the TMR compare match output can only be used when the setting of MSTPCRA.MSTPA5 is 0.

Note 14. 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 15. When the deep cut bits in the deep standby control register (DPSBYCR.DEEPCUT[1:0]) are set to 11b and the MCU enters deep software standby mode, the voltage detection circuit stops and the low power consumption function of the power-on reset circuit is enabled.

Note 16. 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 17. 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#, WR1#, WR#, BC0#, BC1#, ALE, CKE, SDCS#, RAS#, CAS#, WE#, and DQM0, DQM1) by the output port enable bit (OPE) in the standby control register (SBYCR).



Note 1. When an interrupt that acts as a trigger for release is received during the transition to the program-stopped state after the execution of a WAIT instruction, the transition to the program stopped state does not proceed and interrupt exception processing is executed instead.

Note 2. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ15) or any of peripheral interrupts (the 8-bit timer, RTC alarm, RTC periodic, IWDT, USB0 suspend/resume, voltage-monitoring 1, voltage-monitoring 2, and oscillator-stopped detection interrupts). However, a, 8-bit timer interrupt is only effective if the value of the corresponding module-stop setting bit (MSTPA4 or MSTPA5, respectively) in module-stop control register A (MSTPCRA) for 8-bit timer 0 and 1 (unit 0) or 2 and 3 (unit 1) is 0.

Note 3. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ15) or any of peripheral interrupts (the RTC alarm, RTC periodic, IWDT, USB0 suspend/resume, REMC reception, VBATT tampering detection, voltage-monitoring 1, and voltage-monitoring 2 interrupts).

Note 4. “Interrupts” here indicates those of the pin functions that are sources for the generation of external pin interrupts (the NMI, IRQ0-DS to IRQ15-DS, SCL2-DS, SDA2-DS, or CRX1-DS) or any of peripheral interrupts (the RTC alarm, RTC periodic, USB0 suspend/resume, REMC reception, VBATT tampering detection, 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. Also, USB0 is not released from deep software standby mode using USB0_OVRCURB assigned to pin P22.

Note 5. The LOCO is the source of the operating clock following a transition from the reset state to normal operating mode.

Note 6. The transition to normal operating mode is made due to an interrupt from sleep mode, all-module clock-stop mode, or software standby mode. In the case of recovery from sleep mode, the clock source for subsequent use is selectable. For details, refer to section 11.2.7, Sleep Mode Return Clock Source Switching Register (RSTCKCR). Also in the case of recovery from software standby mode, the clock source will be the same before and at the time of the transition.

Note 7. When an interrupt source as indicated in Note 4 above is generated, an internal reset (deep software standby reset) is generated over a fixed period. Release from deep software standby mode accompanies release from the internal reset state. The transition from deep software standby mode is to normal operating mode, and reset exception processing starts with the LOCO as the source of the operating clock.

When the RES# pin reset, power-on reset, voltage-monitoring 0 reset, voltage-monitoring 1 reset, voltage-monitoring 2 reset, watchdog timer reset, independent watchdog timer reset, or software reset is generated in any state, a transition to the reset state is made. The voltage-monitoring 1 reset and voltage-monitoring 2 reset cannot be used for release from deep software standby mode.

Figure 11.1 Mode Transitions

11.2 Register Descriptions

11.2.1 Standby Control Register (SBYCR)

Address(es): 0008 000Ch

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

Bit	Symbol	Bit Name	Description	R/W
b13 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	OPE	Output Port Enable	0: In software standby mode or deep software standby mode, the address bus and bus control signals are set to the high-impedance state. 1: In software standby mode or deep software standby mode, the address bus and bus control signals retain the output state.	R/W
b15	SSBY	Software Standby	0: Shifts to sleep mode or all-module clock stop mode after the WAIT instruction is executed 1: Shifts to software standby mode after the WAIT instruction is executed	R/W

OPE Bit (Output Port Enable)

The OPE bit specifies whether to retain the output of the address bus and bus control signals (CS0# to CS7#, RD#, WR0#, WR1#, WR#, BC0#, BC1#, ALE, CKE, SDCS#, RAS#, CAS#, WE#, and DQM0, DQM1) 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 MCU enters software standby mode after execution of the WAIT instruction. When the MCU returns to normal operating mode after an interrupt has initiated release from software standby mode, the SSBY bit remains 1. Write 0 to this bit to clear it.

When the oscillation stop detection function enable bit in the oscillation stop detection control register (OSTDCR.OSTDE) is 1, the setting of the SSBY bit is ineffective. Even if the SSBY bit is 1, the MCU will enter sleep mode or all module clock stop mode on execution of the WAIT instruction.

When the code flash memory P/E mode entry bit in the flash P/E mode entry register (FENTRYR.FENTRYC) is 1 or the data flash memory P/E mode entry bit (FENTRYR.FENTRYD) is 1, the setting of this bit is ineffective. Sleep mode is entered on execution of the WAIT instruction even if this bit has been set to 1.

11.2.2 Module Stop Control Register A (MSTPCRA)

Address(es): 0008 0010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ACSE	—	MSTPA29	MSTPA28	MSTPA27	—	—	MSTPA24	—	—	—	—	—	—	MSTPA17	MSTPA16
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	—	MSTPA11	MSTPA10	MSTPA9	—	—	—	MSTPA5	MSTPA4	—	—	MSTPA1	MSTPA0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	MSTPA0	Compare Match Timer W (Unit 1) Module Stop	Target module: CMTW1 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b1	MSTPA1	Compare Match Timer W (Unit 0) Module Stop	Target module: CMTW0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b3, b2	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b4	MSTPA4	8-Bit Timer 3/2 (Unit 1) Module Stop	Target module: TMR3/TMR2 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b5	MSTPA5	8-Bit Timer 1/0 (Unit 0) Module Stop	Target module: TMR1/TMR0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b8 to b6	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b9	MSTPA9	Multifunction Timer Pulse Unit 3 Module Stop	Target module: MTU 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b10	MSTPA10	Programmable Pulse Generator (Unit 1) Module Stop	Target module: PPG1 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b11	MSTPA11	Programmable Pulse Generator (Unit 0) Module Stop	Target module: PPG0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b12	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b13	MSTPA13	16-Bit Timer Pulse Unit 0 (Unit 0) Module Stop	Target module: TPU unit 0 (TPU0 to TPU5) 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b14	MSTPA14	Compare Match Timer (Unit 1) Module Stop	Target module: CMT unit 1 (CMT2, CMT3) 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b15	MSTPA15	Compare Match Timer (Unit 0) Module Stop	Target module: CMT unit 0 (CMT0, CMT1) 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b16	MSTPA16	12-bit A/D Converter (Unit 1) Module Stop	Target module: S12AD1 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b17	MSTPA17	12-bit A/D Converter (Unit 0) Module Stop	Target module: S12AD 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b23 to b18	—	Reserved	These bits are read as 1. The write value should be 1.	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: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b29	MSTPA29	EXDMA Controller Module Stop	Target module: EXDMAC 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b30	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b31	ACSE	All-Module Clock Stop Mode Enable	0: All-module clock stop mode is disabled 1: All-module clock stop mode is enabled	R/W

ACSE Bit (All-Module Clock Stop Mode Enable)

The ACSE bit enables or disables a transition to all-module clock stop mode.

With the ACSE bit set to 1, when the CPU executes the WAIT instruction with the SBYCR.SSBY bit, MSTPCRA, MSTPCRB, MSTPCRC, and MSTPCRD satisfying specified conditions, the MCU enters all-module clock stop mode. For details, refer to section 11.6.2, All-Module Clock Stop Mode.

Whether to stop the 8-bit timers or not can be selected by the MSTPA5 and MSTPA4 bits.

When the MSTPCRA.ACSE bit = 0 while the SBYCR.SSBY = 0, a transition to sleep mode is made after the WAIT instruction is executed.

When the code flash P/E mode entry bit in the flash memory P/E mode entry register (FENTRYR.FENTRYC) is 1 or the data flash memory P/E mode entry bit (FENTRYR.FENTRYD) is 1, the setting of this bit is ineffective. Sleep mode is entered on execution of the WAIT instruction if this bit has been set to 1.

11.2.3 Module Stop Control Register B (MSTPCRB)

Address(es): 0008 0014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MSTPB31	MSTPB30	MSTPB29	MSTPB28	MSTPB27	MSTPB26	MSTPB25	MSTPB24	MSTPB23	—	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
	—	—	—	—	—	—	MSTPB9	MSTPB8	—	MSTPB6	—	MSTPB4	—	—	MSTPB1	MSTPB0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	MSTPB0	CAN Module 0 Module Stop*1	Target module: CAN0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b1	MSTPB1	CAN Module 1 Module Stop*1	Target module: CAN1 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b3, b2	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b4	MSTPB4	Serial Communication Interface SCIh Module Stop	Target module: SCI12 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b5	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b6	MSTPB6	Data Operation Circuit Module Stop	Target module: DOC 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b7	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b8	MSTPB8	Temperature Sensor Module Stop	Target module: Temperature sensor 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b9	MSTPB9	Event Link Controller Module Stop	Target module: ELC 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b15 to b10	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b16	MSTPB16	Serial Peripheral Interface 1 Module Stop	Target module: RSPI1 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b17	MSTPB17	Serial Peripheral Interface 0 Module Stop	Target module: RSPI0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b18	MSTPB18	USB 2.0 FS Interface 1 Module Stop*2	Target module: USB1 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b19	MSTPB19	USB 2.0 FS Interface 0 Module Stop*2	Target module: USB0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b20	MSTPB20	I^2C Bus Interface 1 Module Stop	Target module: RIIC1 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b21	MSTPB21	I^2C Bus Interface 0 Module Stop	Target module: RIIC0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b22	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

Bit	Symbol	Bit Name	Description	R/W
b23	MSTPB23	CRC Calculator Module Stop	Target module: CRC 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b24	MSTPB24	Serial Communication Interface 7 Module Stop	Target module: SCI7 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b25	MSTPB25	Serial Communication Interface 6 Module Stop	Target module: SCI6 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b26	MSTPB26	Serial Communication Interface 5 Module Stop	Target module: SCI5 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b27	MSTPB27	Serial Communication Interface 4 Module Stop	Target module: SCI4 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b28	MSTPB28	Serial Communication Interface 3 Module Stop	Target module: SCI3 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b29	MSTPB29	Serial Communication Interface 2 Module Stop	Target module: SCI2 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b30	MSTPB30	Serial Communication Interface 1 Module Stop	Target module: SCI1 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b31	MSTPB31	Serial Communication Interface 0 Module Stop	Target module: SCI0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W

Note 1. The MSTPBi bit should be rewritten while the oscillation of the clock controlled by the MSTPBi bit is stabilized. For entering software standby mode after writing a new value to the MSTPBi bit, wait for two cycles of the CAN clock (CANMCLK) to elapse after writing the new value, and then execute a WAIT instruction (i = 0, 1).

Note 2. For entering software standby mode after writing a new value to the MSTPB19 bit or MSTPB18 bit, wait for two cycles of the USB clock (UCLK) to elapse after writing the new value, and then execute a 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	—	MSTPC17	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	MSTPC7	—	—	—	—	—	—	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	RAM Module Stop*1	Target module: RAM (0000 0000h to 0005 FFFFh) 0: RAM operating 1: RAM stopped	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	MSTPC7	Standby RAM Module Stop*2	Target module: Standby RAM 0: Standby RAM operating 1: Standby RAM stopped	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b17	MSTPC17	I^2C Bus Interface 2 Module Stop	Target module: RIIC2 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b18	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b19	MSTPC19 *3	CAC Module Stop	Target module: CAC 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b21, b20	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b22	MSTPC22	Serial Peripheral Interface 2 Module Stop	Target module: RSPI2 0: Release from the module-stop state 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 Communications Interface 11 Module Stop	Target module: SCI11 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b25	MSTPC25	Serial Communications Interface 10 Module Stop	Target module: SCI10 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b26	MSTPC26	Serial Communications Interface 9 Module Stop	Target module: SCI9 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b27	MSTPC27	Serial Communications Interface 8 Module Stop	Target module: SCI8 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b31 to b28	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. The MSTPC0 bit should not be set to 1 during access to the RAM. The RAM should not be accessed while the MSTPC0 bit is set to 1.

Note 2. The MSTPC7 bit should not be set to 1 during access to the standby RAM. The standby RAM should not be accessed while the MSTPC7 bit is set to 1.

Note 3. The MSTPC19 bit should be rewritten while the oscillation of the clock to be controlled by this bit is stable. For entering software standby mode after writing a new value to this bit, wait for two cycles of the slowest clock among the clocks output by the oscillators actually oscillating, and then execute a WAIT instruction.

11.2.5 Module Stop Control Register D (MSTPCRD)

Address(es): 0008 001Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	MSTPD27	MSTPD26	—	—	—	—	—	—	MSTPD19	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MSTPD15	—	—	MSTPD12	—	—	—	—	MSTPD7	—	MSTPD5	—	MSTPD3	MSTPD2	MSTPD1	—
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	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b1	MSTPD1	Quad-SPI memory Interface Module Stop	Target module: QSPIX 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b2	MSTPD2	Serial Communications Interface 11 Module Stop	Target module: RSCI11 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b3	MSTPD3	Serial Communications Interface 10 Module Stop	Target module: RSCI10 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b4	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b5	MSTPD5	High-Speed I^2C bus interfaces Module Stop	Target module: RIICHS 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b6	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7	MSTPD7	Remote control signal receiver Module Stop	Target module: REMC 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b11 to b8	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b12	MSTPD12	Capacitive Touch Sensing Unit Module Stop	Target module: CTSU 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b14, b13	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b15	MSTPD15	Serial Sound Interface Module Stop	Target module: SSIE0 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b18 to b16	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b19	MSTPD19	SD Host Interface Module Stop	Target module: SDHI 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b25 to b20	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b26	MSTPD26	Serial Peripheral Interface Module Stop	Target module: RSPIA 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b27	MSTPD27	Trusted Secure IP Module Stop	Target module: Trusted Secure IP 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
b31 to b28	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

11.2.6 Operating Power Control Register (OPCCR)

Address(es): 0008 00A0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	OPCMTSF	—	OPCM[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

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

Register OPCCR is used to reduce power consumption in normal operating mode, sleep mode, and all-module clock stop mode. Power consumption can be reduced according to the operating frequency and operating voltage to be used by the OPCCR setting.

There is no difference in power consumption between low speed mode 1 and 2 when the same conditions (frequency, voltage) are set in both modes.

OPCCR should not be modified in the following cases:

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

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

On return from software standby, the chip enters the high-speed operating mode. Even if a WAIT instruction is executed, if release from software standby precedes completion of the transition, the mode remains the same as before execution of the WAIT instruction. If this creates a problem, set the OPCCR.OPCM[2:0] bits to 000b during processing of the return interrupt.

OPCM[2:0] Bits (Operating Power Control Mode Select)

The OPCM[2:0] bits select operating power control mode in normal operating mode, sleep mode, and all-module clock stop mode.

Table 11.3 shows the operating power control modes along with the operating frequency ranges, operating voltage ranges, and power consumption.

Table 11.3 Relationship between Operating Power Control Mode, Operating Range, and Power Consumption

Operating Power Control Mode	OPCM [2:0] Bits	Operating Frequency Range							Operating Voltage Range		Power Consumption
		Reading Flash Memory						Programming or Erasing Flash Memory	Reading Flash Memory	Programming or Erasing Flash Memory	
		ICLK	FCLK	PCLKA	PCLKB	PCLKC PCLKD	BCLK	FCLK			
High-speed operating mode	000b	120 MHz max	60 MHz max	120 MHz max	60 MHz max	60 MHz max*1	120 MHz max	4 MHz to 60 MHz	2.7 V to 3.6 V	2.7 V to 3.6 V	High
Low-speed operating mode 1*2	110b	1 MHz max	1 MHz max	1 MHz max	1 MHz max	1 MHz max*1	1 MHz max	P/E disabled	2.7 V to 3.6 V	P/E disabled	
Low-speed operating mode 2*2	111b	32 kHz to 264 kHz	32 kHz to 264 kHz	264 kHz max	264 kHz max	264 kHz max*1	264 kHz max	P/E disabled	2.7 V to 3.6 V	P/E disabled	Low

Note 1. When the 12-bit A/D converter is used in high-speed operating mode or low-speed operating mode 1, the frequency must be set to at least 1 MHz. The 12-bit A/D converter cannot be used in low-speed operating mode 2.
Note 2. There is no difference in power consumption between low speed mode 1 and 2 when the same conditions (frequency, voltage) are set in both modes.

Each operating power control mode is described below.

- High-speed operating mode

This mode allows high-speed operation.
During reading the flash memory (FLASH), the maximum operating frequency of ICLK, PCLKA, and BCLK is 120 MHz, and that of FCLK and PCLKB is 60 MHz. The S12AD conversion clocks PCLKC and PCLKD can be operated in the operating frequency from 1 MHz to 60 MHz. During Flash memory programming/erasure (P/E), the FCLK can be operated in the operating frequency from 4 MHz to 60 MHz. The operating voltage is in the range of 2.7 to 3.6 V both for Flash memory read and P/E. When the frequency of ICLK is set to faster than 60 MHz, the value of the ROMWT register needs to be modified. After release from a reset, this MCU is activated in this mode.

- Low-speed operating mode 1

This mode reduces power consumption for low-speed operation.
During reading the flash memory (FLASH), the maximum operating frequency of ICLK, FCLK, PCLKA, PCLKB and BCLK is 1 MHz. The operating voltage is in the range of 2.7 to 3.6 V.
In low-speed operating mode 1, P/E operation of Flash memory is disabled, and writing to set the PLLCR2.PLLEN bit to 0 (PLL operation) is prohibited.
In this mode, lower power consumption is possible than in high-speed operating mode when the same operation is performed under the same conditions (operating frequency, operating voltage).

- Low-speed operating mode 2

As compare to low-speed operating mode 1, this mode reduces power consumption for low-speed operation.
During reading the flash memory (FLASH), the maximum operating frequency of ICLK, FCLK, PCLKA, PCLKB and BCLK is 264 kHz, and the minimum operating frequency of ICLK and FCLK is 32 kHz. The operating voltage is in the range of 2.7 to 3.6 V.
The following restrictions apply when low-speed operating mode 2 is selected:

- P/E operations for flash memory are prohibited.
- Using the PLL or HOCO is prohibited.

- Using the oscillation stop detection function of the main clock oscillator is prohibited.

When the clock source select bits in the system clock control register 3 (SCKCR3.CKSEL[2:0]) are 011b (sub-clock oscillator selected) and the system clock select bits (ICK[3:0]) or Flash-IF clock select bits (FCK[3:0]) in the system clock control register (SCKCR) are not 0000b (no frequency dividing), OPCM[2:0] bits cannot be set to 111b.
When the PLL stop control bit (PLLCR2.PLLEN) in PLL control register 2 is 0 (PLL operation), writing 110b (low-speed operating mode 1) and 111b (low-speed operating mode 2) to the OPCM[2:0] bits is not possible.

OPCMTSF Flag (Operating Power Control Mode Transition Status Flag)

The OPCMTSF flag indicates the switching control state when the operating power control mode is switched.
When a write access is attempted to change the operating power control mode, the OPCMTSF flag is set to 1. The flag becomes 0 after a transition to the changed control mode is completed. Make sure that the OPCMTSF flag is 0 (completed operating power control mode transition) before the next processing.

11.2.7 Sleep Mode Return Clock Source Switching Register (RSTCKCR)

Address(es): 0008 00A1h

	b7	b6	b5	b4	b3	b2	b1	b0
	RSTCKEN	—	—	—	—	RSTCKSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	RSTCKSEL [2:0]	Sleep Mode Return Clock Source Select	b2 b0 0 0 1: HOCO is selected 0 1 0: Main clock oscillator is selected Settings other than above are prohibited while the RSTCKEN bit is 1.	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	RSTCKEN	Sleep Mode Return Clock Source Switching Enable	0: Clock source switching on release from sleep mode is disabled 1: Clock source switching on release from sleep mode is enabled	R/W

Register RSTCKCR is used to control clock source switching at the time of release from sleep mode.
When operation is restored from sleep mode by setting RSTCKCR, the main clock oscillator stop bit in the main clock oscillator control register (MOSCCR.MOSTP) and HOCO stop bit in the high-speed on-chip oscillator control register (HOCOCR.HCSTP) corresponding to the clock source to be used on restoration are automatically modified to the operating state. The value of RSTCKSEL[2:0] bits is automatically reloaded to the clock source select bits in the system clock control register 3 (SCKCR3.CKSEL[2:0]).
The sleep mode return clock source switching function and clock source switching function by the ELC cannot be used at the same time. To enable the sleep mode return clock source switching function, write 1 to the RSTCKCR.RSTCKEN bit with the ELC clock source switching function disabled. The ELC clock source switching function should be enabled with the RSTCKCR.RSTCKEN bit being 0.
When the setting of register RSTCKCR is for the HOCO to be used in recovery from sleep mode, the power supply for the HOCO is not automatically switched on. If the HOCO to be used in recovery from sleep mode, the power supply for the HOCO must be on when the transition to sleep mode takes place.
When return from sleep mode is made while clock source switching on release from sleep mode is enabled (RSTCKCR.RSTCKEN is 1), and operating power control mode select bits (OPCCR.OPCM[2:0]) are set so as to select low-speed operating mode 1 (110b) or low-speed operating mode 2 (111b), the OPCCR.OPCM[2:0] bits are automatically switched to high-speed mode (000b).

RSTCKSEL[2:0] Bits (Sleep Mode Return Clock Source Select)

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

RSTCKEN Bit (Sleep Mode Return Clock Source Switching Enable)

The RSTCKEN bit enables or disables clock source switching at the time of release from sleep mode.
On release from sleep mode, the clock source should be switched only when LOCO or sub clock is selected as a clock for a transition to sleep mode. To make a transition to sleep mode with HOCO, main clock, or PLL selected as the clock source, the RSTCKEN bit should not be set to 1.

11.2.8 Deep Standby Control Register (DPSBYCR)

Address(es): 0008 C280h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	DEEPCUT [1:0]	Deep Cut	b1 b0 0 0: Power is supplied to the standby RAM and USB0 resume detecting unit and REMC in deep software standby mode. 0 1: Power is not supplied to the standby RAM and USB0 resume detecting unit and REMC in deep software standby mode. 1 0: Setting prohibited 1 1: Power is not supplied to the standby RAM and USB0 resume detecting unit and REMC in deep software standby mode. In addition, LVD is stopped and the low power consumption function in a power-on reset circuit is enabled.	R/W
b5 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	IOKEEP	I/O Port Retention	0: Release from deep software standby mode and cancellation of I/O port retention proceed simultaneously. 1: The I/O port state is retained even after release from deep software standby mode. Then, writing 0 to the IOKEEP bit cancels the I/O port retention.	R/W
b7	DPSBY	Deep Software Standby	SSBY b7 0 0: Transition to sleep mode or all-module clock stop mode is made after the WAIT instruction is executed. 0 1: Transition to sleep mode or all-module clock stop mode is made after the WAIT instruction is executed. 1 0: Transition to software standby mode is made after the WAIT instruction is executed. 1 1: Transition to deep software standby mode is made after the WAIT instruction is executed.	R/W

Register DPSBYCR is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, refer to 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 supply of power to the standby RAM, USB0 resume detecting unit and REMC in deep software standby mode. In addition, these bits control the state of LVD and power-on reset circuit in deep software standby mode.

The internal supply of power to the standby RAM, USB0 resume detecting unit and REMC is controllable by the DEEPCUT[1:0] bits.

When a USB0 suspend/resume interrupt*1 or a REMC reception interrupt is used as a source to release from deep software standby, set the DEEPCUT[1:0] bits to 00b.

When the LVD is used in deep software standby mode, set the DEEPCUT[1:0] bits to 00b or 01b.

For lower power consumption, set the DEEPCUT[1:0] bits to 11b so that the LVD is stopped and the low power consumption function of the power-on reset circuit is enabled.

Note 1. USB0 is not released from deep software standby mode using USB0_OVRCURB assigned to pin P22.

IOKEEP Bit (I/O Port Retention)

In deep software standby mode, I/O ports keep retaining the same states from software standby mode. The IOKEEP bit specifies whether to keep retaining the I/O port states from deep software standby mode even after release from deep software standby mode, or to cancel retention of the I/O port states.

DPSBY Bit (Deep Software Standby)

The DPSBY bit controls transitions to deep software standby mode.
When the WAIT instruction is executed while the SBYCR.SSBY and DPSBY bits are both 1, the MCU enters deep software standby mode through software standby mode.
The DPSBY bit remains 1 when release from deep software standby mode is triggered by certain pins which are sources of external pin interrupts (NMI, IRQ0-DS to IRQ15-DS, SCL2-DS, SDA2-DS, and CRX1-DS) or a peripheral interrupt (RTC alarm, RTC periodic, USB0 suspend/resume*1, REMC reception, VBATT tampering detection, voltage monitoring 1, or voltage monitoring 2). Write 0 to this bit to clear it.
The setting of the DPSBY bit becomes invalid when the IWDT is in auto-start mode and the OFS0.IWDTSLCSTP is 0 (counting continues) or the IWDT is in register start mode and the SLCSTP bit in IWDTCSTPR is 0.
Instead, even when the SBYCR.SSBY bit is 1 and the DPSBY bit 1, the transition after the execution of a WAIT instruction is to software standby mode.
The setting of the DPSBY bit becomes invalid when voltage monitoring 1 reset is enabled by the voltage monitoring 1 circuit mode select bit (LVD1CR0.LVD1RI = 1) or when a voltage monitoring 2 reset is enabled by the voltage monitoring 2 circuit mode bit (LVD2CR0.LVD2RI = 1). In this case, even when the SBYCR.SSBY bit is 1 and the DPSBY bit is 1, the transition after the execution of a WAIT instruction is to software standby mode.

Note 1. USB0 is not released from deep software standby mode using USB0_OVRCURB assigned to pin P22.

11.2.9 Deep Standby Interrupt Enable Register 0 (DPSIER0)

Address(es): 0008 C282h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ7E	DIRQ6E	DIRQ5E	DIRQ4E	DIRQ3E	DIRQ2E	DIRQ1E	DIRQ0E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ0E	IRQ0-DS Pin Enable	0: Release from deep software standby mode by the IRQ0-DS pin is disabled 1: Release from deep software standby mode by the IRQ0-DS pin is enabled	R/W
b1	DIRQ1E	IRQ1-DS Pin Enable	0: Release from deep software standby mode by the IRQ1-DS pin is disabled 1: Release from deep software standby mode by the IRQ1-DS pin is enabled	R/W
b2	DIRQ2E	IRQ2-DS Pin Enable	0: Release from deep software standby mode by the IRQ2-DS pin is disabled 1: Release from deep software standby mode by the IRQ2-DS pin is enabled	R/W
b3	DIRQ3E	IRQ3-DS Pin Enable	0: Release from deep software standby mode by the IRQ3-DS pin is disabled 1: Release from deep software standby mode by the IRQ3-DS pin is enabled	R/W
b4	DIRQ4E	IRQ4-DS Pin Enable	0: Release from deep software standby mode by the IRQ4-DS pin is disabled 1: Release from deep software standby mode by the IRQ4-DS pin is enabled	R/W
b5	DIRQ5E	IRQ5-DS Pin Enable	0: Release from deep software standby mode by the IRQ5-DS pin is disabled 1: Release from deep software standby mode by the IRQ5-DS pin is enabled	R/W
b6	DIRQ6E	IRQ6-DS Pin Enable	0: Release from deep software standby mode by the IRQ6-DS pin is disabled 1: Release from deep software standby mode by the IRQ6-DS pin is enabled	R/W
b7	DIRQ7E	IRQ7-DS Pin Enable	0: Release from deep software standby mode by the IRQ7-DS pin is disabled 1: Release from deep software standby mode by the IRQ7-DS pin is enabled	R/W

Register DPSIER0 is not initialized by the internal reset signal used as deep software standby mode releasing source. For details, refer to Table 6.2, Targets to be Initialized by Each Reset Source.

After the setting of DPSIER0 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR0 being set to 1. Therefore, DPSIFR0 should be set to 0 before a transition to deep software standby mode.

11.2.10 Deep Standby Interrupt Enable Register 1 (DPSIER1)

Address(es): 0008 C283h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ15E	DIRQ14E	DIRQ13E	DIRQ12E	DIRQ11E	DIRQ10E	DIRQ9E	DIRQ8E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ8E	IRQ8-DS Pin Enable	0: Release from deep software standby mode by the IRQ8-DS pin is disabled 1: Release from deep software standby mode by the IRQ8-DS pin is enabled	R/W
b1	DIRQ9E	IRQ9-DS Pin Enable	0: Release from deep software standby mode by the IRQ9-DS pin is disabled 1: Release from deep software standby mode by the IRQ9-DS pin is enabled	R/W
b2	DIRQ10E	IRQ10-DS Pin Enable	0: Release from deep software standby mode by the IRQ10-DS pin is disabled 1: Release from deep software standby mode by the IRQ10-DS pin is enabled	R/W
b3	DIRQ11E	IRQ11-DS Pin Enable	0: Release from deep software standby mode by the IRQ11-DS pin is disabled 1: Release from deep software standby mode by the IRQ11-DS pin is enabled	R/W
b4	DIRQ12E	IRQ12-DS Pin Enable	0: Release from deep software standby mode by the IRQ12-DS pin is disabled 1: Release from deep software standby mode by the IRQ12-DS pin is enabled	R/W
b5	DIRQ13E	IRQ13-DS Pin Enable	0: Release from deep software standby mode by the IRQ13-DS pin is disabled 1: Release from deep software standby mode by the IRQ13-DS pin is enabled	R/W
b6	DIRQ14E	IRQ14-DS Pin Enable	0: Release from deep software standby mode by the IRQ14-DS pin is disabled 1: Release from deep software standby mode by the IRQ14-DS pin is enabled	R/W
b7	DIRQ15E	IRQ15-DS Pin Enable	0: Release from deep software standby mode by the IRQ15-DS pin is disabled 1: Release from deep software standby mode by the IRQ15-DS pin is enabled	R/W

Register DPSIER1 is not initialized by the internal reset signal used as deep software standby mode releasing source. For details, refer to Table 6.2, Targets to be Initialized by Each Reset Source.

After the setting of DPSIER1 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR1 being set to 1. Therefore, DPSIFR1 should be set to 0 before a transition to deep software standby mode.

11.2.11 Deep Standby Interrupt Enable Register 2 (DPSIER2)

Address(es): 0008 C284h

	b7	b6	b5	b4	b3	b2	b1	b0
	DUSBIE	DRIICCIE	DRIICDIE	DNMIE	DRTCAIE	DRTCIIE	DLVD2IE	DLVD1IE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DLVD1IE	LVD1 Deep Standby Release Signal Enable	0: Disable release from deep software standby mode by the voltage monitoring 1 signal 1: Enable release from deep software standby mode by the voltage monitoring 1 signal	R/W
b1	DLVD2IE	LVD2 Deep Standby Release Signal Enable	0: Disable release from deep software standby mode by the voltage monitoring 2 signal 1: Enable release from deep software standby mode by the voltage monitoring 2 signal	R/W
b2	DRTCIIE	RTC Periodic Interrupt Deep Standby Release Signal Enable	0: Release from deep software standby mode by the RTC periodic interrupt signal is disabled 1: Release from deep software standby mode by the RTC periodic interrupt signal is enabled	R/W
b3	DRTCAIE	RTC Alarm Interrupt Deep Standby Release Signal Enable	0: Release from deep software standby mode by the RTC alarm interrupt signal is disabled 1: Release from deep software standby mode by the RTC alarm interrupt signal is enabled	R/W
b4	DNMIE	NMI Pin Enable	0: Release from deep software standby mode by the NMI pin is disabled 1: Release from deep software standby mode by the NMI pin is enabled	R/W*1
b5	DRIICDIE	SDA2-DS Deep Standby Release Signal Enable	0: Release from deep software standby mode by the SDA2-DS signal is disabled 1: Release from deep software standby mode by the SDA2-DS signal is enabled	R/W
b6	DRIICCIE	SCL2-DS Deep Standby Release Signal Enable	0: Release from deep software standby mode by the SCL2-DS signal is disabled 1: Release from deep software standby mode by the SCL2-DS signal is enabled	R/W
b7	DUSBIE	USB0 Suspend/Resume Deep Standby Release Signal Enable	0: Release from deep software standby mode by the USB0 suspend/resume is disabled 1: Release from deep software standby mode by the USB0 suspend/resume is enabled	R/W

Note 1. 1 can be written only once. Once 1 is written to this bit, subsequent write accesses are disabled.

Register DPSIER2 is not initialized by the internal reset signal used as deep software standby mode releasing source. For details, refer to Table 6.2, Targets to be Initialized by Each Reset Source.
After the setting of DPSIER2 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR2 being set to 1. Therefore, DPSIFR2 should be set to 0 before a transition to deep software standby mode.

DUSBIE Bit (USB0 Suspend/Resume Deep Standby Release Signal Enable)

The DUSBIE bit is an enable bit for USB0. USB0 is not released from deep software standby mode using USB0_OVRCURB assigned to pin P22.

11.2.12 Deep Standby Interrupt Enable Register 3 (DPSIER3)

Address(es): 0008 C285h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	DTADIE	DRMCIE	DCANIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DCANIE	CRX1-DS Deep Standby Release Signal Enable	0: Release from deep software standby mode by the CRX1-DS pin is disabled 1: Release from deep software standby mode by the CRX1-DS pin is enabled	R/W
b1	DRMCIE	REMC Interrupt Deep Standby Release Signal Enable	0: Disable release from deep software standby mode by the REMC interrupt 1: Enable release from deep software standby mode by the REMC interrupt	R/W
b2	DTADIE	VBATT Tamper Detection Deep Standby Release Signal Enable	0: Disable release from deep software standby mode by the VBATT tamper detection 1: Enable release from deep software standby mode by the VBATT tamper detection	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Register DPSIER3 is not initialized by the internal reset signal used as deep software standby mode releasing source. For details, refer to Table 6.2, Targets to be Initialized by Each Reset Source.

After the setting of DPSIER3 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR3 being set to 1. Therefore, DPSIFR3 should be set to 0 before a transition to deep software standby mode.

11.2.13 Deep Standby Interrupt Flag Register 0 (DPSIFR0)

Address(es): 0008 C286h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ7F	DIRQ6F	DIRQ5F	DIRQ4F	DIRQ3F	DIRQ2F	DIRQ1F	DIRQ0F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ0F	IRQ0-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ0-DS pin 1: Request for release is being generated on the IRQ0-DS pin	R(/W) *1
b1	DIRQ1F	IRQ1-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ1-DS pin 1: Request for release is being generated on the IRQ1-DS pin	R(/W) *1
b2	DIRQ2F	IRQ2-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ2-DS pin 1: Request for release is being generated on the IRQ2-DS pin	R(/W) *1
b3	DIRQ3F	IRQ3-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ3-DS pin 1: Request for release is being generated on the IRQ3-DS pin	R(/W) *1
b4	DIRQ4F	IRQ4-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ4-DS pin 1: Request for release is being generated on the IRQ4-DS pin	R(/W) *1
b5	DIRQ5F	IRQ5-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ5-DS pin 1: Request for release is being generated on the IRQ5-DS pin	R(/W) *1
b6	DIRQ6F	IRQ6-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ6-DS pin 1: Request for release is being generated on the IRQ6-DS pin	R(/W) *1
b7	DIRQ7F	IRQ7-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ7-DS pin 1: Request for release is being generated on the IRQ7-DS pin	R(/W) *1

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

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

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

These flags indicate that a request for release has been generated on the IRQn-DS pin.
[Setting condition]

- A request for release is generated on the IRQn-DS pin specified by the DPSIEGR0 register.

[Clearing condition]

- 0 is written to these flags after confirming these flags are 1.

11.2.14 Deep Standby Interrupt Flag Register 1 (DPSIFR1)

Address(es): 0008 C287h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ15F	DIRQ14F	DIRQ13F	DIRQ12F	DIRQ11F	DIRQ10F	DIRQ9F	DIRQ8F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ8F	IRQ8-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ8-DS pin 1: Request for release is being generated on the IRQ8-DS pin	R(/W) *1
b1	DIRQ9F	IRQ9-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ9-DS pin 1: Request for release is being generated on the IRQ9-DS pin	R(/W) *1
b2	DIRQ10F	IRQ10-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ10-DS pin 1: Request for release is being generated on the IRQ10-DS pin	R(/W) *1
b3	DIRQ11F	IRQ11-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ11-DS pin 1: Request for release is being generated on the IRQ11-DS pin	R(/W) *1
b4	DIRQ12F	IRQ12-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ12-DS pin 1: Request for release is being generated on the IRQ12-DS pin	R(/W) *1
b5	DIRQ13F	IRQ13-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ13-DS pin 1: Request for release is being generated on the IRQ13-DS pin	R(/W) *1
b6	DIRQ14F	IRQ14-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ14-DS pin 1: Request for release is being generated on the IRQ14-DS pin	R(/W) *1
b7	DIRQ15F	IRQ15-DS Pin Deep Standby Release Flag	0: Request for release is not being generated on the IRQ15-DS pin 1: Request for release is being generated on the IRQ15-DS pin	R(/W) *1

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

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

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

These flags indicate that a request for release has been generated on the IRQn-DS pin.
[Setting condition]

- A request for release is generated on the IRQn-DS pin specified by the DPSIEGR1 register.

[Clearing condition]

- 0 is written to these flags after confirming these flags are 1.

11.2.15 Deep Standby Interrupt Flag Register 2 (DPSIFR2)

Address(es): 0008 C288h

	b7	b6	b5	b4	b3	b2	b1	b0
	DUSBIF	DRIICCIF	DRIICDIF	DNMIF	DRTCAIF	DRTCIIF	DLVD2IF	DLVD1IF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DLVD1IF	LVD1 Deep Standby Release Flag	0: Request for release by the voltage monitor 1 signal is not being generated 1: Request for release by the voltage monitor 1 signal is being generated	R(/W) *1
b1	DLVD2IF	LVD2 Deep Standby Release Flag	0: Request for release by the voltage monitor 2 signal is not being generated 1: Request for release by the voltage monitor 2 signal is being generated	R(/W) *1
b2	DRTCIIF	RTC Periodic Interrupt Deep Standby Release Flag	0: Request for release by the RTC periodic interrupt signal is not being generated 1: Request for release by the RTC periodic interrupt signal is being generated	R(/W) *1
b3	DRTCAIF	RTC Alarm Interrupt Deep Standby Release Flag	0: Request for release by the RTC alarm interrupt signal is not being generated 1: Request for release by the RTC alarm interrupt signal is being generated	R(/W) *1
b4	DNMIF	NMI Deep Standby Release Flag	0: Request for release is not being generated on the NMI pin 1: Request for release is being generated on the NMI pin	R(/W) *1
b5	DRIICDIF	SDA2-DS Deep Standby Release Flag	0: Request for release by the SDA2-DS signal is not being generated 1: Request for release by the SDA2-DS signal is being generated	R(/W) *1
b6	DRIICCIF	SCL2-DS Deep Standby Release Flag	0: Request for release by the SCL2-DS signal is not being generated 1: Request for release by the SCL2-DS signal is being generated	R(/W) *1
b7	DUSBIF	USB0 Suspend/Resume Deep Standby Release Flag	0: Request for release by the USB0 suspend/resume is not being generated 1: Request for release by the USB0 suspend/resume is being generated	R(/W) *1

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

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

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

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

- A request for release is generated by the voltage monitoring m signal that is selected in the DPSIEGR2 register.

[Clearing condition]

- 0 is written to this flag after confirming this flag is 1.

DRTCIIF Flag (RTC Periodic Interrupt Deep Standby Release Flag)

This flag indicates that a request for release by the RTC periodic interrupt signal has been generated.
[Setting condition]

- A request for release by the RTC periodic interrupt signal is generated.

[Clearing condition]

- 0 is written to this flag after confirming this flag is 1.

DRTCAIF Flag (RTC Alarm Interrupt Deep Standby Release Flag)

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

- A request for release by the RTC alarm interrupt signal is generated.

[Clearing condition]

- 0 is written to this flag after confirming this flag is 1.

DNMIF Flag (NMI Deep Standby Release Flag)

This flag indicates that a request for release has been generated on the NMI pin.
[Setting condition]

- A request for release is generated on the NMI pin specified by DPSIEGR2.

[Clearing condition]

- 0 is written to this flag after confirming this flag is 1.

DRIICDIF Flag (SDA2-DS Deep Standby Release Flag)

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

- A request for release is generated on the SDA2-DS pin specified by DPSIEGR2.

[Clearing condition]

- 0 is written to this flag after confirming this flag is 1.

DRIICCIF Flag (SCL2-DS Deep Standby Release Flag)

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

- A request for release is generated on by the SCL2-DS pin specified by DPSIEGR2.

[Clearing condition]

- 0 is written to this flag after confirming this flag is 1.

DUSBIF Flag (USB0 Suspend/Resume Deep Standby Release Flag)

This flag indicates that a request for release by the USB0 suspend/resume has been generated.
The DUSBIF flag is a flag for USB0. USB0 is not released from deep software standby mode using USB0_OVRCURB assigned to pin P22.
Figure 11.2 shows the configuration of the DUSBIF flag.
[Setting condition]

- A request for release by the USB0 suspend/resume is generated.

[Clearing condition]

- 0 is written to this flag after confirming this flag is 1.

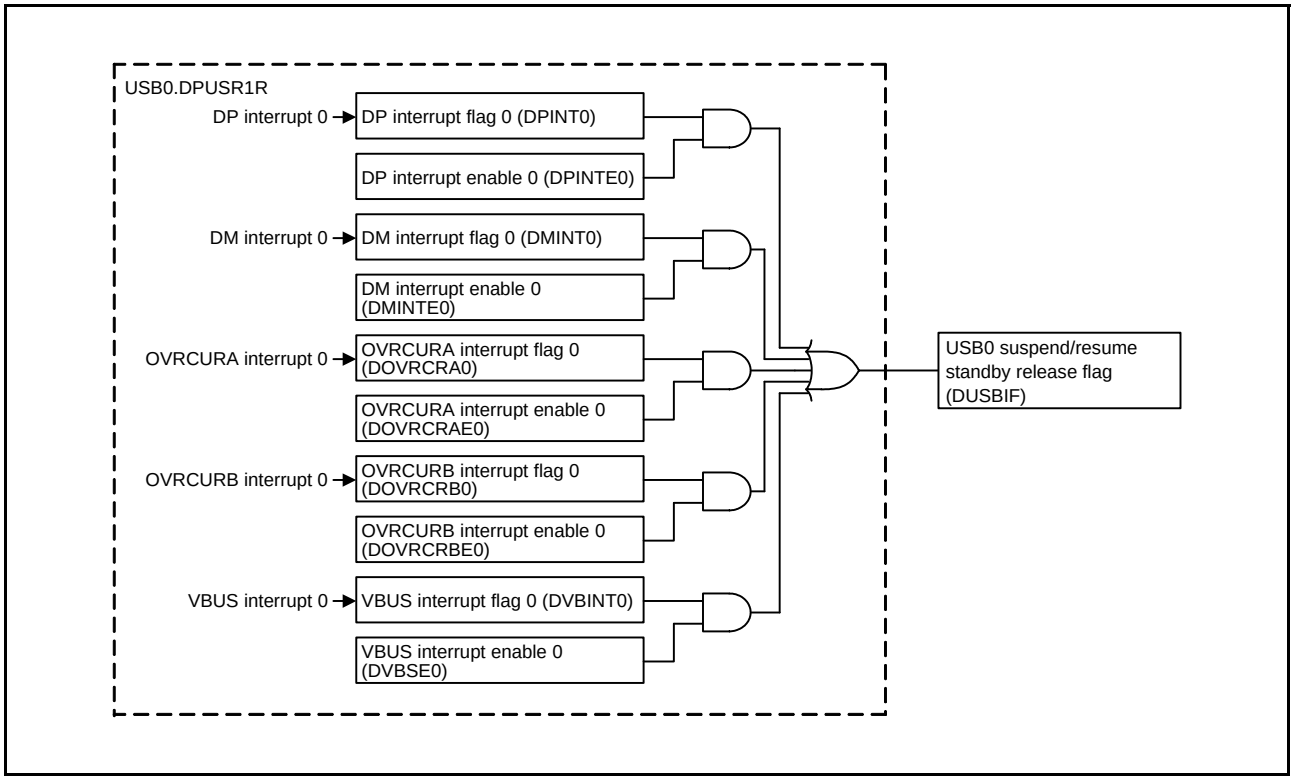


Figure 11.2 Configuration of the USB0 Suspend/Resume Deep Standby Release Flag (DUSBIF)

11.2.16 Deep Standby Interrupt Flag Register 3 (DPSIFR3)

Address(es): 0008 C289h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	DTADIF	DRMCIF	DCANIF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DCANIF	CRX1-DS Deep Standby Release Flag	0: Request for release is not being generated on the CRX1-DS pin. 1: Request for release is being generated on the CRX1-DS pin.	R(/W) *1
b1	DRMCIF	REMC Interrupt Deep Standby Release Flag	0: Request for release by the REMC reception is not being generated. 1: Request for release by the REMC reception is being generated.	R(/W) *1
b2	DTADIF	VBATT Tamper Detection Deep Standby Release Flag	0: Request for release by the VBATT tamper detection is not being generated. 1: Request for release by the VBATT tamper detection is being generated.	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.

Each flag is set to 1 when a request for release specified by the DPSIEGR3 register is generated.
Each flag may be set to 1 when a request for release is generated in any mode (not even deep software standby mode) or when the setting of the DPSIER3 register is modified. Therefore, a transition to deep software standby mode should be made after the DPSIFR3 register is set to 00h.
To set the DPSIFR3 register to 00h after modifying the DPSIER3 register, wait for at least six PCLKB cycles, read the DPSIFR3 register, and then write 0 to the DPSIFR3 register. Six or more PCLKB cycles can be secured, for example, by reading the DPSIER3 register.
The DPSIFR3 register is not initialized by the internal reset signal used as deep software standby mode releasing source. For details, refer to Table 6.2, Targets to be Initialized by Each Reset Source.

DCANIF Flag (CRX1-DS Deep Standby Release Flag)

This flag indicates that a request for release has been generated on the CRX1-DS pin.
[Setting condition]

- A request for release is generated on the CRX1-DS pin specified by the DPSIEGR3 register.

[Clearing condition]

- 0 is written to this flag after confirming this flag is 1.

DRMCIF Flag (REMC Interrupt Deep Standby Release Flag)

This flag indicates that a request for release by the REMC reception has been generated.
[Setting condition]

- A request for release by the REMC reception is generated.

[Clearing condition]

- 0 is written to this flag after confirming this flag is 1.

DTADIF Flag (VBATT Tamper Detection Deep Standby Release Flag)

This flag indicates that a request for release by the VBATT tamper detection has been generated.
[Setting condition]

- A request for release by the VBATT tamper detection is generated.

[Clearing condition]

- 0 is written to this flag after confirming this flag is 1.

11.2.17 Deep Standby Interrupt Edge Register 0 (DPSIEGR0)

Address(es): 0008 C28Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ7EG	DIRQ6EG	DIRQ5EG	DIRQ4EG	DIRQ3EG	DIRQ2EG	DIRQ1EG	DIRQ0EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ0EG	IRQ0-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b1	DIRQ1EG	IRQ1-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b2	DIRQ2EG	IRQ2-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b3	DIRQ3EG	IRQ3-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b4	DIRQ4EG	IRQ4-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b5	DIRQ5EG	IRQ5-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b6	DIRQ6EG	IRQ6-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b7	DIRQ7EG	IRQ7-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W

Register DPSIEGR0 is not initialized by the internal reset signal that is the source for release from deep software standby mode. For details, refer to Table 6.2, Targets to be Initialized by Each Reset Source.

11.2.18 Deep Standby Interrupt Edge Register 1 (DPSIEGR1)

Address(es): 0008 C28Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ15EG	DIRQ14EG	DIRQ13EG	DIRQ12EG	DIRQ11EG	DIRQ10EG	DIRQ9EG	DIRQ8EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ8EG	IRQ8-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b1	DIRQ9EG	IRQ9-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b2	DIRQ10EG	IRQ10-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b3	DIRQ11EG	IRQ11-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b4	DIRQ12EG	IRQ12-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b5	DIRQ13EG	IRQ13-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b6	DIRQ14EG	IRQ14-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b7	DIRQ15EG	IRQ15-DS Pin Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W

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

11.2.19 Deep Standby Interrupt Edge Register 2 (DPSIEGR2)

Address(es): 0008 C28Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	DRIICCEG	DRIICDEG	DNMIEG	—	—	DLVD2EG	DLVD1EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DLVD1EG	LVD1 Edge Select	0: A request for release is generated when VCC < Vdet1 (fall) is detected 1: A request for release is generated when VCC ≥ Vdet1 (rise) is detected	R/W
b1	DLVD2EG	LVD2 Edge Select	0: A request for release is generated when VCC < Vdet2 (fall) is detected 1: A request for release is generated when VCC ≥ Vdet2 (rise) is detected	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DNMIEG	NMI Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b5	DRIICDEG	SDA2-DS Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b6	DRIICCEG	SCL2-DS Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

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

11.2.20 Deep Standby Interrupt Edge Register 3 (DPSIEGR3)

Address(es): 0008 C28Dh

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

Bit	Symbol	Bit Name	Description	R/W
b0	DCANIEG	CRX1-DS Edge Select	0: A request for release is generated at a falling edge 1: A request for release is generated at a rising edge	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

11.3 Reducing Power Consumption by Switching Clock Signals

When the SCKCR.FCK[3:0], ICK[3:0], BCK[3:0], PCKA[3:0], PCKB[3:0], PCKC[3:0], and PCKD[3:0] bits are set, the clock frequency changes. The CPU, DMAC, DTC, code flash memory, and RAM operate on the operating clock specified by the ICK[3:0] bits.
Peripheral modules operate on the operating clock specified by the PCKA[3:0], PCKB[3:0], PCKC[3:0], and PCKD[3:0] bits.
The external bus operates on the operating clock specified by the BCK[3:0] bits. For details, refer to section 9, Clock Generation Circuit.

11.4 Module-Stop Function

The module-stop function can be set for each on-chip peripheral module.
When the MSTPmi bit (m = A to D, i = 31 to 0) in registers MSTPCRA to MSTPCRD is set to 1, the specified module stops operating and enters the module-stop state, but the CPU continues to operate independently. When the corresponding MSTPmi bit is set to 0, the module is released from the module-stop state and restarts operating at the end of the bus cycle. The internal states of modules are retained in the module-stop state.
After release from a reset, all modules except for the DMAC, DTC, EXDMAC, RAM, and standby RAM are placed in the module-stop state. Though read/write access cannot be made to the registers of the module that are in the module-stop state, some registers may be written to directly after the setting to the module-stop state. Therefore, care should be paid.

11.5 Function for Lower Operating Power Consumption

By selecting an appropriate operating power consumption control mode according to the operating frequency and operating voltage, power consumption can be reduced in normal operation, sleep mode and all-module clock stop mode.

11.5.1 Setting Operating Power Consumption Control Mode

Examples of the procedures for switching operating power consumption control modes are shown below:

(1) Switching from Normal Power Consumption Mode to Low Power Consumption Mode

Example: From high-speed operating mode to low-speed operating mode 1

(High-speed operation in the operating power consumption control mode used before mode-switching)
↓
Set to switch from the HOCO clock to the LOCO clock (clock source and frequency division ratio)
↓
Write to register OPCCR (high-speed operating mode → low-speed operating mode 1)
↓
Confirm that the OPCCR.OPCMTSF flag is 0
↓
(Low-speed operation in the switched operating power consumption control mode)

(2) Switching from Low Power Consumption Mode to Normal Power Consumption Mode

Example: From low-speed operating mode 2 to high-speed operating mode

(Low-speed operation in the operating power consumption control mode used before mode-switching)
↓
Write to register OPCCR (low-speed operating mode 2 → high-speed operating mode)
↓
Confirm that the OPCCR.OPCMTSF flag is 0
↓
Set to switch from the LOCO clock to the HOCO clock (clock source and frequency division ratio)
↓
(High-speed operation in the switched operating power consumption control mode)

11.6 Low Power Consumption Modes

11.6.1 Sleep Mode

11.6.1.1 Transition to Sleep Mode

When the WAIT instruction is executed while the SBYCR.SSBY bit is 0, the CPU enters sleep mode. In sleep mode, the CPU stops operating but the contents of its internal registers are retained. Other peripheral functions do not stop.
When the WDT is used, the WDT stops counting when sleep mode is entered.
Counting by the IWDT stops if a transition to sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1 (stopping counting by the IWDT at transitions to low-power-consumption modes). In the same way, counting by the IWDT stops if a transition to sleep mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 1.
Furthermore, counting by the IWDT continues if a transition to sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low-power-consumption modes). In the same way, counting by the IWDT continues if a transition to sleep mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 0.
To use sleep mode, make the following settings and then execute a WAIT instruction.

(1) Set the PSW.I bit*1 of the CPU to 0.
(2) Set the interrupt request destination*2 to be used for recovery from sleep mode to the CPU.
(3) Set the priority*3 of the interrupt to be used for recovery from sleep mode to a level higher than the setting of the PSW.IPL[3:0] bits*1 of the CPU.
(4) Set the IERm.IENj bit*3 for that interrupt to 1.
(5) For the last I/O register to which writing proceeded, read the register to confirm that the value written has been reflected.
(6) Execute the WAIT instruction (this automatically sets the I bit*1 in the PSW of the CPU to 1).

Note 1. For details, refer to section 2, CPU.
Note 2. For details, refer to section 15.7.3, Selecting Interrupt Request Destination.
Note 3. For details, refer to section 15, Interrupt Controller (ICUE).

11.6.1.2 Release from Sleep Mode

Release from sleep mode is initiated by a non-maskable interrupt, an interrupt, the RES# pin reset, a power-on reset, a voltage monitoring reset, or a reset caused by an IWDT underflow.

- Release triggered by an interrupt signal
  Generation of an interrupt triggers release from sleep mode and the interrupt exception processing starts. If a maskable interrupt has been masked by the CPU (the priority level*1 of the interrupt has been set to a value lower than that of the PSW.IPL[3:0] bits*2 of the CPU), release from sleep mode does not proceed.
- Release due to a reset on the RES# pin
  When the RES# pin is driven low, the MCU enters the reset state. When the RES# pin is driven high after the reset signal is input for a predetermined time period, the CPU starts the reset exception processing.
- Release due to a power-on reset
  Release from sleep mode is initiated by a power-on reset.
- Release due to a voltage monitoring reset
  Release from sleep mode is initiated by a voltage monitoring reset from the voltage detection circuit.
- Release due to the independent watchdog timer reset
  Release from sleep mode is initiated by an internal reset generated by an IWDT underflow. However, when such conditions are set that stop IWDT counting in sleep mode (OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1), the IWDT is stopped and release from sleep mode is not initiated by the independent watchdog timer reset.

Note 1. For details, refer to section 15, Interrupt Controller (ICUE).
Note 2. For details, refer to 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, refer to section 11.2.7, Sleep Mode Return Clock Source Switching Register (RSTCKCR). For setting a waiting time for oscillation stabilization, refer to section 9.2.19, Main Clock Oscillator Wait Control Register (MOSCWTCR).

11.6.2 All-Module Clock Stop Mode

11.6.2.1 Transition to All-Module Clock Stop Mode

After setting the MSTPCRA.ACSE bit to 1 and placing modules controlled by MSTPCRA, MSTPCRB, MSTPCRC, and MSTPCRD registers in the module-stop state (MSTPCRA = FFFF FF[C-F]Fh, MSTPCRB = FFFF FFFFh, MSTPCRC[31:16] = FFFFh, MSTPCRD = FFFF FFFFh), executing a WAIT instruction while the SBYCR.SSBY bit is 0 stops the bus controller, I/O ports, and all modules except for the 8-bit timers*1, POE*2, IWDT, RTC, power-on reset circuit, voltage detection circuit at the end of the current bus cycle, and the chip enters all-module clock stop mode*3.
When the WDT is used, the WDT stops counting when all-module clock stop mode is entered.
Counting by the IWDT stops if a transition to all-module clock-stop mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1 (stopping counting by the IWDT at transitions to low-power-consumption modes). In the same way, counting by the IWDT stops if a transition to all-module clock-stop mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 1.
Furthermore, counting by the IWDT continues if a transition to all-module clock-stop mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low-power-consumption modes). In the same way, counting by the IWDT continues if a transition to all-module clock-stop mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 0.
To use all-module clock-stop mode, make the following settings and then execute a WAIT instruction.

(1) Set the PSW.I bit*4 of the CPU to 0.
(2) Set the interrupt request destination*5 to be used for recovery from all-module clock stop mode to the CPU.
(3) Set the priority*6 of the interrupt to be used for recovery from all-module clock stop mode to a level higher than the setting of the PSW.IPL[3:0] bits*4 of the CPU.
(4) Set the IERm.IENj bit*6 for the interrupt to be used for recovery from all-module clock stop mode to 1.
(5) Read the last I/O register to have been written and confirm that its value reflects the value written.
(6) Execute a WAIT instruction (executing a WAIT instruction causes automatic setting of the PSW.I bit*4 of the CPU to 1).

Note 1. The MSTPCRA.MSTPA4 and MSTPA5 bits select operation or stop of these modules.
Note 2. When a POE interrupt source condition is satisfied while the setting to enable POE interrupts is in place, recovery from all-module clock stop mode does not proceed but the flag to indicate satisfaction of the source condition is retained. If a different source leads to recovery from all-module clock stop mode in this situation, a POE interrupt is generated after recovery.
Note 3. Transitions to all-module clock stop mode are not to be made in some states of DTC or DMAC operations. Before setting the MSTPCRA.MSTPA28 bit to 1, clear the DMAST.DMST bit of the DMAC and the DTCST.DTCST bit of the DTC so that the DTC and DMAC are not activated.
Note 4. For details, refer to section 2, CPU.
Note 5. For details, refer to section 15.7.3, Selecting Interrupt Request Destination.
Note 6. For details, refer to section 15, Interrupt Controller (ICUE).

11.6.2.2 Release from All-Module Clock Stop Mode

Release from all-module clock-stop mode is initiated by an external pin interrupt (the NMI or IRQ0 to IRQ15), a peripheral interrupt (8-bit timer*1, RTC alarm, RTC periodic, IWDT, USB0 suspend/resume, REMC reception, VBATT tampering detection, 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*2, and the transition to the normal program execution state proceeds via exception processing for the given interrupt or reset.
However, note that in cases where a maskable interrupt has been masked by the CPU (priority level*3 of the interrupt has been set to a value lower than that of the PSW.IPL[3:0] bits*4 of the CPU) or a maskable interrupt has been set up as a trigger to start the DTC or DMA transfer, release from all-module clock stop mode will not proceed.

Note 1. The MSTPA4 and MSTPA5 bits of MSTPCRA select operation or stopping of these modules.
Note 2. If a condition for the independent watchdog timer to stop counting applied (OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1) at the time of a transition to all-module clock-stop mode, using a reset from the independent watchdog timer to release the chip from all-module clock-stop mode is impossible because the independent watchdog timer is stopped.
Note 3. For details, refer to section 15, Interrupt Controller (ICUE).
Note 4. For details, refer to section 2, CPU.

11.6.3 Software Standby Mode

11.6.3.1 Transition to Software Standby Mode

When a WAIT instruction is executed with the SBYCR.SSBY bit set to 1 and the DPSBYCR.DPSBY bit set to 0, a transition to software standby mode is made. In this mode, the CPU, on-chip peripheral functions, and the oscillator functions stop. However, the contents of the CPU internal registers, RAM data, on-chip peripheral functions, and the states of the I/O ports are retained. Whether the address bus and bus control signals are placed in the high-impedance or the output state is retained can be specified by the SBYCR.OPE bit. Software standby mode allows significant reduction in power consumption because the oscillator stops in this mode. Yet, the main and sub clock oscillators can be operated or stopped. For details, refer to Table 11.2, Entering and Exiting Low Power Consumption Modes and Operating States in Each Mode.

Set the DMAST.DMST bit of the DMAC and the DTCST.DTCST bit of the DTC to 0 before executing the WAIT instruction.

When the WDT is used, the WDT stops counting when software standby mode is entered because the oscillator stops. Counting by the IWDT stops if a transition to software standby mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1 (stopping counting by the IWDT at transitions to low-power-consumption modes). In the same way, counting by the IWDT stops if a transition to software standby mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 1.

Furthermore, counting by the IWDT continues if a transition to software standby mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low-power-consumption modes). In the same way, counting by the IWDT continues if a transition to software standby mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 0.

When the oscillation stop detection function is enabled (OSTDCR.OSTDE = 1), software standby mode cannot be entered. To make a transition to software standby mode, execute a WAIT instruction after disabling the oscillation stop detection function (OSTDCR.OSTDE = 0).

To use software standby mode, make the following settings and then execute a WAIT instruction.

(1) Set the PSW.I bit*1 of the CPU to 0.
(2) Set the interrupt request destination*2 to be used for recovery from software standby mode to the CPU.
(3) Set the priority*3 of the interrupt to be used for recovery from software standby mode to a level higher than the setting of the PSW.IPL[3:0] bits*1 of the CPU.
(4) Set the IERm.IENj bit*3 for the interrupt to be used for recovery from software standby mode to 1.
(5) For the last I/O register to which writing proceeded, read the register to confirm that the value written has been reflected.
(6) Execute a WAIT instruction (executing a WAIT instruction causes automatic setting of the PSW.I bit*1 of the CPU to 1).

Note 1. For details, refer to section 2, CPU.
Note 2. For details, refer to section 15.7.3, Selecting Interrupt Request Destination.
Note 3. For details, refer to section 15, Interrupt Controller (ICUE).

11.6.3.2 Release from Software Standby Mode

Release from software standby mode is initiated by an external pin interrupt (the NMI or IRQ0 to IRQ15), peripheral interrupts (the RTC alarm, RTC periodic, IWDT, USB0 suspend/resume, REMC reception, VBATT tampering detection, voltage monitoring 1, and voltage monitoring 2 interrupts), an RES# pin reset, a power-on reset, a voltage monitoring reset, or an independent watchdog timer reset.
When an interrupt initiates release from software standby, the oscillators which were stopped by the transition to software standby are restarted. After the oscillation of all these oscillators has become stable, operation returns from software standby.
Note that the oscillators are not stopped by the transition to software standby under the following two conditions, but the return from software standby still follows the period for stabilization of oscillation by the oscillators if they had been stopped.

- MOFCR.MOFXIN = 1 and MOSCCR.MOSTP = 0
- RCR3.RTCEN = 1 and SOSCCR.SOSTP = 0

(1) Release due to an interrupt
When an interrupt request from the NMI, IRQ0 to IRQ15, RTC alarm, RTC periodic, IWDT, USB0 suspend/resume, REMC reception, VBATT tampering detection, voltage monitoring 1, or voltage monitoring 2 interrupt is generated, each oscillator which was operating before a transition to software standby mode resumes oscillation. After the time for recovery from software standby mode has elapsed, the chip is released from software standby and starts interrupt exception processing.
The time for recovery from software standby mode is the oscillation stabilization waiting time plus the time required for operations by the software standby release sequencer.

tSBYi = tSBYOSCWT + tSBYSEQ
tSBYi (i = MC, EX, PC, PE, PH, SC, HO, LO): Recovery time from software standby mode
tSBYOSCWT: Oscillation stabilization waiting time
tSBYSEQ: Time required for operations by the software standby release sequencer

For the oscillation stabilization waiting time to be used in calculating the time for recovery from software standby mode, use the greatest value of the oscillation stabilization waiting time of the oscillators which are to be started. For the oscillation stabilization waiting times of the oscillators, refer to section 56, Electrical Characteristics.

(2) Release due to a reset on the RES# pin
Clock oscillation starts when the low level is applied to the RES# pin. Clock supply for the MCU starts at the same time. Keep the level on the RES# pin low over the time required for oscillation of the clocks to become stable. Reset exception processing starts when the high level is applied to the RES# pin.

(3) Release due to a power-on reset
Release from software standby mode proceeds after a fall in the power-supply voltage leads to the generation of a power-on reset.

(4) Release due to a voltage monitoring reset
Release from software standby mode proceeds after a fall in the power-supply voltage leads to the generation of a voltage monitoring reset.

(5) Release due to an independent watchdog timer reset
An internal reset due to an underflow of the IWDT leads to release from software standby mode.
However, if a condition for the independent watchdog timer to stop counting applied (OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1) at the time of a transition to software standby, using a reset from the independent watchdog timer to release the chip from software standby is impossible because the independent watchdog timer is stopped.

11.6.3.3 Example of Software Standby Mode Application

Figure 11.3 shows an example where a transition to software standby mode is made at the falling edge of the IRQn pin, and release from software standby mode is initiated at the rising edge of the IRQn pin.
In this example, an IRQn interrupt is accepted with the IRQCRi.IRQMD[1:0] bits of the ICU set to 01b (falling edge), and then the IRQCRi.IRQMD[1:0] bits are set to 10b (rising edge). After that, the SBYCR.SSBY bit is set to 1 and the WAIT instruction is executed. Thus a transition to software standby mode is made. After that, release from software standby mode is initiated at the rising edge of the IRQn pin.
To return from software standby mode, settings of the interrupt controller (ICU) are also necessary. For details, refer to section 15, Interrupt Controller (ICUE).

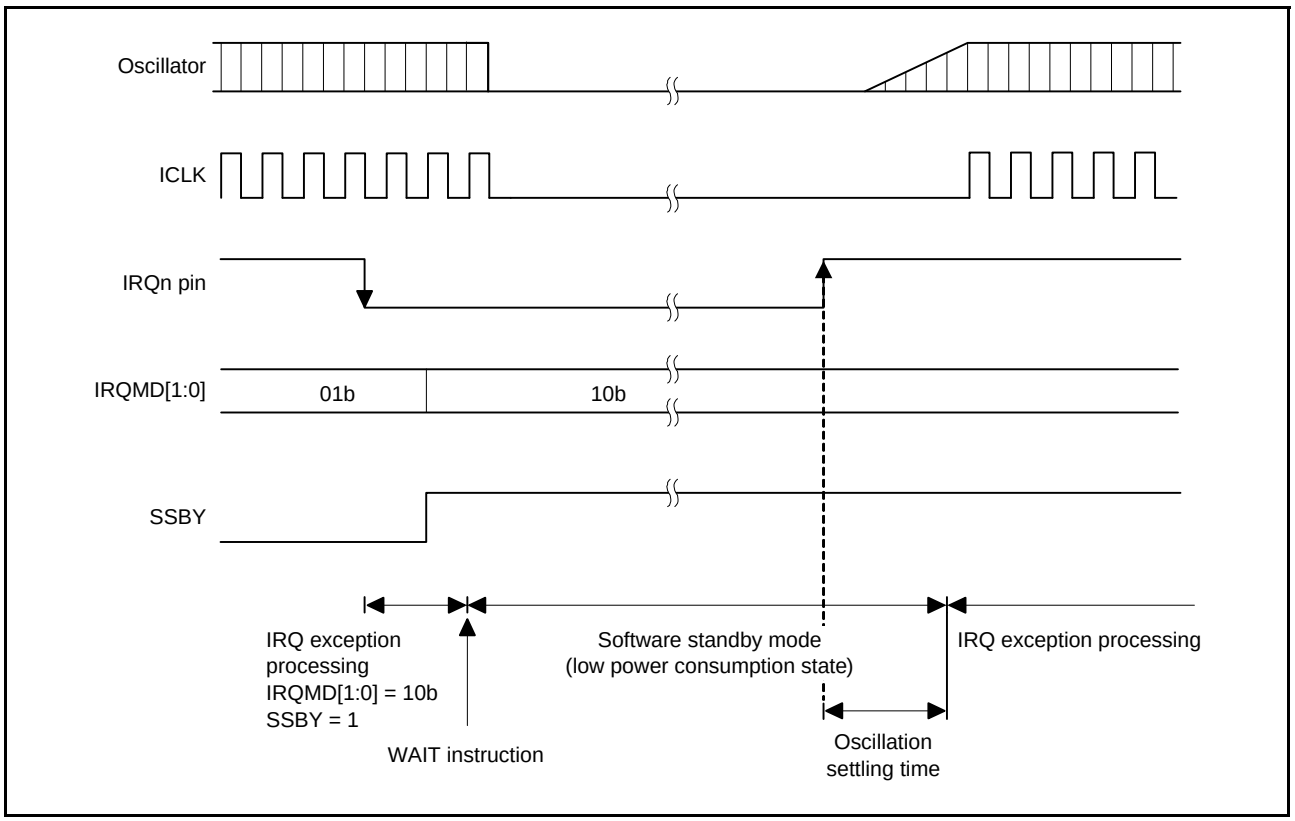


Figure 11.3 Example of Software Standby Mode Application

11.6.4 Deep Software Standby Mode

11.6.4.1 Transition to Deep Software Standby Mode

When the WAIT instruction is executed with the SBYCR.SSBY bit set to 1, a transition to software standby mode*1 is made. At this time, when the DPSBYCR.DPSBY bit is set to 1, a transition to deep software standby mode is made. On deep software standby mode, the CPU, internal peripheral modules (except for parts of the RTC alarm, RTC periodic, SCL2-DS, SDA2-DS, CRX1-DS, USB0 suspend/resume detecting unit, and REMC), RAM, and functions of the oscillators are stopped; furthermore, since the internal supply of power for these modules is stopped, power consumption is markedly reduced. Yet, the main and sub clock oscillators can be operated or stopped. For details, refer to Table 11.2, Entering and Exiting Low Power Consumption Modes and Operating States in Each Mode. At this time, the contents of all the registers of the CPU and internal peripheral modules (except for parts of the RTC alarm, RTC periodic, SCL2-DS, SDA2-DS, CRX1-DS, USB0 suspend/resume detecting unit, and REMC) become undefined.

Data in the standby RAM are preserved if the setting of the DEEPCUT[1:0] bits is 00b. If the setting of the DEEPCUT[1:0] bits is 01b, the internal supply of power to the standby RAM, REMC, and the USB0 resume detecting unit is cut off, reducing power consumption. Data in the standby RAM become undefined at this time. If the setting of the DEEPCUT[1:0] bits is 11b, the internal supply of power to the standby RAM, REMC, and the USB0 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. At this time, the detection voltage of the power-on reset circuit differs from that when the low power consumption function is disabled. For details, refer to section 56, Electrical Characteristics.

When the WDT is in use, since the oscillators and power supply to the WDT are stopped by the transition to deep software standby mode, counting also stops.

Power supply to the IWDT-dedicated clock and the IWDT is stopped and counting by the IWDT stops if a transition to deep software standby mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1 (stopping counting by the IWDT at transitions to low-power-consumption modes). In the same way, power supply to the IWDT-dedicated clock and the IWDT is stopped and counting by the IWDT stops if a transition to deep software standby mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 1.

Furthermore, counting by the IWDT continues if a transition to software standby mode is made but not to deep software standby mode while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low-power-consumption modes). In the same way, counting by the IWDT continues if a transition to software standby mode is made but not to deep software standby mode while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 0.

When the voltage monitoring 1 reset function (LVD1CR0.LVD1RI = 1) or voltage monitoring 2 reset function (LVD2CR0.LVD2RI = 1) is selected for the voltage detection circuit, a transition to deep software standby mode cannot be made, but to software standby.

The I/O port states remain unchanged from software standby mode.

Note 1. Conditions on the DTC, DMAC, and IWDT for transition to software standby mode should be met before the WAIT instruction is executed. For details, refer to section 11.6.3, Software Standby Mode.

11.6.4.2 Release from Deep Software Standby Mode

Release from deep software standby mode is initiated by any of the external pin interrupt source pins (the NMI, IRQ0-DS to IRQ15-DS, SCL2-DS, SDA2-DS, or CRX1-DS), peripheral interrupts (the RTC alarm, RTC periodic, USB0 suspend/resume*1, REMC reception, VBATT tampering detection, voltage monitoring 1, and voltage monitoring 2 interrupts), an RES# pin reset, a power-on reset, or a voltage monitoring 0 reset.

Note 1. USB0 is not released from deep software standby mode using USB0_OVRCURB assigned to pin P22.

(1) Release triggered by an external interrupt pin or internal interrupt signal
Release from deep software standby mode is controlled by registers DPSIERn (n = 0 to 3) and DPSIFRn (n = 0 to 3). When a deep software standby release interrupt is generated, the corresponding flag in DPSIFRn is set to 1. At this time, if the releasing source is enabled in DPSIERn, release from deep software standby mode proceeds. Rising edge or falling edge can be selected by DPSIEGRn (n = 0 to 3) registers. The interrupts for which an edge can be selected are the NMI, IRQ0-DS to IRQ15-DS, SCL2-DS, SDA2-DS, CRX1-DS, voltage monitoring 1, and voltage monitoring 2 interrupts.
When a deep software standby mode releasing source is generated, the internal power supply and LOCO clock oscillation begin, and then a deep software standby reset is generated for the entire MCU.
A stable LOCO clock is then supplied to the entire MCU, which is released from deep software standby reset. This is accompanied by release from deep software standby, and reset exception processing then starts.
When release from deep software standby is triggered by an external interrupt pin or internal interrupt signal, the RSTSR0.DPSRSTF flag is set to 1.

(2) Release due to a reset on the RES# pin
The low level being applied to the RES# pin triggers release from deep software standby.
At this time, the RES# pin should be held low according to the specifications described in section 56, Electrical Characteristics. Reset exception processing starts when the high level is applied to the RES# pin.

(3) Release due to a power-on reset
Release from deep software standby mode proceeds after a fall in the power-supply voltage leads to the generation of a power-on reset.

(4) Release due to a voltage monitoring 0 reset
Release from deep software standby mode proceeds after a fall in the power-supply voltage leads to the generation of a voltage monitoring 0 reset.

11.6.4.3 Pin States at the Time of Release from Deep Software Standby Mode

In deep software standby mode, the I/O ports retain the same states from software standby mode. The inside of the MCU is initialized by an internal reset generated on release from deep software standby mode. Upon release from deep software standby mode, the reset exception processing starts. The following shows the states of I/O ports at this time. Whether to initialize the I/O ports or to retain the I/O port states at the time of software standby mode can be selected by the DPSBYCR.IOKEEP bit.

- When the DPSBYCR.IOKEEP bit = 0
  I/O ports are initialized by an internal reset generated on release from deep software standby mode.
- When the DPSBYCR.IOKEEP bit = 1
  Although the inside of the MCU is initialized by an internal reset generated on release from deep software standby mode, I/O ports retain their states from software standby mode regardless of the MCU 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 setting the DPSBYCR.IOKEEP bit to 0, and the MCU operates according to the internal state.

The DPSBYCR.IOKEEP bit is not initialized by an internal reset generated on release from deep software standby mode.

11.6.4.4 Example of Deep Software Standby Mode Application

Figure 11.4 shows an example where a transition to deep software standby mode is made at the falling edge of the IRQn-DS pin, and release from deep software standby mode is initiated at the rising edge of the IRQn-DS pin.
In this example, an IRQn interrupt is accepted with the IRQCRi.IRQMD[1:0] bits of the ICU set to 01b (falling edge). Then, after the DPSIEGRy.DIRQnEG (y = 0 or 1, n = 0 to 15) bit is set to 1 (rising edge) and the SBYCR.SSBY bit and DPSBYCR.DPSBY bit are both set to 1, the WAIT instruction is executed. Thus a transition to deep software standby mode is made.
After that, release from deep software standby mode is initiated at the rising edge of the IRQ-DS pin.

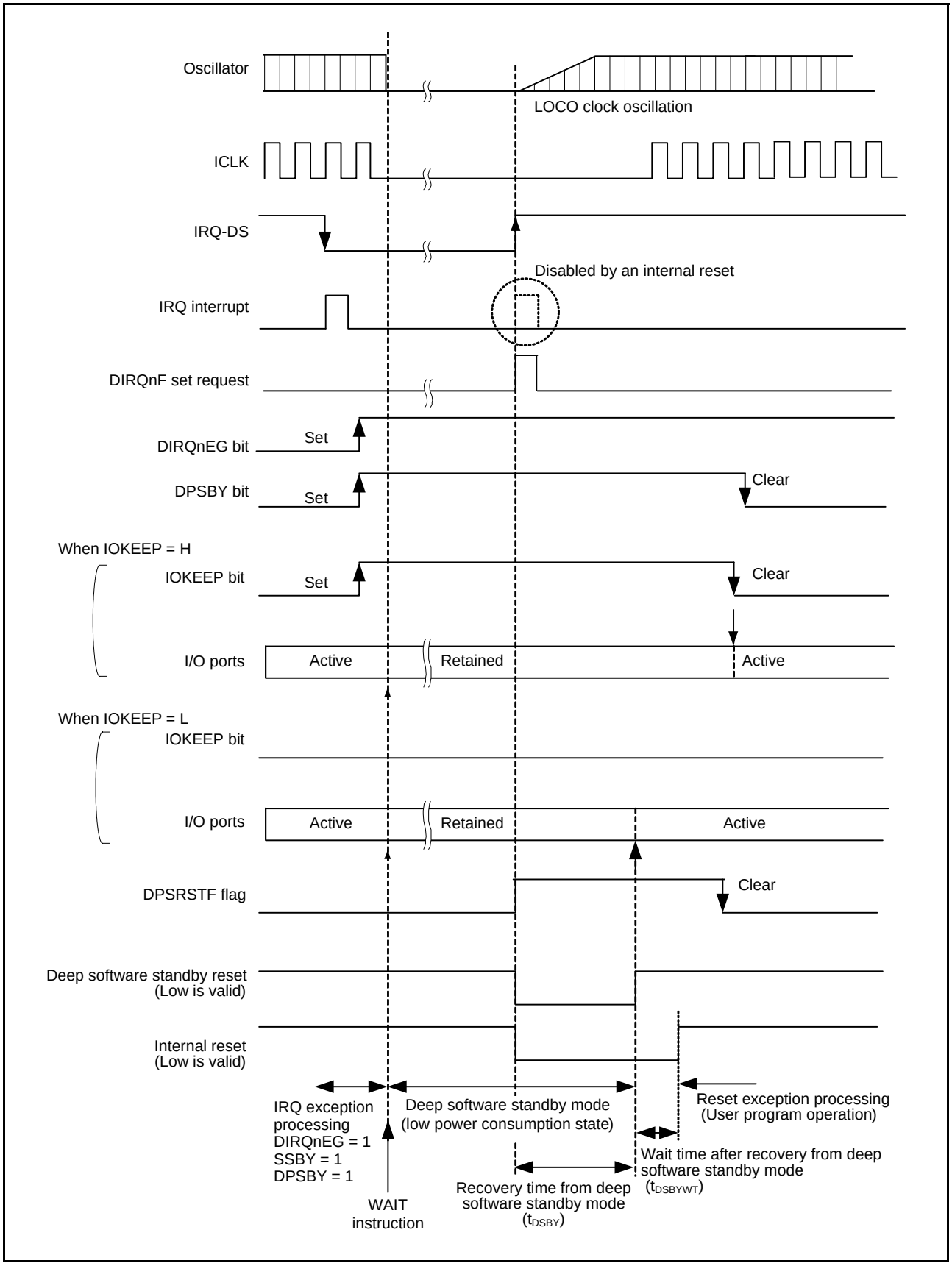


Figure 11.4 Example of Deep Software Standby Mode Application

11.6.4.5 Flowchart to Use Deep Software Standby Mode

Figure 11.5 shows an example of a flowchart to use deep software standby mode.
In this example, the RSTSR0.DPSRSTF flag of the reset function is read after the reset exception processing to determine whether a reset was generated by the RES# pin or by release from deep software standby mode.
In the case of a reset by the RES# pin, a transition to deep software standby mode is made after the required register settings have been made.
In the case of a reset by release from deep software standby mode, the DPSBYCR.IOKEEP bit is set to 0 after the I/O port settings have been made.

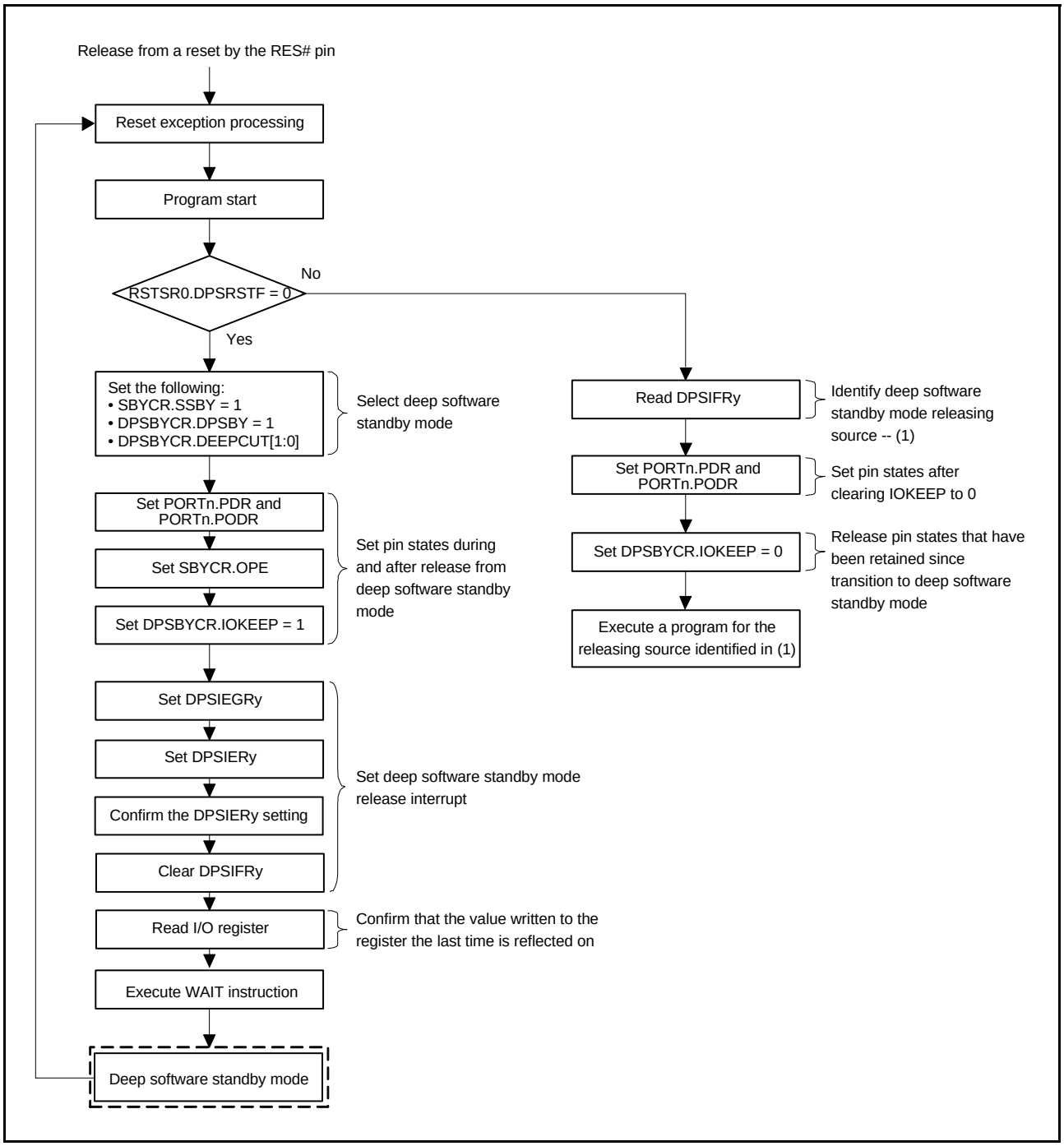


Figure 11.5 Example of Flowchart to Use Deep Software Standby Mode

11.7 Usage Notes

11.7.1 I/O Port States

I/O port states are retained in software standby mode and deep software standby mode.

11.7.2 Module-Stop State of DMAC and DTC

Before setting the MSTPCRA.MSTPA28 bit to 1, set 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, refer to section 18, DMA Controller (DMACAb) and section 20, Data Transfer Controller (DTCb).

11.7.3 On-Chip Peripheral Module Interrupts

These interrupts do not operate in the module-stop state. Therefore, if the module-stop state is entered after an interrupt request is generated, a CPU interrupt source or a DMAC or DTC startup source cannot be cleared. For this reason, disable interrupts before entering the module-stop state.

11.7.4 Write Access to MSTPCRA, MSTPCRB, MSTPCRC, and MSTPCRD

Write accesses to registers MSTPCRA, MSTPCRB, MSTPCRC and MSTPCRD should be made only by the CPU.

11.7.5 Input Buffer Control by DIRQnE Bit (n = 0 to 15)

Setting the DPSIERy.DIRQnE (y = 0 or 1, n = 0 to 15) bit to 1 enables the input buffer of the IRQ0-DS to IRQ15-DS pins. Therefore, note that, although inputs to these pins are sent to the DPSIFRy.DIRQnF (y = 0 or 1, n = 0 to 15) bits, they are not sent to the interrupt controller, peripheral modules, and I/O ports.

11.7.6 Timing of WAIT Instructions

A WAIT instruction that follows register writing may be executed before the writing is completed. Accordingly, the WAIT instruction may be executed before the change to the setting of an I/O register is reflected, in which case operation may not be as intended. To avoid this, always execute the WAIT instruction after confirming that the last writing to the register has completed.

11.7.7 Rewriting the Register by DMAC and DTC in Sleep Mode

The WDT stops in sleep mode. Do not set up the 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 Point for Caution when Shifting from Low-Speed Operating Mode to Software Standby Mode

On return from software standby, the chip enters high-speed operating mode. Even if a WAIT instruction is executed in low-speed operating mode, if generation of the return interrupt precedes completion of the transition to software standby and processing for the transition is canceled, the chip does not return to the mode before execution of the WAIT instruction. If this creates a problem, set the OPCCR.OPCM[2:0] bits to 000b during processing of the return interrupt.

12. Battery Backup Function (VBATTB)

12.1 Overview

Backup power supplied by a battery can keep the realtime clock (RTC) operating and retain the important data while the VCC power is turned off. The tamper detection function can be used to erase the backed-up data in cases where an intrusion to the system has been detected.

Table 12.1 lists the specifications of the VBATT module, Figure 12.1 is a block diagram, and Table 12.2 lists the I/O pins of the VBATT module.

Table 12.1 VBATT Specifications

Item	Description
Scope of backup	All modules in the backup domain • Backup registers • Sub-clock oscillator • Power-down detector • Tamper detector • Realtime clock (RTC)
Backup registers	128 bytes • The backup registers can immediately be erased in response to the detection of physical tampering.
Backup domain power-down detection	A backup domain reset signal is generated when the supply voltage for the backup domain has fallen below V_{PDR}.
Tamper event detection	When the tamper detector has detected an intrusion to the system, it indicates this by setting a flag and, if this is selected, generating an interrupt. • Each tamper event can acquire a timestamp on tamper detection. • Three tamper input pins (TAMPI0 to TAMPI2) • Noise filter (sampling rate: 32.768 kHz; passes signal transitions in response to three consecutive samples with the same level) • Capable of triggering wakeup from deep software standby mode

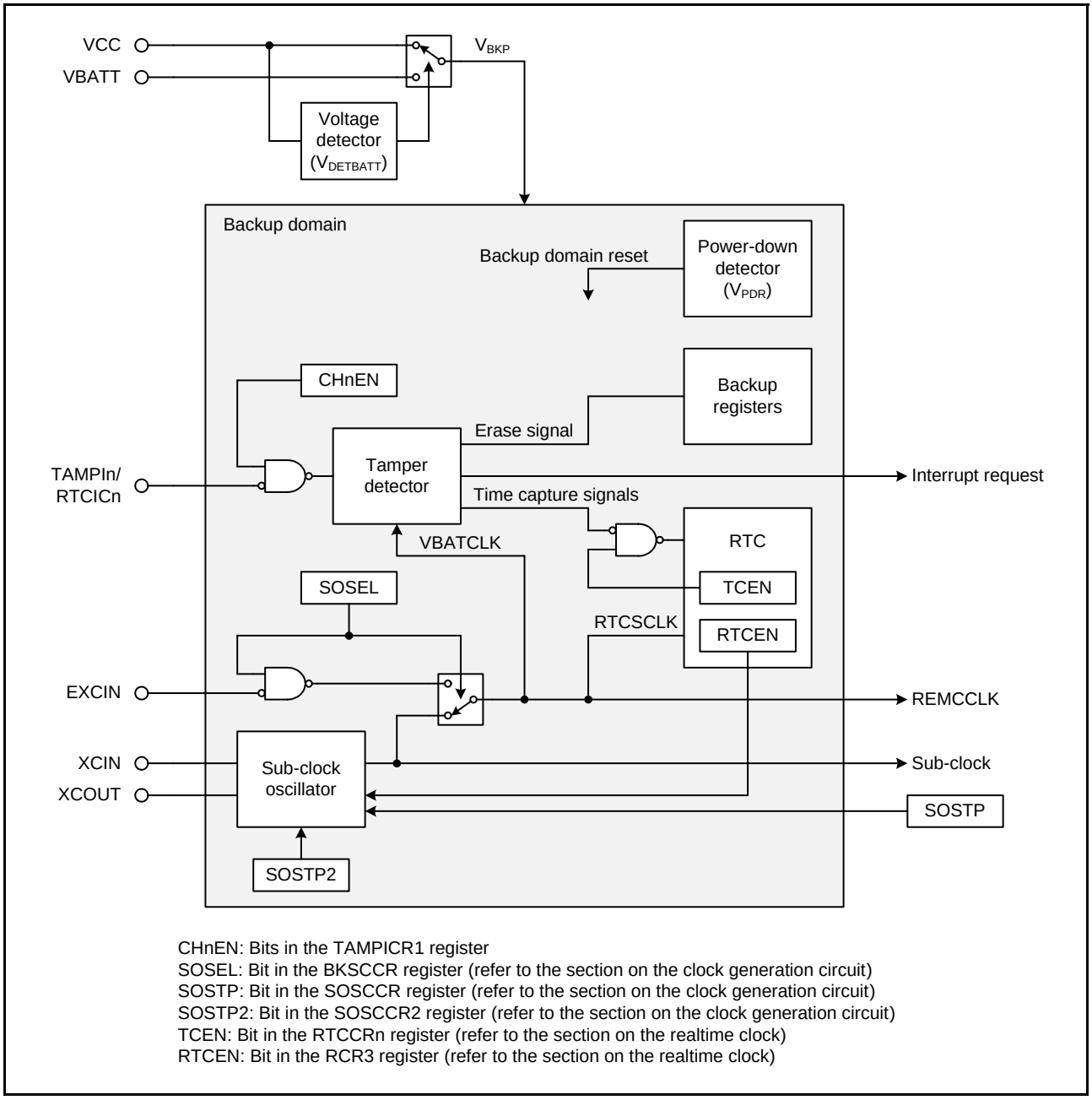


Figure 12.1 Block Diagram of VBATT (n = 0 to 2)

Table 12.2 VBATT Input/Output Pins

Pin Name	I/O	Function
VCC	Input	Power-supply pin
VBATT	Input	Pin for backup battery (power-supply pin)
XCIN	Input	Pins for a crystal for the sub-clock Connect a 32.768-kHz crystal to these pins.
XCOUT	Output	
EXCIN	Input	Clock input pin for an external sub-clock oscillator
TAMPI0 to TAMPI2	Input	Tamper input pins
RTCIC0 to RTCIC2	Input	Time capture event input pins for RTC

12.2 Register Descriptions

Registers in the battery backup function (VBATT) can only be initialized by the backup domain reset. Resets generated by other sources such as the RES# pin or software will not affect these registers.

12.2.1 Backup Domain Power Status Register (BKPSR)

Address(es): SYSTEM.BKPSR 0008 CC46h

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

Bit	Symbol	Bit Name	Description	R/W
b0	PDDF	Power-Down Detected Flag	0: The voltage for the backup domain has not fallen below V_{PDR}. 1: The voltage for the backup domain has fallen below V_{PDR}.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	—	Reserved	This bit is read as 1. Writing to this bit has no effect.	R
b5	—	Reserved	The read value is undefined. Writing to this bit has no effect.	R
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be initialized by the backup domain reset. It is not affected by the MCU reset.

The BKPSR register indicates the state of the backup domain power supply.

PDDF Flag (Power-Down Detected Flag)

The PDDF flag indicates that the supply voltage for the backup domain has fallen below V_{PDR} and a backup domain reset was generated.

[Setting condition]

- The voltage supplied to the backup domain falling below V_{PDR}.

[Clearing conditions]

- 0 being written to the PDDF flag.

12.2.2 Tamper Status Register (TAMPSR)

Address(es): SYSTEM.TAMPSR 0008 CC48h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TAMP2F	TAMP1F	TAMP0F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TAMP0F	Tamper 0 Detection Flag	0: Tamper 0 event has not been detected. 1: Tamper 0 event has been detected.	R/(W) *1
b1	TAMP1F	Tamper 1 Detection Flag	0: Tamper 1 event has not been detected. 1: Tamper 1 event has been detected.	R/(W) *1
b2	TAMP2F	Tamper 2 Detection Flag	0: Tamper 2 event has not been detected. 1: Tamper 2 event has been detected.	R/(W) *1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be initialized by the backup domain reset. It is not affected by the MCU reset.
Note 1. Only 0 can be written to this bit, to clear the flag. To clear this flag, confirm that its value is 1 and then write 0 to it.

The TAMPSR register indicates the status of the tamper detection events.

TAMPnF Flag (Tamper n Detection Flag) (n = 0 to 2)

Each TAMPnF flag indicates that a trigger (an effective edge) was input to the TAMPIn pin. The trigger is selected by the TAMPICR2.CHnTRG bit.

[Setting condition]

- The trigger set by the TAMPICR2.CHnTRG bit being detected on the TAMPIn pin.

[Clearing conditions]

- 0 being written to the TAMPnF flag after having confirmed that its value is 1.

12.2.3 Tamper Control Register (TAMPCR)

Address(es): SYSTEM.TAMPCR 0008 CC49h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	TAMP2EE	TAMP1EE	TAMP0EE	—	TAMP2IE	TAMP1IE	TAMP0IE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TAMP0IE	Tamper 0 Detection Interrupt Enable	0: Tamper 0 detection interrupt is disabled. 1: Tamper 0 detection interrupt is enabled.	R/W
b1	TAMP1IE	Tamper 1 Detection Interrupt Enable	0: Tamper 1 detection interrupt is disabled. 1: Tamper 1 detection interrupt is enabled.	R/W
b2	TAMP2IE	Tamper 2 Detection Interrupt Enable	0: Tamper 2 detection interrupt is disabled. 1: Tamper 2 detection interrupt is enabled.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	TAMP0EE	Tamper 0 Erase Enable	0: Backup registers are not erased in response to a tamper 0 event. 1: Backup registers are erased in response to a tamper 0 event.	R/W
b5	TAMP1EE	Tamper 1 Erase Enable	0: Backup registers are not erased in response to a tamper 1 event. 1: Backup registers are erased in response to a tamper 1 event.	R/W
b6	TAMP2EE	Tamper 2 Erase Enable	0: Backup registers are not erased in response to a tamper 2 event. 1: Backup registers are erased in response to a tamper 2 event.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: This register can only be initialized by the backup domain reset. It is not affected by the MCU reset.

The TAMPCR register controls operations in response to the detection of tampering.

TAMPnIE Bit (Tamper n Detection Interrupt Enable) (n = 0 to 2)

The TAMPnIE bit enables or disables the tamper n detection interrupt.

TAMPnEE Bit (Tamper n Erase Enable) (n = 0 to 2)

The TAMPnEE bit selects whether the backup registers should be or should not be erased in response to a tamper n event.

12.2.4 Time Capture Event Control Register (TCECR)

Address(es): SYSTEM.TCECR 0008 CC4Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TCE2S	TCE1S	TCE0S
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TCE0S	Time Capture Event 0 Source Select	0: Input signal from the RTCIC0 pin 1: Tamper 0 event	R/W
b1	TCE1S	Time Capture Event 1 Source Select	0: Input signal from the RTCIC1 pin 1: Tamper 1 event	R/W
b2	TCE2S	Time Capture Event 2 Source Select	0: Input signal from the RTCIC2 pin 1: Tamper 2 event	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be initialized by the backup domain reset. It is not affected by the MCU reset.

The TCECR register selects the sources for time capture events of the RTC.

TCEnS Bit (Time Capture Event n Source Select) (n = 0 to 2)

The TCEnS bit selects the source of time capture event n for the RTC.

When this bit is set to 0, a signal input to the RTCICn pin is input to the RTC. Selection of this setting is recommended when tamper detection is not to be used.

When this bit is set to 1, the tamper n event detection signal is input to the RTC. Selection of this setting is recommended when a tamper event is to cause a timestamp to be recorded.

12.2.5 Tamper/RTCIC Input Control Register 1 (TAMPICR1)

Address(es): SYSTEM.TAMPICR1 0008 CC4Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CH2EN	CH1EN	CH0EN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CH0EN	Channel 0 Input Enable	0: The TAMPI0/RTCIC0 signal input is disabled. 1: The TAMPI0/RTCIC0 signal input is enabled.	R/W
b1	CH1EN	Channel 1 Input Enable	0: The TAMPI1/RTCIC1 signal input is disabled. 1: The TAMPI1/RTCIC1 signal input is enabled.	R/W
b2	CH2EN	Channel 2 Input Enable	0: The TAMPI2/RTCIC2 signal input is disabled. 1: The TAMPI2/RTCIC2 signal input is enabled.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be initialized by the backup domain reset. It is not affected by the MCU reset.

The TAMPICR1 register enables or disables input from the TAMPIn/RTCICn pins (n = 0 to 2).

CHnEN Bit (Channel n Input Enable) (n = 0 to 2)

The CHnEN bit enables or disables the input from the TAMPIn/RTCICn pin.

Set each bit to 1 not only if tamper events are to be detected but also if the RTC time capture function is to be used.

12.2.6 Tamper/RTCIC Input Control Register 2 (TAMPICR2)

Address(es): SYSTEM.TAMPICR2 0008 CC4Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	CH2TRG	CH1TRG	CH0TRG	—	CH2NFE	CH1NFE	CH0NFE
Value after reset:	0	1	1	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CH0NFE	Channel 0 Noise Filter Enable	0: Noise filter for the TAMPI0/RTCIC0 pin is disabled. 1: Noise filter for the TAMPI0/RTCIC0 pin is enabled.	R/W
b1	CH1NFE	Channel 1 Noise Filter Enable	0: Noise filter for the TAMPI1/RTCIC1 pin is disabled. 1: Noise filter for the TAMPI1/RTCIC1 pin is enabled.	R/W
b2	CH2NFE	Channel 2 Noise Filter Enable	0: Noise filter for the TAMPI2/RTCIC2 pin is disabled. 1: Noise filter for the TAMPI2/RTCIC2 pin is enabled.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	CH0TRG	Channel 0 Trigger Select	Select the trigger for tamper 0 event detection. 0: A falling edge of the input on the TAMPI0 pin 1: A rising edge of the input on the TAMPI0 pin	R/W
b5	CH1TRG	Channel 1 Trigger Select	Select the trigger for tamper 1 event detection. 0: A falling edge of the input on the TAMPI1 pin 1: A rising edge of the input on the TAMPI1 pin	R/W
b6	CH2TRG	Channel 2 Trigger Select	Select the trigger for tamper 2 event detection. 0: A falling edge of the input on the TAMPI2 pin 1: A rising edge of the input on the TAMPI2 pin	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: This register can only be initialized by the backup domain reset. It is not affected by the MCU reset.

The TAMPICR2 register is used to enable or disable the noise filters for the TAMPIn/RTCICn pins (n = 0 to 2) and select the triggers for tamper event detection.

CHnNFE Bit (Channel n Noise Filter Enable) (n = 0 to 2)

The CHnNFE bit enables or disables the noise filter for the TAMPIn/RTCICn pin. When tamper n event detection is not to be used, set the bit to 0.

Five cycles of the sub-clock are necessary for the output of the noise filter to become stable after this bit has been set to 1.

CHnTRG Bit (Channel n Trigger Select) (n = 0 to 2)

The CHnTRG bit is used to select the effective edge for the input signal from the TAMPIn pin for use as a trigger of tamper event detection.

12.2.7 Tamper/RTCIC Input Monitoring Register (TAMPIMR)

Address(es): SYSTEM.TAMPIMR 0008 CC4Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CH2LVL	CH1LVL	CH0LVL
Value after reset:	0	0	0	0	0	x	x	x

Bit	Symbol	Bit Name	Description	R/W
b0	CH0LVL	Channel 0 Level Monitoring Flag	0: The low level is being input on the TAMPI0 pin. 1: The high level is being input on the TAMPI0 pin.	R
b1	CH1LVL	Channel 1 Level Monitoring Flag	0: The low level is being input on the TAMPI1 pin. 1: The high level is being input on the TAMPI1 pin.	R
b2	CH2LVL	Channel 2 Level Monitoring Flag	0: The low level is being input on the TAMPI2 pin. 1: The high level is being input on the TAMPI2 pin.	R
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: This register can only be initialized by the backup domain reset. It is not affected by the MCU reset.

The TAMPIMR register is used to monitor the input levels on the TAMPIn pins (n = 0 to 2).

CHnLVL Flag (Channel n Level Monitoring Flag) (n = 0 to 2)

Each CHnLVL flag indicates the input level on an TAMPIn pin.

12.2.8 Backup Register n (BKRn) (n = 0 to 127)

Address(es): SYSTEM.BKR0 0008 CE00h to SYSTEM.BKR127 0008 CE7Fh

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

Note: These registers can only be initialized by the backup domain reset. They are not affected by the MCU reset.

The contents of the BKRn registers are retained even in the battery backup mode.
The BKRn registers can be erased when a tamper event is detected.

12.3 Operation

12.3.1 Battery Backup Function

This function is used to retain the operations of the modules in the backup domain by supplying power from the VBATT pin (battery backup mode) while the main power supply is turned off.
If the VCC voltage falls below $V_{DETBATT}$, the backup domain power supply is automatically switched to VBATT. When the VCC voltage rises back to $V_{DETBATT}$, the backup domain power supply is automatically switched back to VCC.
When the battery backup function is to be used, enable the voltage monitor 0 reset.
Figure 12.2 shows the structure of the power switches and Figure 12.3 shows switching of the power supply for the backup domain.

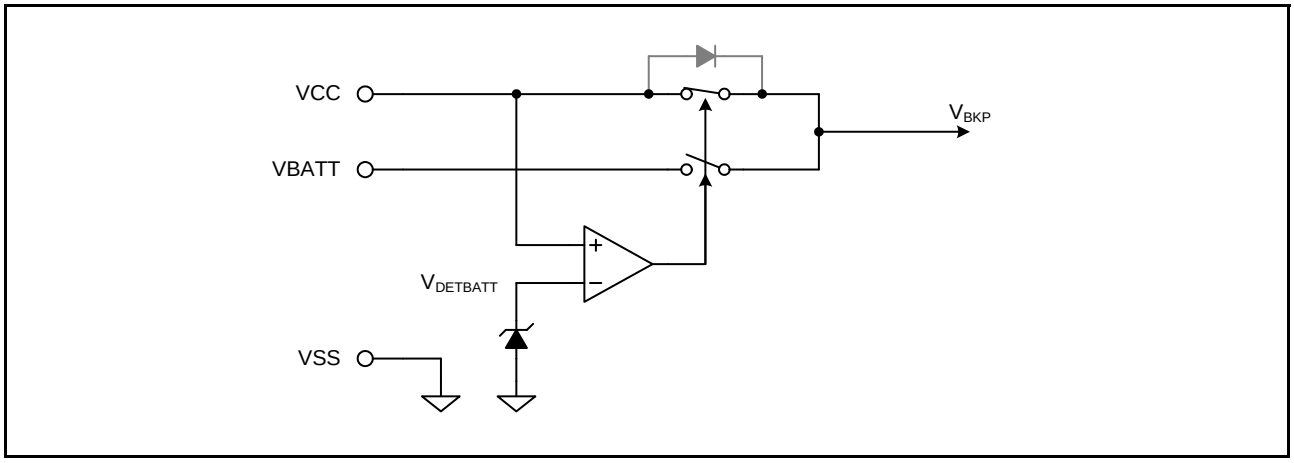


Figure 12.2 Power Switches

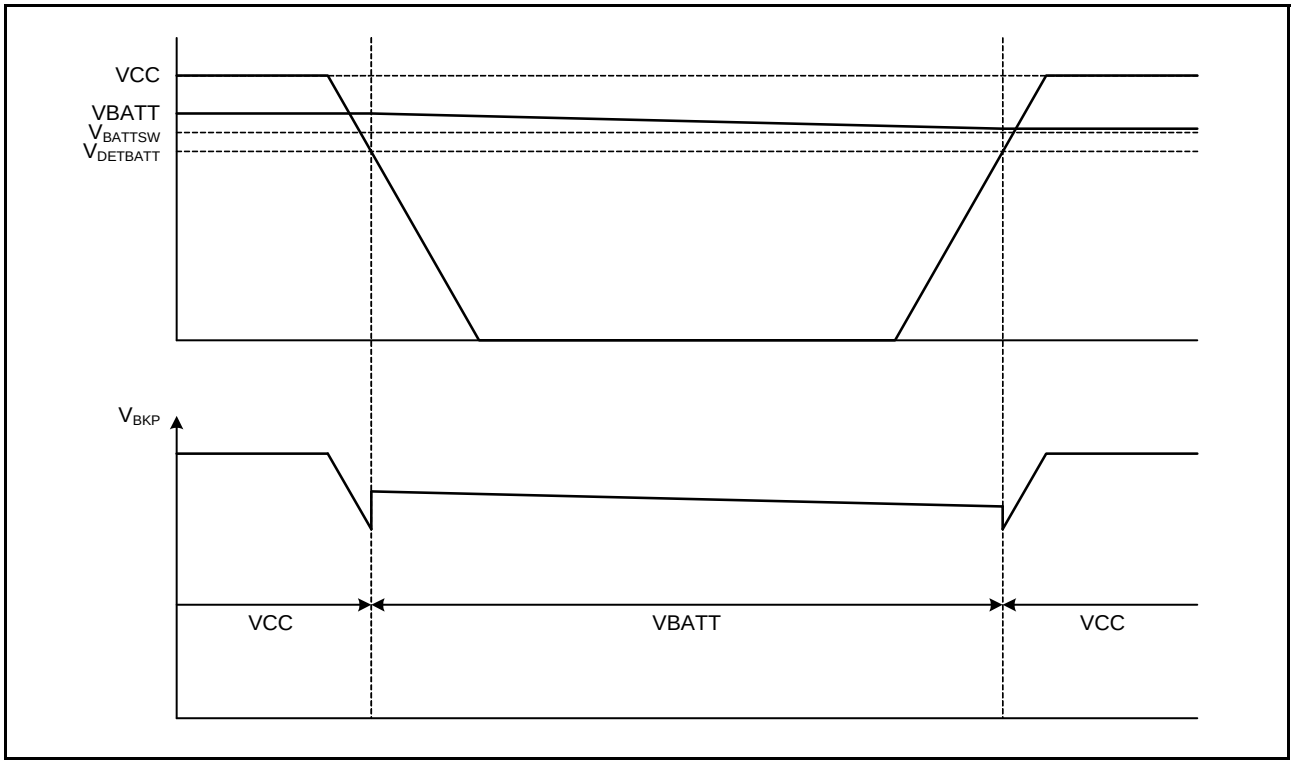


Figure 12.3 Switching of the Power Supply for the Backup Domain

The following modules are in the backup domain.

- Backup registers
- Sub-clock oscillator
- Power-down detector
- Tamper detector
- RTC

The VBATT pin only supplies power for the following pins in the battery backup mode.

- The XCIN and XCOUT pins for a crystal resonator
- The EXCIN (PJ3) pin for an external crystal oscillator
- The TAMPI0 (P30), TAMPI1 (P31), and TAMPI2 (P32) pins for tamper detection

12.3.2 Sub-Clock Oscillator

The sub-clock oscillator is in the backup domain. The backup domain has a register to stop the sub-clock oscillation so that re-starting of oscillation can be prevented even if the VCC power is turned off while oscillation by the sub-clock oscillator has stopped.
Figure 12.4 is a block diagram of the sub-clock oscillator and Table 12.3 lists the relations between bit settings and the state of the sub-clock oscillator in each operating mode of the MCU.

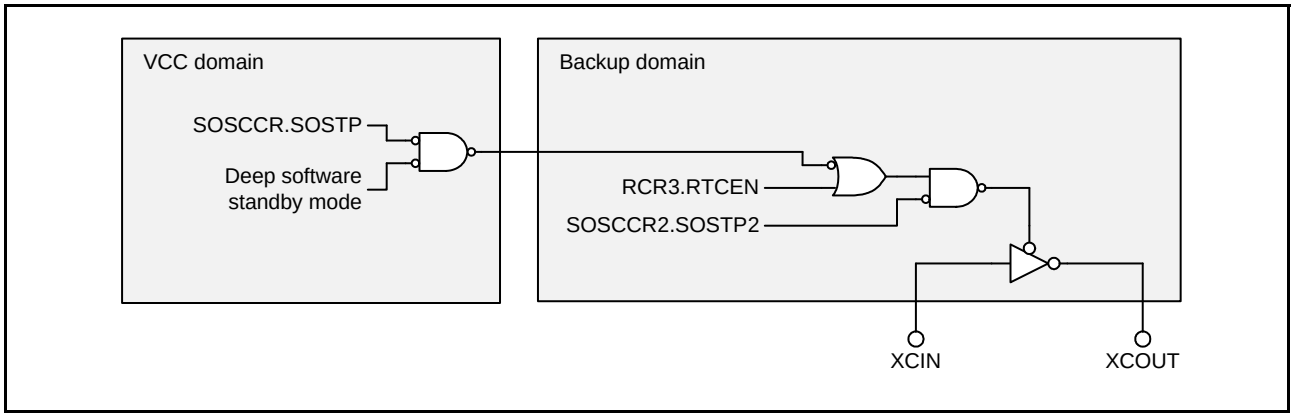


Figure 12.4 Block Diagram of the Sub-Clock Oscillator

Table 12.3 Bit Settings and Sub-Clock Operation

MCU Operating Mode	SOSCCR2.SOSTP2	SOSCCR.SOSTP	RCR3.RTCEN	Sub-Clock Oscillator
Normal operating mode, sleep mode, all-module clock stop mode, and software standby mode	0	0	x	Oscillates
	0	1	0	Stopped
	0	x	1	Oscillates
	1	1*1	x	Stopped
Deep software standby mode	0	1*2	0	Stopped
	0	1*2	1	Oscillates
	1	1*1	x	Stopped
Battery backup mode	0	0*3	x	Oscillates
	1	0*3	x	Stopped

Note 1. When the SOSTP2 bit is set to 1, also set the SOSTP bit to 1.
Note 2. The output is fixed to 1 in the deep software standby mode.
Note 3. The output becomes 0 V (= logic 0) when VCC is turned off.

12.3.3 Power-Down Detector

This module monitors the supply voltage to the backup domain (V_{BKP}) and generates a backup domain reset signal. Specifically, the backup domain reset signal is asserted when the V_{BKP} voltage falls below $V_{PDR(BKP)}$. When the V_{BKP} voltage rises above $V_{PDR(BKP)}$, the reset signal is negated.

A backup domain reset only affects the registers described in this chapter and the SOSCCR2 and BKSCCR registers described in the chapter on the clock generation circuit. However, the reset does not affect the registers and counters in the RTC.

The value of the BKPSR.PDDF flag is 1 when a backup domain reset had been generated. If the BKPSR.PDDF flag is 1, set all registers in the backup domain again.

Figure 12.5 shows the structure of the power-down detector and Figure 12.6 shows its operation.

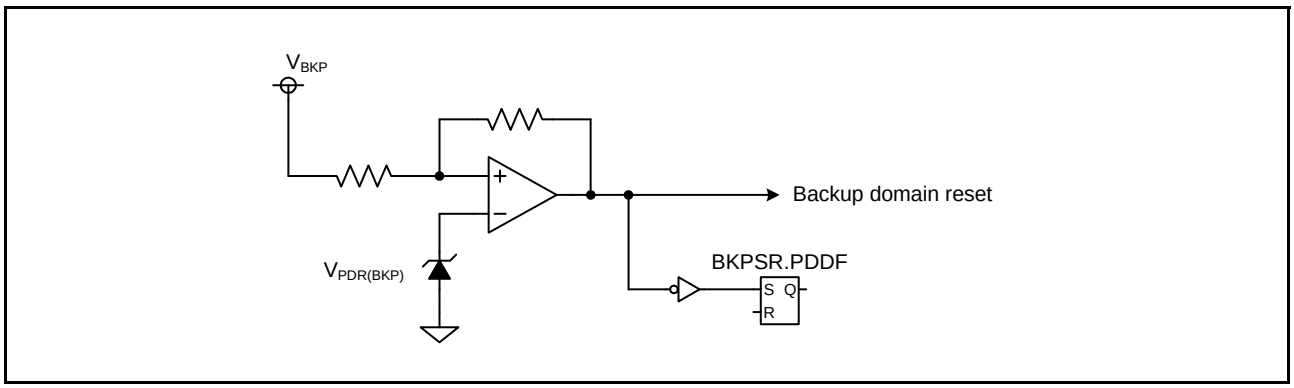


Figure 12.5 Structure of the Power-Down Detector

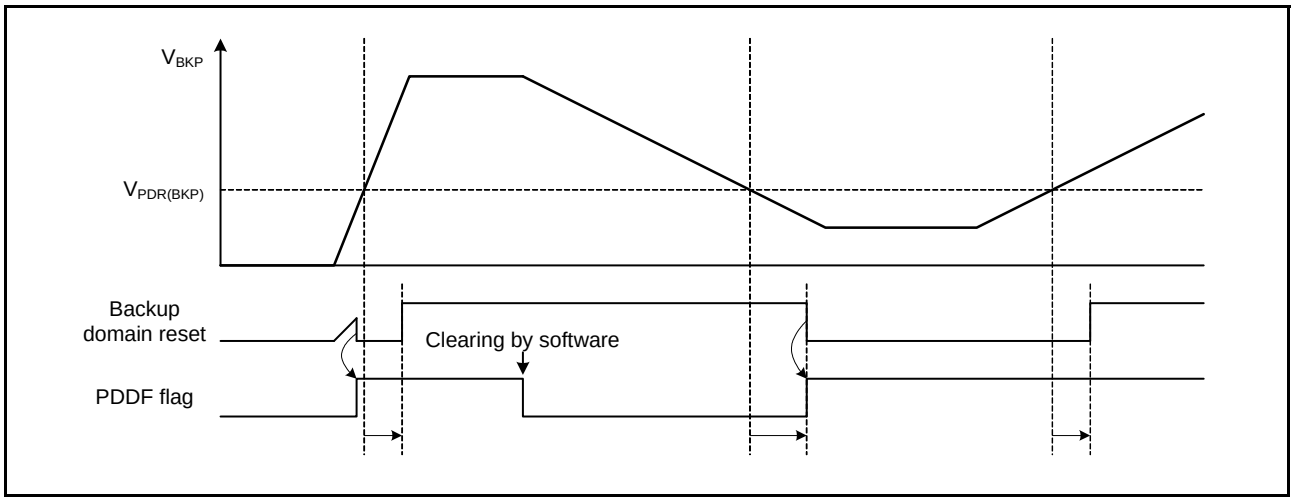


Figure 12.6 Operation of the Power-Down Detector

12.3.4 Tamper Detector

This MCU has three tamper pins (TAMPI0 to TAMPI2) which remain functional in the battery backup mode and can detect physical intrusions.
The tamper inputs can be configured for rising or falling edge detection and with or without noise filtering.
The following operations can be selected in response to the detection of tampering.

- Generating an interrupt request
- Erasing the backup registers
- Generating RTC timestamp events

Figure 12.7 is a block diagram of the tamper detector.

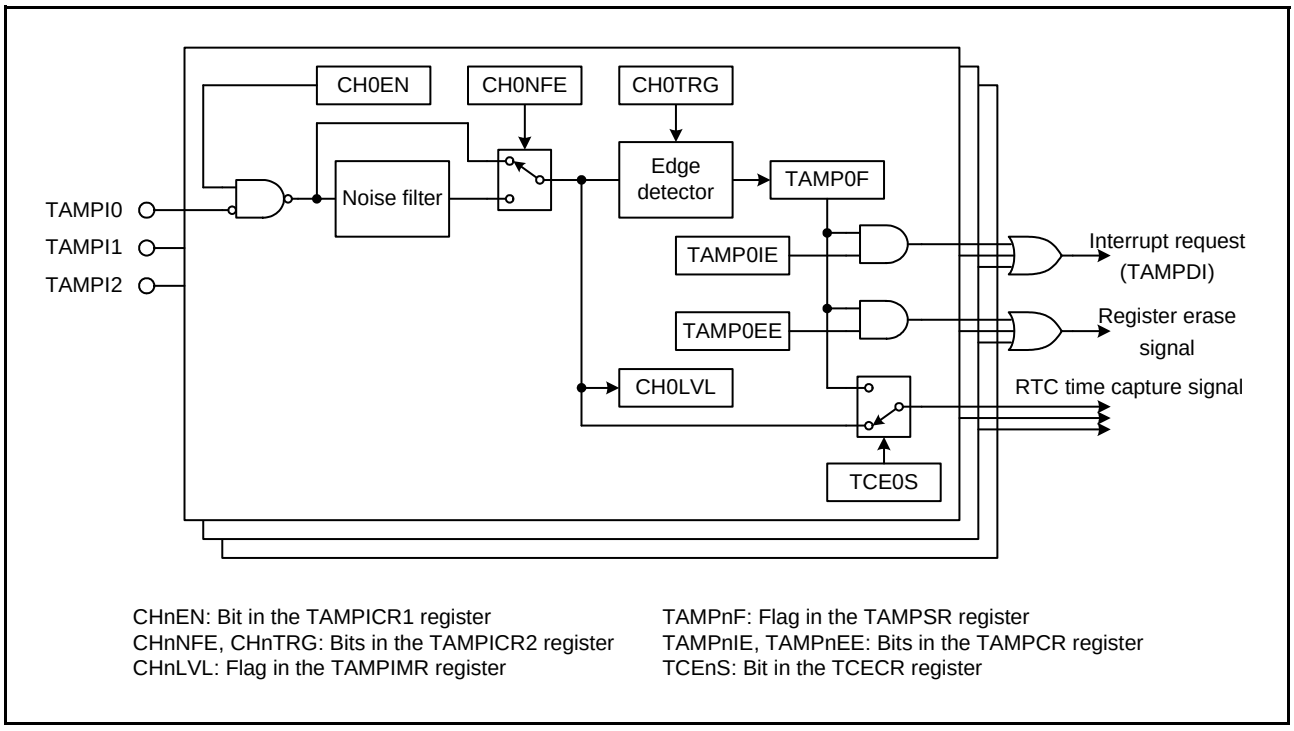


Figure 12.7 Tamper detector (n = 0 to 2)

If the value of the TAMPICR1.CHnEN bit or the CHnNFE or CHnTRG bit in the TAMPICR2 register is changed, a TAMPSR.TAMPnF flag may become 1 (n = 0 to 2). Set the TAMPnIE and TAMPnEE bits in the TAMPCR register to 0 before setting these bits. Clear the TAMPnF flags after setting these bits.
The TAMPIMR.CHnLVL flags are for monitoring the levels being input on the TAMPIn pins.
When an effective edge is detected in any of the channels, the corresponding TAMPnF flag becomes 1. If the TAMPnIE bit is 1 at this point, the TAMPDI interrupt is generated. If the TAMPnEE bit is 1, the backup registers are erased to 00h.
Setting the TCECR.TCEnS bit to 1 enables a timestamp on tamper detection by using the time capture function of the RTC. For details on the time capture function, refer to section 31, Realtime Clock (RTCd).

Figure 12.8 shows the setting flow when using the tamper detector.

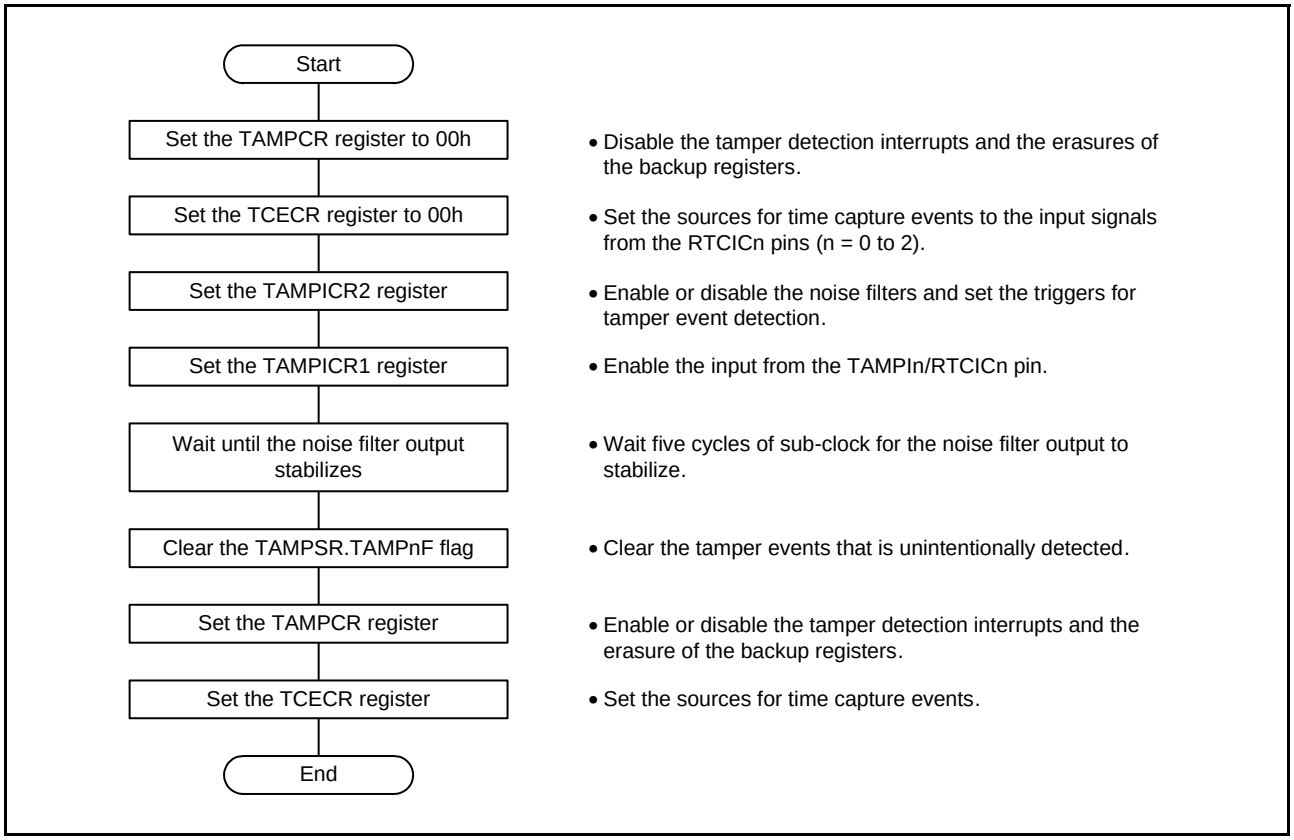


Figure 12.8 Setting Flow for Tamper Detector

12.4 Interrupt

The VBATT has the three interrupt sources listed in Table 12.4.

Table 12.4 VBATT Interrupt Sources

Name	Symbol	Interrupt Source	Interrupt Flag	Interrupt Enable Bit
Tamper detection interrupt	TAMPDI	Tamper 0 detection	TAMP0F	TAMP0IE
		Tamper 1 detection	TAMP1F	TAMP1IE
		Tamper 2 detection	TAMP2F	TAMP2IE

12.5 Usage Notes

12.5.1 Voltage Monitor 0 Reset

When the battery backup function is to be used, enable the voltage monitor 0 reset.

12.5.2 When the Battery Backup Function is Not to be Used

If the battery backup function is not to be used, externally connect the VBATT pin to the VCC pin.

12.5.3 Current Injection Flowing into the VBATT Pin

During operation in the battery backup mode, the voltage of the VCC pin being higher than V_{BATT} may inject a current into the VBATT pin through the parasitic diode connected to the power switch (refer to Figure 12.2). If the power supply or battery connected to the VBATT pin cannot support this current injection, it is recommended that an external low-drop diode be connected between this power supply and the VBATT pin.

13. Register Write Protection Function

The register write protection function protects important registers from being overwritten for in case a program runs out of control. The registers to be protected are set with the protect register (PRCR).

Table 13.1 lists the association between the PRCR bits and the registers to be protected.

Table 13.1 Association between PRCR Bits and Registers to be Protected

PRCR Bit	Register to be Protected
PRC0	• Registers related to the clock generation circuit: SCKCR, SCKCR2, SCKCR3, PLLCR, PLLCR2, BCKCR, MOSCCR, SOSCCR, LOCOCR, ILOCOCR, HOCOCR, HOCOCR2, OSTDCR, OSTDSR, CKOCR, FLLCR1, FLLCR2, HOCOTRR0, HOCOTRR1, HOCOTRR2, CTSUTRMR
PRC1	• Registers related to the operating modes: SYSCR0, SYSCR1 • Registers related to the low power consumption functions: SBYCR, MSTPCRA, MSTPCRB, MSTPCRC, MSTPCRD, OPCCR, RSTCKCR, DPSBYCR, DPSIER0 to DPSIER3, DPSIFR0 to DPSIFR3, DPSIEGR0 to DPSIEGR3 • Registers related to clock generation circuit: MOSCWTCR, SOSCWTCR, MOFCR, HOCOPCR • Software reset register: SWRR • Battery backup-related registers: BKSCCR, BKPSR, SOSCCR2, TAMPSR, TAMPCR, TCECR, TAMPICR1, TAMPICR2, TAMPIMR
PRC3	• Registers related to the LVD: LVCMPCR, LVDLVLR, LVD1CR0, LVD1CR1, LVD1SR, LVD2CR0, LVD2CR1, LVD2SR

13.1 Register Descriptions

13.1.1 Protect Register (PRCR)

Address(es): 0008 03FEh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PRKEY[7:0]								—	—	—	—	PRC3	—	PRC1	PRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	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, clock generation circuit, low power consumption, software reset, and battery backup. 0: Write disabled 1: Write enabled	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	PRC3	Protect Bit 3	Enables writing to the registers related to the LVD. 0: Write disabled 1: Write enabled	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	PRKEY[7:0]	PRC Key Code	These bits control permission and prohibition of writing to the PRCR register. To modify the PRCR register, write A5h to the eight higher-order bits and the desired value to the eight lower-order bits as a 16-bit unit.	R/(W)*1

Note 1. Written values are not retained. These bits are read as 00h.

PRCi Bits (Protect Bit i) (i = 0, 1, 3)

These bits enable or disable writing to the corresponding registers to be protected.

Setting the PRCi bits to 1 and 0 enables and disables writing to the corresponding registers to be protected, respectively.

14. Exception Handling

14.1 Exception Events

During execution of a program by the CPU, the occurrence of a certain event may cause execution of that program to be suspended and execution of another program to be started. Such kinds of events are called exception events.
The RXv3 CPU supports nine types of exceptions. The types of exception events are shown in Figure 14.1.
The occurrence of an exception causes the processor mode to shift to supervisor mode.

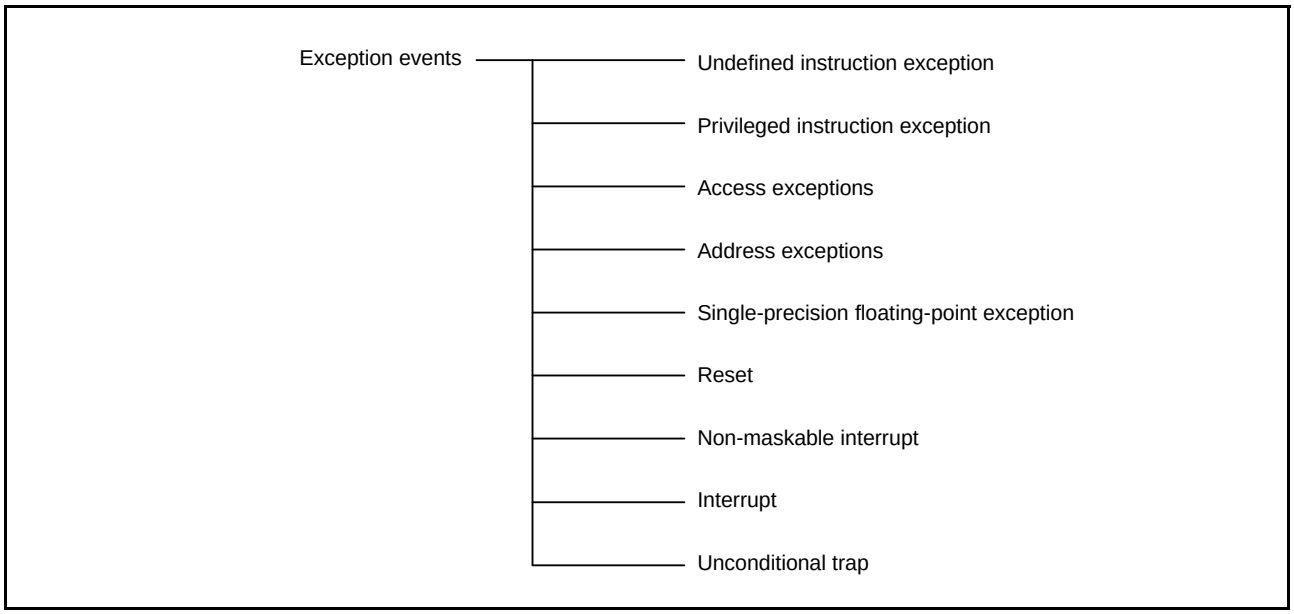


Figure 14.1 Types of Exception Events

14.1.1 Undefined Instruction Exception

An undefined instruction exception occurs when execution of an undefined instruction (an instruction not implemented) is detected.

14.1.2 Privileged Instruction Exception

A privileged instruction exception occurs when execution of a privileged instruction is detected in user mode. Privileged instructions can be executed only in supervisor mode.

14.1.3 Access Exceptions

An access exception occurs when an error is detected in access to memory by the CPU. If the memory-protection unit detects an instruction memory-protection error, an instruction-access exception occurs, and if the unit detects a data memory protection error, an operand-access exception occurs.

14.1.4 Address Exceptions

Address exceptions are generated in response to 64-bit operand access to addresses that are not on 32-bit boundaries.

14.1.5 Single-Precision Floating-Point Exception

Single-precision floating-point exceptions are generated when any of the five exceptions specified in the IEEE 754 standard, namely overflow, underflow, inexact, division-by-zero, or invalid operation, or an attempt to use processing that is not implemented, is detected upon execution of a single-precision floating-point operation instruction. Exception handling by the CPU only proceeds when any among the EX, EU, EZ, EO, or EV bits in the FPSW, which corresponding to the five types of exception, is set to 1.

14.1.6 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.

14.1.7 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.

14.1.8 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.

14.1.9 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.

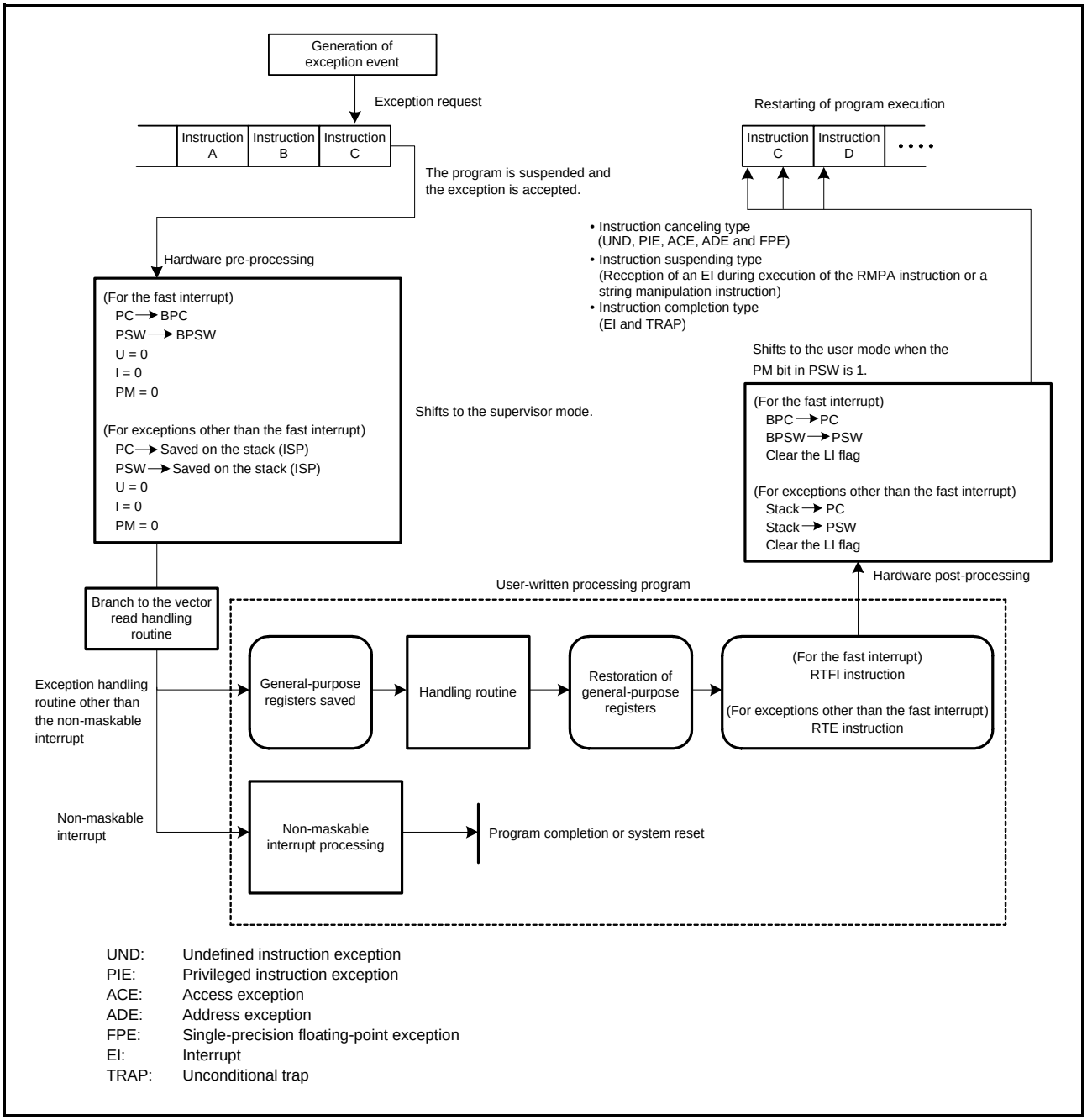


Figure 14.2 Outline of Exception Handling Procedure

When an exception is accepted, hardware processing by the RXv3 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 RXv3 CPU handles saving of the values of the program counter (PC) and processor status word (PSW). In the case of the fast interrupt, the values of the PC and PSW are saved in the backup PC (BPC) and the backup PSW (BPSW), respectively. In the case of exceptions other than the fast interrupt, the contents are saved on the stack. The values of general purpose registers and control registers other than the PC and PSW that are to be used within an exception handling routine must be saved by the user program at the start of the exception handling routine.

On completion of processing by an exception handling routine, saved registers 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 RXv3 CPU handles restoration of the contents of the PC and PSW. In the case of the fast interrupt, the values of the BPC and BPSW are restored to the PC and PSW, respectively. In the case of other exceptions, the values are restored from the stack to the PC and PSW.

The stack or the register-saving bank can be used to save and restore the general-purpose and other registers at the start and end of an exception handling routine.

Saving to and restoring from the register-saving bank is executed by using the SAVE and RSTR instructions. To save and restore a register that is not within the scope of saving and restoring by the SAVE and RSTR instructions, use the PUSH and POP instructions for saving to and restoring from the stack.

Using the register-saving bank is usually faster than using the stack, except when the number of registers that require saving and restoring in transitions to and from an exception-handling routine is extremely small.

14.3 Acceptance of Exception Events

When an exception occurs, the CPU suspends the execution of the program and processing branches to the exception handling routine.

14.3.1 Acceptance Timing and Saved PC Value

Table 14.1 lists the timing of acceptance and the program counter (PC) value to be saved for each exception event.

Table 14.1 Acceptance Timing and Saved PC Value

Exception Event		Type of Handling	Acceptance Timing	Value Saved in BPC or on the Stack
Undefined instruction exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Privileged instruction exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Access exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Address exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Single-precision floating-point exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Reset		Instruction abandonment type	Any machine cycle	None
Non-maskable interrupt	During execution of the RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE instructions	Instruction suspending type	During instruction execution	PC value of the instruction being executed
	Other than above	Instruction completion type	At the next break between instructions	PC value of the next instruction
Interrupt	During execution of the RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE instructions	Instruction suspending type	During instruction execution	PC value of the instruction being executed
	Other than above	Instruction completion type	At the next break between instructions	PC value of the next instruction
Unconditional trap		Instruction completion type	At the next break between instructions	PC value of the next instruction

14.3.2 Vector and Site for Saving the Values in the PC and PSW

The vector for each type of exception and the site for saving the values of the program counter (PC) and processor status word (PSW) are listed in Table 14.2. The addresses where the exception vector table and interrupt vector table start must be set. For details, see section 2.6, Vector Table.

Table 14.2 Vector and Site for Saving the Values in the PC and PSW

Exception		Vector	Site for Saving the Values in the PC and PSW
Undefined instruction exception		Exception vector table (EXTB)	Stack
Privileged instruction exception		Exception vector table (EXTB)	Stack
Access exception		Exception vector table (EXTB)	Stack
Address exception		Exception vector table (EXTB)	Stack
Single-precision floating-point exception		Exception vector table (EXTB)	Stack
Reset		Exception vector table (EXTB)	Nowhere
Non-maskable interrupt		Exception vector table (EXTB)	Stack
Interrupt	Fast interrupt	FINTV	BPC and BPSW
	Other than above	Interrupt vector table (INTB)	Stack
Unconditional trap		Interrupt vector table (INTB)	Stack

14.4 Hardware Processing for Accepting and Returning from Exceptions

This section describes the hardware processing for accepting and returning from exceptions other than a reset.

(1) Hardware Pre-Processing for Accepting an Exception

(a) Saving PSW

- For a fast interrupt
  PSW → BPSW
- For exceptions other than a fast interrupt
  PSW → Stack

Note: The FPSW is not saved by the hardware pre-processing. If single-precision floating-point operation instructions are used within an exception-handling routine, the user must save the FPSW 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

(c) Clearing the LI flag

14.5 Hardware Pre-Processing

The hardware pre-processing from reception of each exception request to execution of the associated exception handling routine are explained below.

14.5.1 Undefined Instruction Exception

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from the value of EXTB + address 0000 005Ch.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.2 Privileged Instruction Exception

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from the value of EXTB + address 0000 0050h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.3 Access Exceptions

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from the value of EXTB + address 0000 0054h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.4 Address Exceptions

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from the value of EXTB + address 0000 0060h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.5 Single-Precision Floating-Point Exception

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from the value of EXTB + address 0000 0064h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.6 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.

14.5.7 Non-Maskable Interrupt

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. If the interrupt was generated during the execution of an RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, or SWHILE instruction, the value of the program counter (PC) for that instruction is saved on the stack (ISP). For other instructions, the PC value of the next instruction is saved.
4. The processor interrupt priority level bits (IPL[3:0]) in PSW are set to Fh.
5. The vector is fetched from the value of EXTB + address 0000 0078h.
6. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.8 Interrupt

1. The value of the processor status word (PSW) is saved on the stack (ISP) or, for the fast interrupt, in the backup PSW (BPSW).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. If the interrupt was generated during the execution of an RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, or SWHILE instruction, the value of the program counter (PC) for that instruction is saved. For other instructions, the PC value of the next instruction is saved. Saving of the PC is in the backup PC (BPC) for fast interrupts.
4. The processor interrupt priority level bits (IPL[3:0]) in PSW indicate the interrupt priority level of the interrupt.
5. The vector for an interrupt source other than the fast interrupt is fetched from the interrupt vector table. For the fast interrupt, the address is fetched from the fast interrupt vector register (FINTV).
6. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.5.9 Unconditional Trap

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) for the next instruction is saved on the stack (ISP).
4. For the INT instruction, the value at the vector corresponding to the INT instruction number is fetched from the interrupt vector table.
   For the BRK instruction, the value at the vector from the start address is fetched from the interrupt vector table.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

14.6 Return from Exception Handling Routine

Executing the instruction listed in Table 14.3 at the end of the corresponding exception handling routine leads to restoration of the values of the program counter (PC) and processor status word (PSW) that were saved on the stack or in the control registers (BPC and BPSW) immediately before the exception handling sequence.

Table 14.3 Return from Exception Handling Routine

Exception		Instruction for Return
Undefined instruction exception		RTE
Privileged instruction exception		RTE
Access exception		RTE
Address exception		RTE
Single-precision floating-point exception		RTE
Reset		Return is impossible
Non-maskable interrupt		Prohibited
Interrupt	Fast interrupt	RTFI
	Other than above	RTE
Unconditional trap		RTE

14.7 Priority of Exception Events

The priority of exception events is listed in Table 14.4. When multiple exceptions are generated at the same time, the exception with the highest priority is accepted first.

Table 14.4 Priority of Exception Events

Priority		Exception Event
High	1	Reset
↑	2	Non-maskable interrupt
	3	Interrupt
	4	Instruction access exception
	5	Undefined instruction exception Privileged instruction exception
	6	Unconditional trap
	7	Address exception
	8	Operand access exception
Low	9	Single-precision floating-point exception

14.8 Exception Generated in Coprocessor

This section describes exceptions generated in the coprocessor and handling of them.

14.8.1 Double-Precision Floating-Point Exceptions

Double-precision floating-point exceptions are generated when any of the five exceptions specified in the IEEE 754 standard, namely overflow, underflow, inexact, division-by-zero, or invalid operation, or an attempt to use processing that is not implemented, is detected upon execution of a double-precision floating-point operation instruction. When a double-precision floating-point exception has been generated, the exception processing proceeds as the sending of an interrupt request to the interrupt controller without exception handling by the CPU. For the five exceptions, an interrupt request only proceeds when the given bit among the DEX, DEU, DEZ, DEO, or DEV bits in the DPSW is 1.
For the vector numbers allocated to the interrupt controller, refer to section 15, Interrupt Controller (ICUE).

15. Interrupt Controller (ICUE)

15.1 Overview

The interrupt controller (ICU) controls various interrupt requests from the peripheral modules and the IRQi pin (i = 0 to 15), and generates an interrupt request to the CPU and a transfer request to the DTC and DMAC.
Table 15.1 lists the ICU specifications, and Figure 15.1 shows a block diagram of the interrupt controller.

Table 15.1 ICU Specifications (1/2)

Item		Description

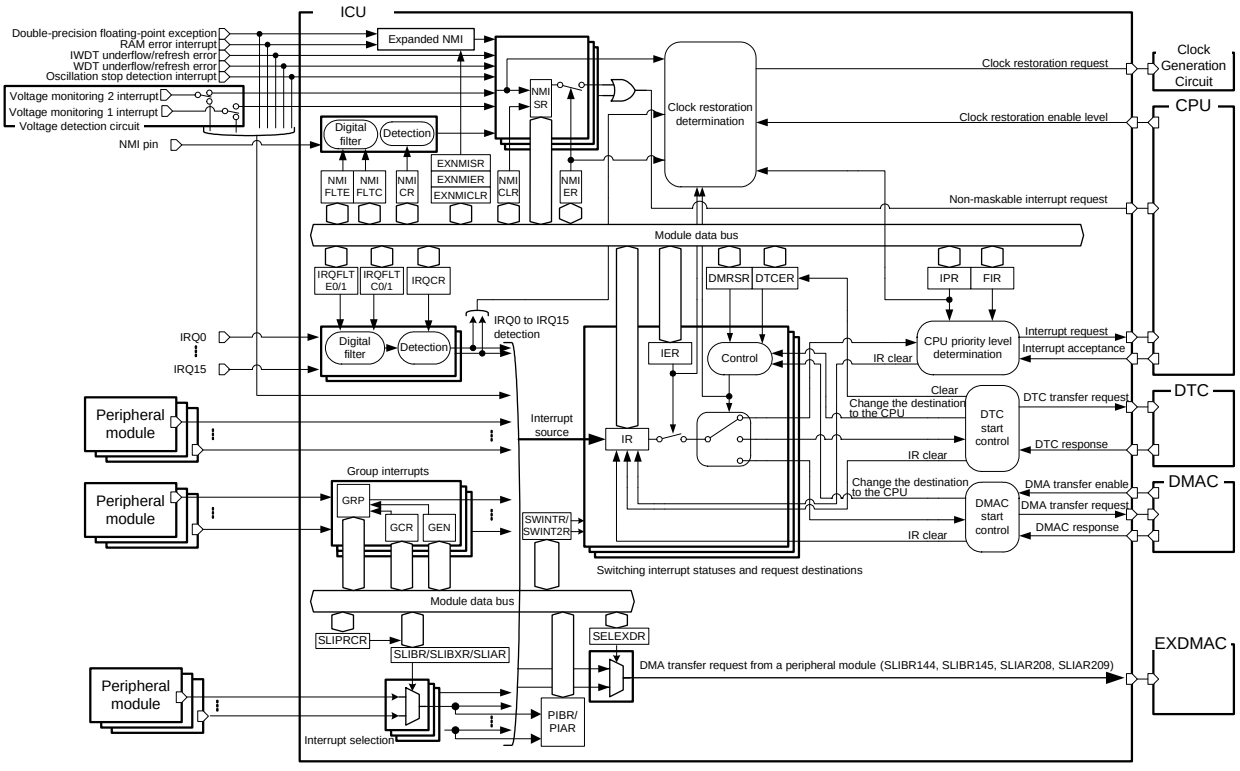
Interrupts	Peripheral interrupts	Interrupts from peripheral modules
		• Interrupt detection method: Edge detection/level detection (fixed for each interrupt source)
		• Group interrupt: Multiple interrupt sources are grouped together and treated as an interrupt source.*1
		  • Group IE0 interrupt: Interrupt sources of coprocessors that use ICLK as the operating clock (edge detection)
		  • Group BE0 interrupt: Interrupt sources of peripheral modules that use PCLKB as the operating clock (edge detection)
		  • Group BL0/BL1/BL2 interrupt: Interrupt sources of peripheral modules that use PCLKB as the operating clock (level detection)
		  • Group AL0/AL1 interrupt: Interrupt sources of peripheral modules that use PCLKA as the operating clock (level detection)
		• Software configurable interrupt B: Any of the interrupt sources for peripheral modules that use PCLKB as the operating clock can be assigned to interrupt vector numbers 128 to 207.
		• Software configurable interrupt A: Any of the interrupt sources for peripheral modules that use PCLKA as the operating clock can be assigned to interrupt vector numbers 208 to 255.

	External pin interrupt	Interrupt by the input signal to the IRQi pin (i = 0 to 15) • Interrupt detection method: Detection of low level, falling edge, rising edge, rising and falling edges One of these detection methods can be set for each source. • Digital filter can be used to remove noise.
	Software interrupt	• Interrupt request can be generated by writing to a register. • Two interrupt sources
	Interrupt priority	Priority level can be set with interrupt source priority register r (IPRr) (r = 000 to 255).
	Fast interrupt function	CPU interrupt response time can be reduced. This function can be used for only one interrupt source.
	DTC/DMAC control	Interrupt sources can be used to start the DTC and DMAC.*2
	EXDMAC control	Interrupt selected by software configurable interrupt B source select register 144 or software configurable interrupt A source select register 208 can be used to start EXDMAC0. Interrupt selected by software configurable interrupt B source select register 145 or software configurable interrupt A source select register 209 can be used to start EXDMAC1.
Non-maskable interrupts *3	NMI pin interrupt	Interrupt by the input signal to the NMI pin • Interrupt detection: Falling edge/rising edge • Digital filter can be used to remove noise.
	Oscillation stop detection interrupt *4	This interrupt occurs when the main clock oscillator stop is detected.
	WDT underflow/refresh error interrupt *4	This interrupt occurs when the watchdog timer (WDT) underflows or a refresh error occurs.
	IWDT underflow/refresh error interrupt *4	This interrupt occurs when the independent watchdog timer (IWDT) underflows or a refresh error occurs.
	Voltage monitoring 1 interrupt *4	Interrupt from voltage detection circuit 1 (LVD1)
	Voltage monitoring 2 interrupt *4	Interrupt from voltage detection circuit 2 (LVD2)
	RAM error interrupt *4	This interrupt occurs when a parity check error is detected in the RAM.
	Double-precision floating-point exceptions *4	Exceptions from double-precision floating-point coprocessor

Table 15.1 ICU Specifications (2/2)

Item		Description
Return from low power consumption states	Sleep mode	• Exit sleep mode by any interrupt source.
	All-module clock stop mode	• Exit all-module clock stop mode by the NMI pin interrupt, external pin interrupt, or peripheral interrupt (voltage monitoring 1, voltage monitoring 2, oscillation stop detection interrupt, USB0 resume, RTC alarm, RTC period, IWDT, VBATT tamper detection, REMC interrupt, software configurable interrupt 146 to 157).
	Software standby mode	• Exit software standby mode by the NMI pin interrupt, external pin interrupt, or peripheral interrupt (voltage monitoring 1, voltage monitoring 2, USB0 resume, RTC alarm, RTC period, IWDT, VBATT tamper detection, REMC interrupt).
	Deep software standby mode	• Exit deep software standby mode by the NMI pin interrupt, specific external pin interrupt, or peripheral interrupt (voltage monitoring 1, voltage monitoring 2, USB0 resume, RTC alarm, RTC period, VBATT tamper detection, REMC interrupt).

Note 1. Groups to which no interrupt source is assigned are reserved. Also, there is no register corresponding to that group.
Note 2. For the DTC and DMAC triggers, refer to Table 15.5, Interrupt Vector Table.
Note 3. Once non-maskable interrupts are enabled, they cannot be disabled.
Note 4. Each source for these non-maskable interrupts can be used for maskable interrupts. When using for maskable interrupts, do not change the NMIER and EXNMIER register values from the value after reset. To enable the voltage monitoring 1 interrupt, set the LVD1CR1.LVD1IRQSEL bit to 1, and to enable the voltage monitoring 2 interrupt, set the LVD2CR1.LVD2IRQSEL bit to 1.



NMIER:	Non-maskable interrupt enable register	IPR:	Interrupt source priority register	SLIBXR:	Software configurable interrupt B source select register X
NMICR:	NMI pin interrupt control register	FIR:	Fast interrupt set register	SLIAR:	Software configurable interrupt A source select register
NMICLR:	Non-maskable interrupt status clear register	DTCER:	DTC transfer request enable register	PIBR:	Software configurable interrupt B request register
NMISR:	Non-maskable interrupt status register	DMRSR:	DMAC trigger select register	PIAR:	Software configurable interrupt A request register
EXNMIER:	Expanded non-maskable interrupt enable register	IRQFLTE0/1:	IRQ pin digital filter enable registers	SLIPRCR:	Software configurable interrupt source select register write protect register
EXNMICLR:	Expanded non-maskable interrupt status clear register	IRQFLTC0/1:	IRQ pin digital filter setting registers	SELEXDR:	EXDMAC trigger select register
EXNMISR:	Expanded non-maskable interrupt status register	NMIFLTE:	NMI pin digital filter enable register	SWINTR:	Software interrupt generation register
IRQCR:	IRQ control register	NMIFLTC:	NMI pin digital filter setting register	SWINT2R:	Software Interrupt 2 generation register
IR:	Interrupt request register	GRP:	Group interrupt request register*1		
IER:	Interrupt request enable register	GEN:	Group interrupt request enable register*1		
		GCR:	Group interrupt clear register*1		
		SLIBR:	Software configurable interrupt B source select register		

Note 1. There is no register in the group to which no interrupt source is assigned.

Figure 15.1 Block Diagram of the ICU

Table 15.2 lists I/O pins used for the ICU.

Table 15.2 ICU I/O Pins

Pin Name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQ0 to IRQ15	Input	External interrupt request pins

15.2 Register Descriptions

15.2.1 Interrupt Request Register n (IRn) (n = 016 to 255)

Address(es): ICU.IR016 0008 7010h to ICU.IR255 0008 70FFh

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

Bit	Symbol	Bit Name	Description	R/W
b0	IR	Interrupt Status Flag	0: No interrupt request is generated. 1: An interrupt request is generated.	R/(W) *1
b7 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W

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

The IRn register indicates whether an interrupt request has been generated.
This register is provided for each interrupt vector number, and n matches the interrupt vector number.
For the correspondence between interrupt sources and interrupt vector numbers, refer to Table 15.5, Interrupt Vector Table.

IR Flag (Interrupt Status Flag)

The IR flag is a status flag indicating whether an interrupt request has been generated. This flag becomes 1 when an interrupt request is generated. To detect an interrupt request, set the interrupt enable bit of the peripheral module to enable output of the interrupt request.
An interrupt request can be detected by edge detection or level detection. For interrupts from peripheral modules, the detection method (edge detection or level detection) is determined depending on the source. Refer to Table 15.5, Interrupt Vector Table for details on the detection method for each source. For interrupts from the IRQi pin (i = 0 to 15), edge detection or level detection can be selected by setting the IRQCRi.IRQMD[1:0] bits.
The interrupt status flag for group interrupts is the ISj flag (j = 0 to 31) in the group interrupt request register (GRPIE0, GRPBE0, GRPBL0, GRPBL1, GRPAL0, GRPAL1). When any of the ISj flags becomes 1, the IRn.IR flag corresponding to each group interrupt becomes 1. Group interrupts are detected by level detection.
Refer to section 15.4.4, Group Interrupts for details on group interrupts.

(1) Edge detection

This flag becomes 1 under the following condition:

- The IR flag becomes 1 when an interrupt request for peripheral interrupts or external pin interrupts is generated. For interrupt requests for peripheral modules, refer to the corresponding sections.

This flag becomes 0 under any of the following conditions:

- The IR flag becomes 0 when the interrupt request destination accepts an interrupt request.
- The IR flag becomes 0 by writing 0 to the IR flag. Note that when the interrupt request destination is the DTC or DMAC, do not write 0 to the IR flag.

(2) Level detection

This flag becomes 1 under any of the following conditions:

- The IR flag is 1 while an interrupt request for peripheral interrupts or external pin interrupts is generated. For interrupt requests for peripheral modules, refer to the corresponding sections.
- For group interrupts, the IR flag becomes 1 when the ISj flag in the group interrupt request register (GRPIE0, GRPBE0, GRPBL0, GRPBL1, GRPAL0, GRPAL1) is 1 (interrupt request is generated) while the ENj bit in the group interrupt request enable register (GENIE0, GENBE0, GENBL0, GENBL1, GENAL0, GENAL1) is 1 (enabled) (j = 0 to 31).

This flag becomes 0 under any of the following conditions:

- The IR flag becomes 0 by clearing output of the peripheral module interrupt request. The IR flag does not become 0 when the interrupt request destination accepts the interrupt request. For interrupt requests for peripheral modules, refer to the corresponding sections.
- For group interrupts, the IR flag becomes 0 when the ENj bit in the group interrupt request enable register is 0 (disabled) or when the ISj flag in the group interrupt request register is 0 (interrupt request is not generated).

When level detection is selected for detecting external pin interrupts, set the input level of the IRQi pin to high (i = 0 to 15) to cancel the external pin interrupt that has occurred. When level detection is selected, do not write to the IR flag.

15.2.2 Interrupt Request Enable Register m (IERm) (m = 02h to 1Fh)

Address(es): ICU.IER02 0008 7202h to ICU.IER1F 0008 721Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IEN0	Interrupt Request Enable 0	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b1	IEN1	Interrupt Request Enable 1		R/W
b2	IEN2	Interrupt Request Enable 2		R/W
b3	IEN3	Interrupt Request Enable 3		R/W
b4	IEN4	Interrupt Request Enable 4		R/W
b5	IEN5	Interrupt Request Enable 5		R/W
b6	IEN6	Interrupt Request Enable 6		R/W
b7	IEN7	Interrupt Request Enable 7		R/W

Note: When the interrupt source of the interrupt vector number is reserved, set the corresponding bit to 0. The read value is 0.

The IERm register enables or disables output of the interrupt request to the interrupt request destination.

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

When the IENj bit is 1, an interrupt request is output to the destination. When the IENj bit is 0, an interrupt request is not output to the destination.

The IRn.IR flag (n = 016 to 255) is not affected by the IENj bit setting. Even when the IENj bit is 0, the IR flag changes according to the conditions described in section 15.2.1, Interrupt Request Register n (IRn) (n = 016 to 255).

The IERm.IENj bit is provided for each interrupt vector number.

Refer to Table 15.5, Interrupt Vector Table for the correspondence between interrupt sources and the IERm.IENj bit.

Note that m and j can be calculated by the following formula:

m = quotient when n divided by 8

j = remainder when n divided by 8

Refer to section 15.7.3.1, Interrupt Request Destination Setting Procedure for the procedure to set the IERm.IENj bit to select the interrupt request destination.

15.2.3 Interrupt Source Priority Register r (IPRr) (r = 000 to 255)

Address(es): ICU.IPR000 0008 7300h to ICU.IPR255 0008 73FFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	IPR[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IPR[3:0]	Interrupt Priority Level Select	b3 b0 0 0 0 0: Level 0 (interrupt disabled) *1 0 0 0 1: Level 1 0 0 1 0: Level 2 0 0 1 1: Level 3 0 1 0 0: Level 4 0 1 0 1: Level 5 0 1 1 0: Level 6 0 1 1 1: Level 7 1 0 0 0: Level 8 1 0 0 1: Level 9 1 0 1 0: Level 10 1 0 1 1: Level 11 1 1 0 0: Level 12 1 1 0 1: Level 13 1 1 1 0: Level 14 1 1 1 1: Level 15 (highest)	R/W
b7 to b4	—	Reserved	The read value is 0. The write value should be 0.	R/W

Note 1. For an interrupt source of a fast interrupt, even when the IPR[3:0] bits are set to level 0, the priority level is level 15.

The IPRr register sets the interrupt priority level of an interrupt source that is assigned to the corresponding interrupt vector number.

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

The IPR[3:0] bits select the interrupt priority level of the corresponding interrupt source.
The priority level selected by the IPR[3:0] bits is used for only determining the priority level of interrupt requests to the CPU. It does not affect transfer requests to the DTC and DMAC.
The CPU accepts only interrupt requests that have the higher priority level than the processor interrupt priority level indicated by the PSW.IPL[3:0] bits.
When multiple interrupt requests are concurrently generated, the priority levels selected by the corresponding IPR[3:0] bits are compared. When multiple interrupt requests that have the same priority level are concurrently generated, the interrupt request that has the smallest interrupt vector number has priority.
Write to this register while the corresponding IERm.IENj bit is 0 (interrupt request is disabled) (m = 02h to 1Fh; j = 0 to 7).
Refer to Table 15.5, Interrupt Vector Table for the correspondence between interrupt vectors and the IPRr register. Note that r matches the vector number when the interrupt vector number is 32 or greater.

15.2.4 Fast Interrupt Set Register (FIR)

Address(es): ICU.FIR 0008 72F0h

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

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	FVCT[7:0]	Fast Interrupt Vector Number	Set the vector number of an interrupt source that is assigned to a fast interrupt.	R/W
b14 to b8	—	Reserved	The read value is 0. The write value should be 0.	R/W
b15	FIEN	Fast Interrupt Enable	0: Fast interrupt is disabled 1: Fast interrupt is enabled	R/W

The FIR register sets an interrupt source that is handled as the fast interrupt.
The fast interrupt is enabled only when the destination is the CPU. When the destination is the DTC or DMAC, the DTC or DMA transfer request is not affected by setting the interrupt vector number as the fast interrupt.
Write to this register while the corresponding IERm.IENj bit is 0 (m = 02h to 1Fh; j = 0 to 7).
Refer to section 15.9, Fast interrupt for details on the fast interrupt.

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

The FVCT[7:0] bits set the interrupt vector number of an interrupt source for the fast interrupt.
Refer to Table 15.5, Interrupt Vector Table for interrupt vector numbers that can be set in the FVCT[7:0] bits. Do not set an interrupt vector number that is reserved.

FIEN Bit (Fast Interrupt Enable)

The FIEN bit enables the fast interrupt to be used.
When the FIEN bit is 1, the interrupt source that is assigned to the interrupt vector number set by the FVCT[7:0] bits is handled as the fast interrupt.
When an interrupt request of the interrupt vector number set by the FVCT[7:0] bits is generated to the CPU while the FIEN bit is 1, an interrupt request is output to the CPU as the fast interrupt regardless of the IPRr register setting (r = 000 to 255). Note that the IPRr register setting is required when using the fast interrupt to exit software standby mode. Refer to section 15.10.3, Exiting Software Standby Mode for details.

15.2.5 Software Interrupt Generation Register (SWINTR)

Address(es): ICU.SWINTR 0008 72E0h

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

Bit	Symbol	Bit Name	Description	R/W
b0	SWINT	Software Interrupt Generation	The read value is 0. When writing 1 to this bit, a software interrupt request is generated. Writing 0 to this bit has no effect.	R/W
b7 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W

The SWINTR register controls generation of a software interrupt request.

SWINT Bit (Software Interrupt Generation)

When the SWINT bit is set to 1, a software interrupt request (SWINT) is generated, and the IR027.IR flag becomes 1. A software interrupt request (SWINT) can be set as a DTC trigger, but it cannot be set as a DMAC trigger.

15.2.6 Software Interrupt 2 Generation Register (SWINT2R)

Address(es): ICU.SWINT2R 0008 72E1h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SWINT2
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SWINT2	Software Interrupt 2 Generation	The read value is 0. When writing 1 to this bit, a software interrupt request 2 is generated. Writing 0 to this bit has no effect.	R/W
b7 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W

The SWINT2R register controls generation of software interrupt request 2.

SWINT2 Bit (Software Interrupt 2 Generation)

When the SWINT2 bit is set to 1, software interrupt request 2 (SWINT2) is generated, and the IR026.IR flag becomes 1. Software interrupt request 2 (SWINT2) can be set as the DTC trigger, but it cannot be set as the DMAC trigger.

15.2.7 DTC Transfer Request Enable Register n (DTCERn) (n = 026 to 255)

Address(es): ICU.DTCER026 0008 711Ah to ICU.DTCER255 0008 71FFh

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

Bit	Symbol	Bit Name	Description	R/W
b0	DTCE	DTC Transfer Request Enable	0: The corresponding interrupt source is selected as an interrupt request to the CPU or as the DMAC trigger. 1: The corresponding interrupt source is selected as the DTC trigger.	R/W
b7 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W

The DTCERn register selects the interrupt source corresponding to interrupt vector number n as the DTC trigger.
Do not set the same interrupt source for both the DTC trigger and DMAC trigger. Refer to Table 15.5, Interrupt Vector Table for the correspondence between interrupt sources and interrupt vector numbers and interrupt sources that can be used as the DTC trigger.

DTCE Bit (DTC Transfer Request Enable)

When the DTCE bit is set to 1, the corresponding interrupt source is selected as the DTC trigger.
This bit becomes 1 under the following condition:

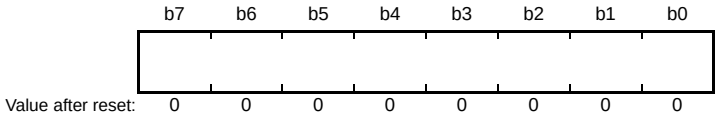
- 1 is written to the DTCE bit

This bit becomes 0 under any of the following conditions:

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

15.2.8 DMAC Trigger Select Register m (DMRSRm) (m = DMAC channel number)

Address(es): ICU.DMRSR0 0008 7400h, ICU.DMRSR1 0008 7404h, ICU.DMRSR2 0008 7408h, ICU.DMRSR3 0008 740Ch, ICU.DMRSR4 0008 7410h, ICU.DMRSR5 0008 7414h, ICU.DMRSR6 0008 7418h, ICU.DMRSR7 0008 741Ch



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	—	These bits set the interrupt vector number of the interrupt source as the DMAC trigger.	R/W

The DMRSRm register sets an interrupt source as the DMACm trigger.
Do not set the same vector number for multiple DMRSRm registers. Do not set the same interrupt source for both the DTC trigger and DMAC trigger. Otherwise, the operation is not guaranteed.
Set the interrupt vector number of an interrupt source used as the DMAC trigger in the DMRSRm register. Do not set vector numbers of interrupt sources that cannot be used as the DMAC trigger.
Refer to Table 15.5, Interrupt Vector Table for interrupt vector numbers of interrupt sources.
Write the DMRSRm register while the DMACm.DMCNT.DTE bit is 0.

15.2.9 IRQ Control Register i (IRQCRi) (i = 0 to 15)

Address(es): ICU.IRQCR0 0008 7500h to ICU.IRQCR15 0008 750Fh

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	The read value is 0. The write value should be 0.	R/W
b3, b2	IRQMD[1:0]	IRQ Detection Select	b3 b2 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Rising and falling edges	R/W
b7 to b4	—	Reserved	The read value is 0. The write value should be 0.	R/W

The IRQCRi register selects the detection method for external pin interrupts.
Write to this register while the corresponding IERm.IENj bit is 0 (m = 02h to 1Fh; j = 0 to 7). After writing to this register, set the IRn.IR flag to 0, and then set the IENj bit to 1 (n = 016 to 255). Note that setting the IR flag to 0 is not required when changing the detection method to level detection.

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

The IRQMD[1:0] bits set the detection method for the IRQi pin interrupt (i = 0 to 15).
Refer to section 15.7.4, Setting the External Pin Interrupt for the procedure to set the external pin interrupts.

15.2.10 IRQ Pin Digital Filter Enable Register 0 (IRQFLTE0)

Address(es): ICU.IRQFLTE0 0008 7520h

	b7	b6	b5	b4	b3	b2	b1	b0
	FLTEN7	FLTEN6	FLTEN5	FLTEN4	FLTEN3	FLTEN2	FLTEN1	FLTEN0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FLTEN0	IRQ0 Digital Filter Enable	0: Digital filter is disabled. 1: Digital filter is enabled.	R/W
b1	FLTEN1	IRQ1 Digital Filter Enable		R/W
b2	FLTEN2	IRQ2 Digital Filter Enable		R/W
b3	FLTEN3	IRQ3 Digital Filter Enable		R/W
b4	FLTEN4	IRQ4 Digital Filter Enable		R/W
b5	FLTEN5	IRQ5 Digital Filter Enable		R/W
b6	FLTEN6	IRQ6 Digital Filter Enable		R/W
b7	FLTEN7	IRQ7 Digital Filter Enable		R/W

The IRQFLTE0 register enables and disables digital filters for pins IRQ0 to IRQ7.

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

When the FLTENi bit is 1, the digital filter for the IRQi pin is enabled. When the FLTENi bit is 0, the digital filter for the IRQi pin is disabled.

The signal input to the IRQi pin is sampled at the sampling clock set by the IRQFLTC0.FCLKSELi[1:0] bits. When the sampled signal level does not match three times in a row, the input signal is removed.

Refer to section 15.7.6, Digital Filter for details on the digital filter.

15.2.11 IRQ Pin Digital Filter Enable Register 1 (IRQFLTE1)

Address(es): ICU.IRQFLTE1 0008 7521h

	b7	b6	b5	b4	b3	b2	b1	b0
	FLTEN15	FLTEN14	FLTEN13	FLTEN12	FLTEN11	FLTEN10	FLTEN9	FLTEN8
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FLTEN8	IRQ8 Digital Filter Enable	0: Digital filter is disabled. 1: Digital filter is enabled.	R/W
b1	FLTEN9	IRQ9 Digital Filter Enable		R/W
b2	FLTEN10	IRQ10 Digital Filter Enable		R/W
b3	FLTEN11	IRQ11 Digital Filter Enable		R/W
b4	FLTEN12	IRQ12 Digital Filter Enable		R/W
b5	FLTEN13	IRQ13 Digital Filter Enable		R/W
b6	FLTEN14	IRQ14 Digital Filter Enable		R/W
b7	FLTEN15	IRQ15 Digital Filter Enable		R/W

The IRQFLTE1 register enables or disables digital filters for pins IRQ8 to IRQ15.

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

When the FLTENi bit is 1, the digital filter for the IRQi pin is enabled. When the FLTENi bit is 0, the digital filter for the IRQi pin is disabled.

The signal input to the IRQi pin is sampled at the sampling clock set by the IRQFLTC1.FCLKSELi[1:0] bits. When the sampled signal level does not match three times in a row, the input signal is removed.

Refer to section 15.7.6, Digital Filter for details on the digital filter.

15.2.12 IRQ Pin Digital Filter Setting Register 0 (IRQFLTC0)

Address(es): ICU.IRQFLTC0 0008 7528h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FCLKSEL7[1:0]		FCLKSEL6[1:0]		FCLKSEL5[1:0]		FCLKSEL4[1:0]		FCLKSEL3[1:0]		FCLKSEL2[1:0]		FCLKSEL1[1:0]		FCLKSEL0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	FCLKSEL0[1:0]	IRQ0 Digital Filter Sampling Clock	0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64	R/W
b3, b2	FCLKSEL1[1:0]	IRQ1 Digital Filter Sampling Clock		R/W
b5, b4	FCLKSEL2[1:0]	IRQ2 Digital Filter Sampling Clock		R/W
b7, b6	FCLKSEL3[1:0]	IRQ3 Digital Filter Sampling Clock		R/W
b9, b8	FCLKSEL4[1:0]	IRQ4 Digital Filter Sampling Clock		R/W
b11, b10	FCLKSEL5[1:0]	IRQ5 Digital Filter Sampling Clock		R/W
b13, b12	FCLKSEL6[1:0]	IRQ6 Digital Filter Sampling Clock		R/W
b15, b14	FCLKSEL7[1:0]	IRQ7 Digital Filter Sampling Clock		R/W

The IRQFLTC0 register sets the sampling clock of the digital filter for pins IRQ0 to IRQ7.

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

The FCLKSELi[1:0] bits set the sampling clock of the digital filter for the IRQi pin.

Refer to section 15.7.6, Digital Filter for details on the digital filter.

15.2.13 IRQ Pin Digital Filter Setting Register 1 (IRQFLTC1)

Address(es): ICU.IRQFLTC1 0008 752Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FCLKSEL15[1:0]		FCLKSEL14[1:0]		FCLKSEL13[1:0]		FCLKSEL12[1:0]		FCLKSEL11[1:0]		FCLKSEL10[1:0]		FCLKSEL9[1:0]		FCLKSEL8[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	FCLKSEL8[1:0]	IRQ8 Digital Filter Sampling Clock	0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64	R/W
b3, b2	FCLKSEL9[1:0]	IRQ9 Digital Filter Sampling Clock		R/W
b5, b4	FCLKSEL10[1:0]	IRQ10 Digital Filter Sampling Clock		R/W
b7, b6	FCLKSEL11[1:0]	IRQ11 Digital Filter Sampling Clock		R/W
b9, b8	FCLKSEL12[1:0]	IRQ12 Digital Filter Sampling Clock		R/W
b11, b10	FCLKSEL13[1:0]	IRQ13 Digital Filter Sampling Clock		R/W
b13, b12	FCLKSEL14[1:0]	IRQ14 Digital Filter Sampling Clock		R/W
b15, b14	FCLKSEL15[1:0]	IRQ15 Digital Filter Sampling Clock		R/W

The IRQFLTC1 register sets the sampling clock of the digital filter for pins IRQ8 to IRQ15.

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

The FCLKSELi[1:0] bits set the sampling clock of the digital filter for the IRQi pin.

Refer to section 15.7.6, Digital Filter for details on the digital filter.

15.2.14 Non-Maskable Interrupt Status Register (NMISR)

Address(es): ICU.NMISR 0008 7580h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	EXNMIST	LVD2ST	LVD1ST	IWDTST	WDTST	OSTST	NMIST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NMIST	NMI Status Flag	0: NMI pin interrupt is not requested. 1: NMI pin interrupt is requested.	R
b1	OSTST	Oscillation Stop Detection Interrupt Status Flag	0: Oscillation stop detection interrupt is not requested. 1: Oscillation stop detection interrupt is requested.	R
b2	WDTST	WDT Underflow/Refresh Error Status Flag	0: WDT underflow/refresh error interrupt is not requested. 1: WDT underflow/refresh error interrupt is requested.	R
b3	IWDTST	IWDT Underflow/Refresh Error Status Flag	0: IWDT underflow/refresh error interrupt is not requested. 1: IWDT underflow/refresh error interrupt is requested.	R
b4	LVD1ST	Voltage Monitoring 1 Interrupt Status Flag	0: Voltage monitoring 1 interrupt is not requested. 1: Voltage monitoring 1 interrupt is requested.	R
b5	LVD2ST	Voltage Monitoring 2 Interrupt Status Flag	0: Voltage monitoring 2 interrupt is not requested. 1: Voltage monitoring 2 interrupt is requested.	R
b6	EXNMIST	Expanded Non-Maskable Interrupt Status Flag	0: Expanded non-maskable interrupt is not requested. 1: Expanded non-maskable interrupt is requested.	R
b7	—	Reserved	This bit is read as 0 and cannot be modified.	R

The NMISR register indicates whether a non-maskable interrupt request has occurred.
Each flag in the NMISR register is not affected by the setting of the corresponding bit in the NMIER register.
In the non-maskable interrupt handler, read the NMISR register to check if the other non-maskable interrupt has occurred. Confirm that all the status flags are 0 before exiting the interrupt handler.

NMIST Flag (NMI Status Flag)

The NMIST flag indicates whether an NMI pin interrupt request has been generated.
This flag is read-only. To set the NMIST flag to 0, set the NMICLR.NMICLR bit to 1.
This flag becomes 1 under the following condition:

- An edge set in the NMICR.NMIMD bit is input to the NMI pin

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.NMICLR bit

OSTST Flag (Oscillation Stop Detection Interrupt Status Flag)

The OSTST flag indicates whether an oscillation stop detection interrupt request has been generated.
The OSTST flag is read-only. To set the OSTST flag to 0, set the NMICLR.OSTCLR bit to 1.
This flag becomes 1 under the following condition:

- An oscillation stop detection interrupt occurs

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.OSTCLR bit

WDTST Flag (WDT Underflow/Refresh Error Status Flag)

The WDTST flag indicates whether an WDT underflow/refresh error interrupt request is generated.
The WDTST flag is read-only. To set the WDTST flag to 0, set the NMICLR.WDTCLR bit to 1.
This flag becomes 1 under the following condition:

- A WDT underflow/refresh error interrupt occurs while the WDTRCR.RSTIRQS bit is 0

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.WDTCLR bit

IWDTST Flag (IWDT Underflow/Refresh Error Status Flag)

The IWDTST flag indicates whether an IWDT underflow/refresh error interrupt request is generated.
The IWDTST flag is read-only. To set the IWDTST flag to 0, set the NMICLR.IWDTCLR bit to 1.
This flag becomes 1 under the following condition:

- An IWDT underflow/refresh error interrupt occurs while the IWDTRCR.RSTIRQS bit is 0

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.IWDTCLR bit

LVD1ST Flag (Voltage Monitoring 1 Interrupt Status Flag)

The LVD1ST flag indicates whether a voltage monitoring 1 interrupt request is generated.
The LVD1ST flag is read-only. To set the LVD1ST flag to 0, set the NMICLR.LVD1CLR bit to 1.
This flag becomes 1 under the following condition:

- A voltage monitoring 1 interrupt occurs while the LVD1CR1.LVD1IRQSEL bit is 0

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.LVD1CLR bit

LVD2ST Flag (Voltage Monitoring 2 Interrupt Status Flag)

The LVD2ST flag indicates whether a voltage monitoring 2 interrupt request is generated.
The LVD2ST flag is read-only. To set the LVD2ST flag to 0, set the NMICLR.LVD2CLR bit to 1.
This flag becomes 1 under the following condition:

- A voltage monitoring 2 interrupt occurs while the LVD2CR1.LVD2IRQSEL bit is 0

This flag becomes 0 under the following condition:

- 1 is written to the NMICLR.LVD2CLR bit

EXNMIST Flag (Expanded Non-Maskable Interrupt Status Flag)

The EXNMIST flag indicates whether an expanded non-maskable interrupt request is generated.
The EXNMIST flag is read-only. To set the EXNMIST flag to 0, clear all error status flags of the expanded non-maskable interrupt request sources.
This flag becomes 1 under the following condition:

- When any of the flag in the EXNMISR register is 1 and the corresponding bit in the EXNMIER register is 1

This flag becomes 0 under the following condition:

- When all sources which set the EXNMIST flag to 1 are cleared.

15.2.15 Non-Maskable Interrupt Enable Register (NMIER)

Address(es): ICU.NMIER 0008 7581h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	EXNMIEN	LVD2EN	LVD1EN	IWDTEN	WDTEN	OSTEN	NMIEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NMIEN	NMI Pin Interrupt Enable	0: NMI pin interrupt is disabled. 1: NMI pin interrupt is enabled.	R/(W) *1
b1	OSTEN	Oscillation Stop Detection Interrupt Enable	0: Oscillation stop detection interrupt is disabled. 1: Oscillation stop detection interrupt is enabled.	R/(W) *1
b2	WDTEN	WDT Underflow/Refresh Error Enable	0: WDT underflow/refresh error interrupt is disabled. 1: WDT underflow/refresh error interrupt is enabled.	R/(W) *1
b3	IWDTEN	IWDT Underflow/Refresh Error Enable	0: IWDT underflow/refresh error interrupt is disabled. 1: IWDT underflow/refresh error interrupt is enabled.	R/(W) *1
b4	LVD1EN	Voltage Monitoring 1 Interrupt Enable	0: Voltage monitoring 1 interrupt is disabled. 1: Voltage monitoring 1 interrupt is enabled.	R/(W) *1
b5	LVD2EN	Voltage Monitoring 2 Interrupt Enable	0: Voltage monitoring 2 interrupt is disabled. 1: Voltage monitoring 2 interrupt is enabled.	R/(W) *1
b6	EXNMIEN	Expanded Non-Maskable Interrupt Enable	0: Expanded non-maskable interrupt is disabled. 1: Expanded non-maskable interrupt is enabled.	R/(W) *1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Once this bit is set to 1, it cannot be set to 0 by software.

The NMIER register enables and disables generation of non-maskable interrupt requests. When each bit is 1, the corresponding interrupt source is used as a non-maskable interrupt.

NMIEN Bit (NMI Pin Interrupt Enable)

The NMIEN bit enables and disables using the NMI pin interrupt.

OSTEN Bit (Oscillation Stop Detection Interrupt Enable)

The OSTEN bit enables and disables generation of a non-maskable interrupt by the oscillation stop detection interrupt. When the oscillation stop detection interrupt is used as a maskable interrupt, leave this bit set to 0.

WDTEN Bit (WDT Underflow/Refresh Error Enable)

The WDTEN bit enables and disables generation of a non-maskable interrupt by the WDT underflow/refresh error interrupt.
When the WDT underflow/refresh error interrupt is used as a maskable interrupt, leave this bit set to 0.

IWDTEN Bit (IWDT Underflow/Refresh Error Enable)

The IWDTEN bit enables and disables generation of a non-maskable interrupt by the IWDT underflow/refresh error interrupt.
When the IWDT underflow/refresh error interrupt is used as a maskable interrupt, leave this bit set to 0.

LVD1EN Bit (Voltage Monitoring 1 Interrupt Enable)

The LVD1EN bit enables and disables generation of a non-maskable interrupt by the voltage monitoring 1 interrupt.
When the voltage monitoring 1 interrupt is used as a maskable interrupt, leave this bit set to 0.

LVD2EN Bit (Voltage Monitoring 2 Interrupt Enable)

The LVD2EN bit enables and disables generation of a non-maskable interrupt by the voltage monitoring 2 interrupt. When the voltage monitoring 2 interrupt is used as a maskable interrupt, leave this bit set to 0.

EXNMIEN Bit (Expanded Non-Maskable Interrupt Enable)

The EXNMIEN bit enables and disables generation of a non-maskable interrupt by the expanded non-maskable interrupt. When the expanded non-maskable interrupt is used as a maskable interrupt, leave this bit set to 0.

15.2.16 Non-Maskable Interrupt Status Clear Register (NMICLR)

Address(es): ICU.NMICLR 0008 7582h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	LVD2CLR	LVD1CLR	IWDTCLR	WDTCLR	OSTCLR	NMICLR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NMICLR	NMI Clear	The read value is 0. When 1 is written to this bit, the NMISR.NMIST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b1	OSTCLR	OST Clear	The read value is 0. When 1 is written to this bit, the NMISR.OSTST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b2	WDTCLR	WDT Clear	The read value is 0. When 1 is written to this bit, the NMISR.WDTST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b3	IWDTCLR	IWDT Clear	The read value is 0. When 1 is written to this bit, the NMISR.IWDTST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b4	LVD1CLR	LVD1 Clear	The read value is 0. When 1 is written to this bit, the NMISR.LVD1ST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b5	LVD2CLR	LVD2 Clear	The read value is 0. When 1 is written to this bit, the NMISR.LVD2ST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b7-b6	—	Reserved	The read value is 0. The write value should be 0.	R/W

The NMICLR register clears each flag in the NMISR register.
When writing 1 to a bit in this register, the corresponding status flag becomes 0.

15.2.17 NMI Pin Interrupt Control Register (NMICR)

Address(es): ICU.NMICR 0008 7583h

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

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	The read value is 0. The write value should be 0.	R/W
b3	NMIMD	NMI Detection Set	0: Falling edge 1: Rising edge	R/W
b7 to b4	—	Reserved	The read value is 0. The write value should be 0.	R/W

The NMICR register selects a detection method for the NMI pin interrupt.
Write to the NMICR register while the NMIER.NMIEN bit is 0.

15.2.18 Expanded Non-Maskable Interrupt Status Register (EXNMISR)

Address(es): ICU.EXNMISR 0008 7584h

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

Bit	Symbol	Bit Name	Description	R/W
b0	RAMST	RAM Error Interrupt Status Flag	0: RAM error interrupt is not requested. 1: RAM error interrupt is requested.	R
b1	DPFPUST	Double-Precision Floating-Point Exception Status Flag	0: Double-precision floating-point exception is not requested. 1: Double-precision floating-point exception is requested.	R
b7 to b2	—	Reserved	These bits are read as 0 and cannot be modified.	R

The EXNMISR register indicates whether a non-maskable interrupt request assigned to expanded non-maskable interrupt source has occurred. Each flag in the EXNMISR register is not affected by the setting of the corresponding bit in the EXNMIER register.

RAMST Flag (RAM Error Interrupt Status Flag)

The RAMST flag indicates whether an RAM error interrupt request is generated from the RAM.
The RAMST flag is read-only. To set the RAMST flag to 0, clear all of the error status flags of the RAM. Refer to section 53.3.2, RAM Error Interrupt Function for details.
This flag becomes 1 under the following condition:

- A parity check error interrupt occurs (When the RAM.RAMSTS.RAMERR flag becomes 1)

This flag becomes 0 under the following condition:

- When all sources which set the RAMST flag to 1 are cleared.

DPFPUST Flag (Double-Precision Floating-Point Exception Status Flag)

The DPFPUST flag indicates whether an double-precision floating-point exception is generated by double-precision floating-point coprocessor.
This flag becomes 1 under the following condition:

- A double-precision floating-point exception occurs

This flag becomes 0 under the following condition:

- 1 is written to the EXNMICLR.DPFPUCLR bit

15.2.19 Expanded Non-Maskable Interrupt Enable Register (EXNMIER)

Address(es): ICU.EXNMIER 0008 7585h

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

Bit	Symbol	Bit Name	Description	R/W
b0	RAMEN	RAM Error Interrupt Enable	0: RAM error interrupt is disabled. 1: RAM error interrupt is enabled.	R/W
b1	DPFPUEN	Double-Precision Floating-Point Exception Enable	0: Double-precision floating-point exception is disabled. 1: Double-precision floating-point exception is enabled.	R/W
b6 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b7	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

The EXNMIER register enables and disables generation of non-maskable interrupt requests assigned to expanded non-maskable interrupt source. When each bit is 1, the corresponding interrupt source is used as a non-maskable interrupt. Set this register before setting the NMIER.EXNMIEN bit to 1.

RAMEN Bit (RAM Error Interrupt Enable)

The RAMEN bit enables and disables generation of a non-maskable interrupt by the RAM error interrupt from the RAM. When the RAM error interrupt is used as a maskable interrupt, set this bit to 0.

DPFPUEN Bit (Double-Precision Floating-Point Exception Enable)

The DPFPUEN bit enables and disables generation of a non-maskable interrupt by the double-precision floating-point exception from double-precision floating-point coprocessor.
When the double-precision floating-point exception is used as a maskable interrupt, leave this bit set to 0.

15.2.20 Expanded Non-Maskable Interrupt Status Clear Register (EXNMICLR)

Address(es): ICU.EXNMICLR 0008 7586h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	DPFPUCLR	—
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
b1	DPFPUCLR	Double-Precision Floating-Point Exception Clear	The read value is 0. When 1 is written to this bit, the EXNMISR.DPFPUST flag becomes 0. Writing 0 to this bit has no effect.	R/(W)
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

The EXNMICLR register clears each flag in the EXNMISR register.
When writing 1 to a bit in this register, the corresponding status flag becomes 0.

15.2.21 NMI Pin Digital Filter Enable Register (NMIFLTE)

Address(es): ICU.NMIFLTE 0008 7590h

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

Bit	Symbol	Bit Name	Description	R/W
b0	NFLTEN	NMI Digital Filter Enable	0: Digital filter is disabled. 1: Digital filter is enabled.	R/W
b7 to b1	—	Reserved	The read value is 0. The write value should be 0.	R/W

The NMIFLTE register enables or disables the digital filter for the NMI pin.

NFLTEN Bit (NMI Digital Filter Enable)

When the NFLTEN bit is 1, the digital filter is enabled. When the NFLTEN bit is 0, the digital filter is disabled.
The signal input to the NMI pin is sampled at the sampling clock set by the NMIFLTC.NFCLKSEL[1:0] bits. When the sampled signal level does not match three times in a row, the input signal is removed.
Refer to section 15.7.6, Digital Filter for details on the digital filter.

15.2.22 NMI Pin Digital Filter Setting Register (NMIFLTC)

Address(es): ICU.NMIFLTC 0008 7594h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	NFCLKSEL[1:0]	NMI Digital Filter Sampling Clock	b1 b0 0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64	R/W
b7 to b2	—	Reserved	The read value is 0. The write value should be 0.	R/W

The NMIFLTC register sets the sampling clock of the digital filter for the NMI pin.

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

The NFCLKSEL[1:0] bits select the sampling clock of the digital filter for the NMI pin.
Refer to section 15.7.6, Digital Filter for details on the digital filter.

15.2.23 Group IE0 Interrupt Request Register (GRPIE0), Group BE0 Interrupt Request Register (GRPBE0), Group BL0/BL1 Interrupt Request Register (GRPBL0/GRPBL1), Group AL0/AL1 Interrupt Request Register (GRPAL0/GRPAL1)

Address(es): ICU.GRPIE0 0008 75B0h, ICU.GRPBE0 0008 7600h, ICU.GRPBL0 0008 7630h, ICU.GRPBL1 0008 7634h, ICU.GRPAL0 0008 7830h, ICU.GRPAL1 0008 7834h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IS31	IS30	IS29	IS28	IS27	IS26	IS25	IS24	IS23	IS22	IS21	IS20	IS19	IS18	IS17	IS16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	IS15	IS14	IS13	IS12	IS11	IS10	IS9	IS8	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IS0	Interrupt Status Flag 0	0: Interrupt is not requested. 1: Interrupt is requested.	R
b1	IS1	Interrupt Status Flag 1		R
b2	IS2	Interrupt Status Flag 2		R
b3	IS3	Interrupt Status Flag 3		R
b4	IS4	Interrupt Status Flag 4		R
b5	IS5	Interrupt Status Flag 5		R
b6	IS6	Interrupt Status Flag 6		R
b7	IS7	Interrupt Status Flag 7		R
b8	IS8	Interrupt Status Flag 8		R
b9	IS9	Interrupt Status Flag 9		R
b10	IS10	Interrupt Status Flag 10		R
b11	IS11	Interrupt Status Flag 11		R
b12	IS12	Interrupt Status Flag 12		R
b13	IS13	Interrupt Status Flag 13		R
b14	IS14	Interrupt Status Flag 14		R
b15	IS15	Interrupt Status Flag 15		R
b16	IS16	Interrupt Status Flag 16		R
b17	IS17	Interrupt Status Flag 17		R
b18	IS18	Interrupt Status Flag 18		R
b19	IS19	Interrupt Status Flag 19		R
b20	IS20	Interrupt Status Flag 20		R
b21	IS21	Interrupt Status Flag 21		R
b22	IS22	Interrupt Status Flag 22		R
b23	IS23	Interrupt Status Flag 23		R
b24	IS24	Interrupt Status Flag 24		R
b25	IS25	Interrupt Status Flag 25		R
b26	IS26	Interrupt Status Flag 26		R
b27	IS27	Interrupt Status Flag 27		R
b28	IS28	Interrupt Status Flag 28		R
b29	IS29	Interrupt Status Flag 29		R
b30	IS30	Interrupt Status Flag 30		R
b31	IS31	Interrupt Status Flag 31		R

These registers indicate the status of each interrupt request of group interrupt sources.
The GRPIE0 register contains the statuses of interrupt sources that are detected by edge detection and use ICLK as the operating clock.
The GRPBE0 register contains the statuses of interrupt sources that are detected by edge detection and use PCLKB as the operating clock.
Registers GRPBL0 and GRPBL1 contain the statuses of interrupt sources that are detected by level detection and use PCLKB as the operating clock.
The GRPAL0 and GRPAL1 registers contain the statuses of interrupt sources that are detected by level detection and use PCLKA as the operating clock.
These registers are collectively referred to as the “group interrupt request register”.
Refer to section 15.4.4, Group Interrupts for details on group interrupts.

ISj Flag (Interrupt Status Flag j) (j = 0 to 31)

The ISj flag indicates the status of interrupt sources assigned to group interrupts.
The ISj flag becomes 1 only when the corresponding ENj bit in the group interrupt request enable register is 1. When any of the ISj flags becomes 1, the IRn.IR flag corresponding to the group interrupt becomes 1 (n = 016 to 255).

(1) Group IE0

This flag becomes 1 under the following condition:

- The GRPIE0.ISj flag becomes 1 when the corresponding peripheral module interrupt request is generated while the GENIE0.ENj bit is 1.

This flag becomes 0 under the following condition:

- The GRPIE0.ISj flag becomes 0 when the GCRIE0.CLRj bit is set to 1.

(2) Group BE0

This flag becomes 1 under the following condition:

- The GRPBE0.ISj flag becomes 1 when the corresponding peripheral module interrupt request is generated while the GENBE0.ENj bit is 1.

This flag becomes 0 under the following condition:

- The GRPBE0.ISj flag becomes 0 when the GCRBE0.CLRj bit is set to 1.

(3) Group BL0/BL1

This flag becomes 1 under the following condition:

- The GRPBL0/GRPBL1.ISj flag becomes 1 while the corresponding peripheral module interrupt request is generated when the GENBL0/GENBL1.ENj bit is 1.

This flag becomes 0 under any of the following conditions:

- The GRPBL0/GRPBL1.ISj flag becomes 0 when the corresponding peripheral module interrupt request is cleared.
- The GRPBL0/GRPBL1.ISj flag becomes 0 when the GENBL0/GENBL1.ENj bit is set to 0.

(4) Group AL0/AL1

This flag becomes 1 under the following condition:

- The GRPAL0/GRPAL1.ISj flag becomes 1 while the corresponding peripheral module interrupt request is generated when the GENAL0/GENAL1.ENj bit is 1.

This flag becomes 0 under any of the following conditions:

- The GRPAL0/GRPAL1.ISj flag becomes 0 when the corresponding peripheral module interrupt request is cleared.
- The GRPAL0/GRPAL1.ISj flag becomes 0 when the GENAL0/GENAL1.ENj bit is set to 0.

15.2.24 Group IE0 Interrupt Request Enable Register (GENIE0), Group BE0 Interrupt Request Enable Register (GENBE0), Group BL0/BL1 Interrupt Request Enable Register (GENBL0/GENBL1), Group AL0/AL1 Interrupt Request Enable Register (GENAL0/GENAL1)

Address(es): ICU.GENIE0 0008 75B4h, ICU.GENBE0 0008 7640h, ICU.GENBL0 0008 7670h, ICU.GENBL1 0008 7674h, ICU.GENAL0 0008 7870h, ICU.GENAL1 0008 7874h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	EN31	EN30	EN29	EN28	EN27	EN26	EN25	EN24	EN23	EN22	EN21	EN20	EN19	EN18	EN17	EN16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EN0	Interrupt Request Enable 0	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b1	EN1	Interrupt Request Enable 1		R/W
b2	EN2	Interrupt Request Enable 2		R/W
b3	EN3	Interrupt Request Enable 3		R/W
b4	EN4	Interrupt Request Enable 4		R/W
b5	EN5	Interrupt Request Enable 5		R/W
b6	EN6	Interrupt Request Enable 6		R/W
b7	EN7	Interrupt Request Enable 7		R/W
b8	EN8	Interrupt Request Enable 8		R/W
b9	EN9	Interrupt Request Enable 9		R/W
b10	EN10	Interrupt Request Enable 10		R/W
b11	EN11	Interrupt Request Enable 11		R/W
b12	EN12	Interrupt Request Enable 12		R/W
b13	EN13	Interrupt Request Enable 13		R/W
b14	EN14	Interrupt Request Enable 14		R/W
b15	EN15	Interrupt Request Enable 15		R/W
b16	EN16	Interrupt Request Enable 16		R/W
b17	EN17	Interrupt Request Enable 17		R/W
b18	EN18	Interrupt Request Enable 18		R/W
b19	EN19	Interrupt Request Enable 19		R/W
b20	EN20	Interrupt Request Enable 20		R/W
b21	EN21	Interrupt Request Enable 21		R/W
b22	EN22	Interrupt Request Enable 22		R/W
b23	EN23	Interrupt Request Enable 23		R/W
b24	EN24	Interrupt Request Enable 24		R/W
b25	EN25	Interrupt Request Enable 25		R/W
b26	EN26	Interrupt Request Enable 26		R/W
b27	EN27	Interrupt Request Enable 27		R/W
b28	EN28	Interrupt Request Enable 28		R/W
b29	EN29	Interrupt Request Enable 29		R/W
b30	EN30	Interrupt Request Enable 30		R/W
b31	EN31	Interrupt Request Enable 31		R/W

Note: When a bit has no corresponding interrupt source (bit is reserved), set the bit to 0.

These registers select whether the ISj flag in the group interrupt request register is set to 1 when each interrupt request for

group interrupt sources is generated. These registers are collectively referred to as the “group interrupt request enable register”.
Registers GENIE0, GENBE0, GENBL0/GENBL1, and GENAL0/GENAL1 control the ISj flag in registers GRPIE0, GRPBE0, GRPBL0/GRPBL1, and GRPAL0/GRPAL1, respectively.
Refer to section 15.4.4, Group Interrupts for details on group interrupts.

ENj Bits (Interrupt Request Enable j) (j = 0 to 31)

The ENj bits select whether the corresponding ISj flag in the group interrupt request register is set to 1 when an interrupt request assigned to group interrupts is generated.

(1) Group IE0

When a peripheral module interrupt request is generated while the corresponding GENIE0.ENj bit is 1, the GRPIE0.ISj flag becomes 1. When the ENj bit is 0, the ISj flag does not become 1.
Even when the ENj bit is set to 0, the ISj flag does not change.

(2) Group BE0

When a peripheral module interrupt request is generated while the corresponding GENBE0.ENj bit is 1, the GRPBE0.ISj flag becomes 1. When the ENj bit is 0, the ISj flag does not become 1.
Even when the ENj bit is set to 0, the ISj flag does not change.

(3) Group BL0/BL1

When a peripheral module interrupt request is generated while the corresponding GENBL0/GENBL1.ENj bit is 1, the GRPBL0/GRPBL1.ISj flag becomes 1. When the ENj bit is 0, the ISj flag does not become 1.
When the ENj bit is set to 0, the ISj flag becomes 0.

(4) Group AL0/AL1

When a peripheral module interrupt request is generated while the corresponding GENAL0/GENAL1.ENj bit is 1, the GRPAL0/GRPAL1.ISj flag becomes 1. When the ENj bit is 0, the ISj flag does not become 1.
When the ENj bit is set to 0, the ISj flag becomes 0.

15.2.25 Group IE0 Interrupt Clear Register (GCRIE0), Group BE0 Interrupt Clear Register (GCRBE0)

Address(es): ICU.GCRIE0 0008 75B8h, ICU.GCRBE0 0008 7680h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CLR31	CLR30	CLR29	CLR28	CLR27	CLR26	CLR25	CLR24	CLR23	CLR22	CLR21	CLR20	CLR19	CLR18	CLR17	CLR16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CLR15	CLR14	CLR13	CLR12	CLR11	CLR10	CLR9	CLR8	CLR7	CLR6	CLR5	CLR4	CLR3	CLR2	CLR1	CLR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CLR0	Interrupt Source Clear 0	The read value is 0. When the CLRj bit is set to 1, the corresponding interrupt status flag (GRPIE0/GRPBE0.ISj) becomes 0 (j = 0 to 31). Writing 0 has no effect.	R/(W)
b1	CLR1	Interrupt Source Clear 1		R/(W)
b2	CLR2	Interrupt Source Clear 2		R/(W)
b3	CLR3	Interrupt Source Clear 3		R/(W)
b4	CLR4	Interrupt Source Clear 4		R/(W)
b5	CLR5	Interrupt Source Clear 5		R/(W)
b6	CLR6	Interrupt Source Clear 6		R/(W)
b7	CLR7	Interrupt Source Clear 7		R/(W)
b8	CLR8	Interrupt Source Clear 8		R/(W)
b9	CLR9	Interrupt Source Clear 9		R/(W)
b10	CLR10	Interrupt Source Clear 10		R/(W)
b11	CLR11	Interrupt Source Clear 11		R/(W)
b12	CLR12	Interrupt Source Clear 12		R/(W)
b13	CLR13	Interrupt Source Clear 13		R/(W)
b14	CLR14	Interrupt Source Clear 14		R/(W)
b15	CLR15	Interrupt Source Clear 15		R/(W)
b16	CLR16	Interrupt Source Clear 16		R/(W)
b17	CLR17	Interrupt Source Clear 17		R/(W)
b18	CLR18	Interrupt Source Clear 18		R/(W)
b19	CLR19	Interrupt Source Clear 19		R/(W)
b20	CLR20	Interrupt Source Clear 20		R/(W)
b21	CLR21	Interrupt Source Clear 21		R/(W)
b22	CLR22	Interrupt Source Clear 22		R/(W)
b23	CLR23	Interrupt Source Clear 23		R/(W)
b24	CLR24	Interrupt Source Clear 24		R/(W)
b25	CLR25	Interrupt Source Clear 25		R/(W)
b26	CLR26	Interrupt Source Clear 26		R/(W)
b27	CLR27	Interrupt Source Clear 27		R/(W)
b28	CLR28	Interrupt Source Clear 28		R/(W)
b29	CLR29	Interrupt Source Clear 29		R/(W)
b30	CLR30	Interrupt Source Clear 30		R/(W)
b31	CLR31	Interrupt Source Clear 31		R/(W)

Note: Write 1 to only the bit corresponding to the flag that is cleared, and write 0 to the other bits.

The GCRIE0 and GCRBE0 registers clear the GRPIE0.ISj and GRPBE0.ISj flags, respectively (j = 0 to 31). Refer to section 15.4.4, Group Interrupts for details on group interrupts.

CLRj Bit (Interrupt Source Clear j) (j = 0 to 31)

When the GCRIE0/GCRBE0.CLRj bit is set to 1, the GRPIE0/GRPBE0.ISj flag becomes 0.

15.2.26 Software Configurable Interrupt B Request Register k (PIBRk) (k = 0h to Ch)

Address(es): ICU.PIBR0 0008 7700h, ICU.PIBR1 0008 7701h, ICU.PIBR2 0008 7702h, ICU.PIBR3 0008 7703h, ICU.PIBR4 0008 7704h, ICU.PIBR5 0008 7705h, ICU.PIBR6 0008 7706h, ICU.PIBR7 0008 7707h, ICU.PIBR8 0008 7708h, ICU.PIBR9 0008 7709h, ICU.PIBRA 0008 770Ah, ICU.PIBRB 0008 770Bh, ICU.PIBRC 0008 770Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	PIR7	PIR6	PIR5	PIR4	PIR3	PIR2	PIR1	PIR0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIR0	Software Configurable Interrupt B Status Flag 0	When reading 0: Interrupt is not requested. 1: Interrupt is requested. When writing *1 0: Ignored. 1: The software configurable interrupt B status flag is cleared.	R/W
b1	PIR1	Software Configurable Interrupt B Status Flag 1		R/W
b2	PIR2	Software Configurable Interrupt B Status Flag 2		R/W
b3	PIR3	Software Configurable Interrupt B Status Flag 3		R/W
b4	PIR4	Software Configurable Interrupt B Status Flag 4		R/W
b5	PIR5	Software Configurable Interrupt B Status Flag 5		R/W
b6	PIR6	Software Configurable Interrupt B Status Flag 6		R/W
b7	PIR7	Software Configurable Interrupt B Status Flag 7		R/W

Note 1. Do not use bit manipulation instructions. If a bit manipulation instruction is used, multiple status flags may be cleared. Write 1 only to the flag to be cleared and write 0 to the other flags (write to this register in 8-bit units).

The PIBRk register is used for polling of interrupt requests of interrupt sources that is assigned to software configurable interrupt B by software. For an interrupt request of software configurable interrupt B set in the SLIBXRn or SLIBRn register, use the corresponding IRn.IR flag for polling (n = 128 to 207).
Refer to Table 15.3, Interrupt Sources for Software Configurable Interrupt B for correspondence between interrupt source numbers and interrupt sources for software configurable interrupt B.

PIRj Flag (Software Configurable Interrupt B Status Flag j) (j = 0 to 7)

When an interrupt request of interrupt sources assigned to software configurable interrupt B is generated, the corresponding PIBRk.PIRj flag becomes 1 regardless of whether the interrupt source is selected in the SLIBXRn or SLIBRn register.
Although the PIRj flag does not become 0 after the interrupt request is accepted by the destination (CPU, DTC, DMAC), generation of interrupt requests is not affected.
When using the PIRj flag for polling, set the PIRj flag to 0 by writing 1 to the flag in advance.
This flag becomes 1 under the following condition:

- An interrupt request is generated

This flag becomes 0 under the following condition:

- 1 is written to the PIBRk.PIRj flag

15.2.27 Software Configurable Interrupt A Request Register k (PIARk) (k = 0h to 5h, Bh)

Address(es): ICU.PIAR0 0008 7900h, ICU.PIAR1 0008 7901h, ICU.PIAR2 0008 7902h, ICU.PIAR3 0008 7903h, ICU.PIAR4 0008 7904h, ICU.PIAR5 0008 7905h, ICU.PIARB 0008 790Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	PIR7	PIR6	PIR5	PIR4	PIR3	PIR2	PIR1	PIR0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIR0	Software Configurable Interrupt A Status Flag 0	When reading 0: Interrupt is not requested. 1: Interrupt is requested. When writing *1 0: Ignored. 1: The Software Configurable interrupt A status flag is cleared.	R/W
b1	PIR1	Software Configurable Interrupt A Status Flag 1		R/W
b2	PIR2	Software Configurable Interrupt A Status Flag 2		R/W
b3	PIR3	Software Configurable Interrupt A Status Flag 3		R/W
b4	PIR4	Software Configurable Interrupt A Status Flag 4		R/W
b5	PIR5	Software Configurable Interrupt A Status Flag 5		R/W
b6	PIR6	Software Configurable Interrupt A Status Flag 6		R/W
b7	PIR7	Software Configurable Interrupt A Status Flag 7		R/W

Note 1. Do not use bit manipulation instructions. If a bit manipulation instruction is used, multiple status flags may be cleared. Write 1 only to the flag to be cleared and write 0 to the other flags (write to this register in 8-bit units).

The PIARk register is used for polling interrupt requests of interrupt sources that is assigned to software configurable interrupt A by software. For an interrupt request of software configurable interrupt A set in the SLIARn register, use the corresponding IRn.IR flag for polling (n = 208 to 255).
Refer to Table 15.4, Interrupt Sources for Software Configurable Interrupt A for correspondence between interrupt source numbers and interrupt sources for software configurable interrupt A.

PIRj Flag (Software Configurable Interrupt A Status Flag j) (j = 0 to 7)

When an interrupt request of interrupt sources assigned to software configurable interrupt A is generated, the corresponding PIARk.PIRj flag becomes 1 regardless of whether the interrupt source is selected in the SLIARn register. Although the PIRj flag does not become 0 after the interrupt request is accepted by the destination (CPU, DTC, DMAC), generation of interrupt requests is not affected.
When using the PIRj flag for polling, set the PIRj flag to 0 by writing 1 to the flag in advance.
This flag becomes 1 under the following condition:

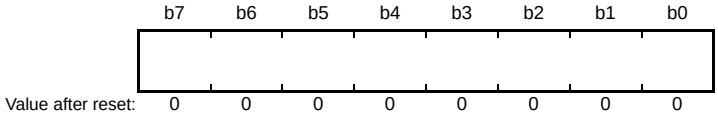
- An interrupt request is generated

This flag becomes 0 under the following condition:

- 1 is written to the PIARk.PIRj flag

15.2.28 Software Configurable Interrupt B Source Select Register Xn (SLIBXRn) (n = 128 to 143)

Address(es): ICU.SLIBXR128 0008 7780h to ICU.SLIBXR143 0008 778Fh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	—	00h: No interrupt source selected. 01h: Interrupt source number 1 : FEh: Interrupt source number 254 FFh: Interrupt source number 255	R/W*1

Note 1. Writing to these bits is ignored while the SLIPRCR.WPRC bit is set to 1.

The SLIBXRn register is used to assign interrupt vector numbers 128 to 143 to interrupt sources assigned to software configurable interrupt B.

Table 15.3, Interrupt Sources for Software Configurable Interrupt B lists interrupt sources assigned to software configurable interrupt B. Set the SLIBXRn register to an interrupt source number that is not reserved. When 00h or FFh are set, no interrupt source is assigned to interrupt vector number n.

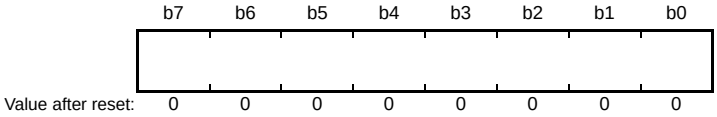
Do not assign the same interrupt source to multiple registers from registers SLIBXRn and SLIBRn.

Some interrupt sources can be used to start the DTC or DMAC. Refer to Table 15.3, Interrupt Sources for Software Configurable Interrupt B for details on which sources can be used to start the DTC or DMAC.

Refer to section 15.7.7, Setting Software Configurable Interrupts for the procedure to set software configurable interrupts.

15.2.29 Software Configurable Interrupt B Source Select Register n (SLIBRn) (n = 144 to 207)

Address(es): ICU.SLIBR144 0008 7790h to ICU.SLIBR207 0008 77CFh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	—	00h: No interrupt source selected. 01h: Interrupt source number 1 : FEh: Interrupt source number 254 FFh: No interrupt source selected.	R/W*1

Note 1. Writing to these bits is ignored while the SLIPRCR.WPRC bit is set to 1.

The SLIBRn register is used to assign interrupt vector numbers 144 to 207 to interrupt sources for software configurable interrupt B.

Table 15.3, Interrupt Sources for Software Configurable Interrupt B lists interrupt sources for software configurable interrupt B. Set the SLIBRn register to an interrupt source number that is not reserved. When 00h or FFh are set, no interrupt source is assigned to interrupt vector number n.

Do not assign the same interrupt source to multiple registers from registers SLIBXRn and SLIBRn.

Some interrupt sources can be used to start the DTC or DMAC. Refer to Table 15.3, Interrupt Sources for Software Configurable Interrupt B for details on which sources can be used to start the DTC or DMAC.

Although interrupt sources of interrupt vector numbers 144 and 145 can be used to start EXDMAC0 and EXDMAC1, respectively, only interrupt source number 20, TPU1.TGI1A (TGRA input capture/compare match) can be set as a source to start the EXDMAC. Set only one of either the SLIBR144 register or SLIBR145 register to 20, and set the corresponding bit (SELEXD0 or SELEXD1 bit) in the SELEXDR register to 0.

Refer to section 15.7.7, Setting Software Configurable Interrupts for the procedure to set software configurable interrupts.

Table 15.3 Interrupt Sources for Software Configurable Interrupt B (1/3)

Interrupt Source No.	Category	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
0	—	None	No interrupt selected (initial value)	N/A	N/A	PIBR0.PIR0
1	Edge	CMT2	CMI2 (CMCOR compare match)	✓	✓	PIBR0.PIR1
2		CMT3	CMI3 (CMCOR compare match)	✓	✓	PIBR0.PIR2
3		TMR0	CMIA0 (TCORA compare match)	✓	N/A	PIBR0.PIR3
4			CMIB0 (TCORB compare match)	✓	N/A	PIBR0.PIR4
5			OVI0 (TCNT overflow)	N/A	N/A	PIBR0.PIR5
6		TMR1	CMIA1 (TCORA compare match)	✓	N/A	PIBR0.PIR6
7			CMIB1 (TCORB compare match)	✓	N/A	PIBR0.PIR7
8			OVI1 (TCNT overflow)	N/A	N/A	PIBR1.PIR0
9		TMR2	CMIA2 (TCORA compare match)	✓	N/A	PIBR1.PIR1
10			CMIB2 (TCORB compare match)	✓	N/A	PIBR1.PIR2
11			OVI2 (TCNT overflow)	N/A	N/A	PIBR1.PIR3
12		TMR3	CMIA3 (TCORA compare match)	✓	N/A	PIBR1.PIR4
13			CMIB3 (TCORB compare match)	✓	N/A	PIBR1.PIR5
14			OVI3 (TCNT overflow)	N/A	N/A	PIBR1.PIR6
15		TPU0	TGI0A (TGRA input capture/compare match)	✓	✓	PIBR1.PIR7
16			TGI0B (TGRB input capture/compare match)	✓	N/A	PIBR2.PIR0
17			TGI0C (TGRC input capture/compare match)	✓	N/A	PIBR2.PIR1
18			TGI0D (TGRD input capture/compare match)	✓	N/A	PIBR2.PIR2
19			TCI0V (TCNT overflow)	N/A	N/A	PIBR2.PIR3
20*1		TPU1	TGI1A (TGRA input capture/compare match)	✓	✓	PIBR2.PIR4
21			TGI1B (TGRB input capture/compare match)	✓	N/A	PIBR2.PIR5
22			TCI1V (TCNT overflow)	N/A	N/A	PIBR2.PIR6
23			TCI1U (TCNT underflow)	N/A	N/A	PIBR2.PIR7
24		TPU2	TGI2A (TGRA input capture/compare match)	✓	✓	PIBR3.PIR0
25			TGI2B (TGRB input capture/compare match)	✓	N/A	PIBR3.PIR1
26			TCI2V (TCNT overflow)	N/A	N/A	PIBR3.PIR2
27			TCI2U (TCNT underflow)	N/A	N/A	PIBR3.PIR3
28		TPU3	TGI3A (TGRA input capture/compare match)	✓	✓	PIBR3.PIR4
29			TGI3B (TGRB input capture/compare match)	✓	N/A	PIBR3.PIR5
30			TGI3C (TGRC input capture/compare match)	✓	N/A	PIBR3.PIR6
31			TGI3D (TGRD input capture/compare match)	✓	N/A	PIBR3.PIR7
32			TCI3V (TCNT overflow)	N/A	N/A	PIBR4.PIR0
33		TPU4	TGI4A (TGRA input capture/compare match)	✓	✓	PIBR4.PIR1
34			TGI4B (TGRB input capture/compare match)	✓	N/A	PIBR4.PIR2
35			TCI4V (TCNT overflow)	N/A	N/A	PIBR4.PIR3
36			TCI4U (TCNT underflow)	N/A	N/A	PIBR4.PIR4
37		TPU5	TGI5A (TGRA input capture/compare match)	✓	✓	PIBR4.PIR5
38			TGI5B (TGRB input capture/compare match)	✓	N/A	PIBR4.PIR6
39			TCI5V (TCNT overflow)	N/A	N/A	PIBR4.PIR7
40			TCI5U (TCNT underflow)	N/A	N/A	PIBR5.PIR0

Table 15.3 Interrupt Sources for Software Configurable Interrupt B (2/3)

Interrupt Source No.	Category	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
41	Edge	CMTW0	IC0I0 (input capture of the CMWICR0 register)	✓	✓	PIBR5.PIR1
42			IC1I0 (input capture of the CMWICR1 register)	✓	✓	PIBR5.PIR2
43			OC0I0 (output compare of the CMWOCR0 register)	✓	✓	PIBR5.PIR3
44			OC1I0 (output compare of the CMWOCR1 register)	✓	✓	PIBR5.PIR4
45		CMTW1	IC0I1 (input capture of the CMWICR0 register)	✓	✓	PIBR5.PIR5
46			IC1I1 (input capture of the CMWICR1 register)	✓	✓	PIBR5.PIR6
47			OC0I1 (output compare of the CMWOCR0 register)	✓	✓	PIBR5.PIR7
48			OC1I1 (output compare of the CMWOCR1 register)	✓	✓	PIBR6.PIR0
49		RTC	CUP (carry interrupt)	N/A	N/A	PIBR6.PIR1
50		CAN0	RXF0 (receive FIFO interrupt)	N/A	N/A	PIBR6.PIR2
51			TXF0 (transmit FIFO interrupt)	N/A	N/A	PIBR6.PIR3
52			RXM0 (mailboxes 0 to 31 message reception completed)	N/A	N/A	PIBR6.PIR4
53			TXM0 (mailboxes 0 to 31 message transmission completed)	N/A	N/A	PIBR6.PIR5
54		CAN1	RXF1 (receive FIFO interrupt)	N/A	N/A	PIBR6.PIR6
55			TXF1 (transmit FIFO interrupt)	N/A	N/A	PIBR6.PIR7
56			RXM1 (mailboxes 0 to 31 message reception completed)	N/A	N/A	PIBR7.PIR0
57			TXM1 (mailboxes 0 to 31 message transmission completed)	N/A	N/A	PIBR7.PIR1
58		Reserved	—	N/A	N/A	PIBR7.PIR2
59		Reserved	—	N/A	N/A	PIBR7.PIR3
60		Reserved	—	N/A	N/A	PIBR7.PIR4
61		Reserved	—	N/A	N/A	PIBR7.PIR5
62		USB0	USBI0 (status interrupt from 15 sources)	N/A	N/A	PIBR7.PIR6
63		USB1	USBI1 (status interrupt from 15 sources)	N/A	N/A	PIBR7.PIR7
64		S12AD	S12ADI (A/D conversion end)	✓	✓	PIBR8.PIR0
65			S12GBADI (group B A/D conversion end interrupt)	✓	✓	PIBR8.PIR1
66			S12GCADI (group C A/D conversion end interrupt)	✓	✓	PIBR8.PIR2
67		Reserved	—	N/A	N/A	PIBR8.PIR3
68		S12AD1	S12ADI1 (A/D conversion end)	✓	✓	PIBR8.PIR4
69			S12GBADI1 (group B A/D conversion end interrupt)	✓	✓	PIBR8.PIR5
70			S12GCADI1 (group C A/D conversion end interrupt)	✓	✓	PIBR8.PIR6
71		Reserved	—	N/A	N/A	PIBR8.PIR7
72		Reserved	—	N/A	N/A	PIBR9.PIR0
73		Reserved	—	N/A	N/A	PIBR9.PIR1
74		Reserved	—	N/A	N/A	PIBR9.PIR2
75		Reserved	—	N/A	N/A	PIBR9.PIR3
76		Reserved	—	N/A	N/A	PIBR9.PIR4
77		Reserved	—	N/A	N/A	PIBR9.PIR5
78		Reserved	—	N/A	N/A	PIBR9.PIR6
79		ELC	ELSR18I (ELC interrupt)	✓	✓	PIBR9.PIR7
80			ELSR19I (ELC interrupt)	✓	✓	PIBRA.PIR0
81 to 84		Reserved	—	N/A	N/A	—

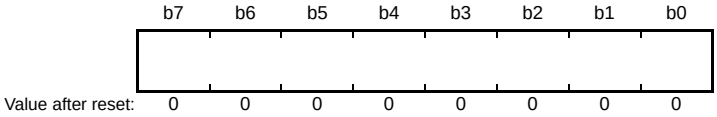
Table 15.3 Interrupt Sources for Software Configurable Interrupt B (3/3)

Interrupt Source No.	Category	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
85	Edge	TSIP	PROC_BUSY (procedure completed)	N/A	N/A	PIBRA.PIR5
86			ROMOK (ROM falsification detected)	N/A	N/A	PIBRA.PIR6
87			LONG_PLG (operation completed)	N/A	N/A	PIBRA.PIR7
88			TEST_BUSY (test busy)	N/A	N/A	PIBRB.PIR0
89			WRRDY0 (write ready 0)	N/A	N/A	PIBRB.PIR1
90			WRRDY1 (write ready 1)	N/A	N/A	PIBRB.PIR2
91			WRRDY4 (write ready 4)	N/A	N/A	PIBRB.PIR3
92			RDRDY0 (read ready 0)	N/A	N/A	PIBRB.PIR4
93			RDRDY1 (read ready 1)	N/A	N/A	PIBRB.PIR5
94			INTEGRATE_WRRDY (integrate write ready)	N/A	N/A	PIBRB.PIR6
95			INTEGRATE_RDRDY (integrate read ready)	N/A	N/A	PIBRB.PIR7
96		CTSU	CTSUWR (write request for setting registers for each channel)	✓	✓	PIBRC.PIR0
97			CTSURD (measurement data transfer request)	✓	✓	PIBRC.PIR1
98			CTSUFN (measurement end)	N/A	N/A	PIBRC.PIR2
99 to 254	—	Reserved	—	N/A	N/A	—
255	—	Reserved/ none *2	—/No interrupt selected	N/A	N/A	—

Note 1. No. 20 can be used as a trigger for EXDMAC0 or EXDMAC1.
Note 2. “Reserved” for the SLIBXRn register, and “none” for the SLIBRn register.

15.2.30 Software Configurable Interrupt A Source Select Register n (SLIARn) (n = 208 to 255)

Address(es): ICU.SLIAR208 0008 79D0h to ICU.SLIAR255 0008 79FFh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	—	00h: No interrupt source selected. 01h: Interrupt source number 1 : FEh: Interrupt source number 254 FFh: No interrupt source selected.	R/(W) *1

Note 1. Writing to these bits is ignored while the SLIPRCR.WPRC bit is set to 1.

The SLIARn register is used to assign interrupt vector numbers 208 to 255 to interrupt sources for software configurable interrupt A.

Table 15.4, Interrupt Sources for Software Configurable Interrupt A lists interrupt sources for software configurable interrupt A. Set the SLIARn register to an interrupt source number that is not reserved. When 00h or FFh are set, no interrupt source is assigned to interrupt vector number n.

Do not assign the same interrupt source to multiple SLIARn registers.

Some interrupt sources can be used to start the DTC or DMAC. Refer to Table 15.4, Interrupt Sources for Software Configurable Interrupt A for details on which sources can be used to start the DTC or DMAC.

Although interrupt sources of interrupt vector numbers 208 and 209 can be used to start EXDMAC0 and EXDMAC1, respectively, only interrupt source number 8, MTU1.TGIA1 (TGRA input capture/compare match) can be set as a source to start the EXDMAC. Set only one of either the SLIAR208 register or SLIAR209 register to 8, and set the corresponding bit (SELEXD0 or SELEXD1) in the SELEXDR register to 1.

Refer to section 15.7.7, Setting Software Configurable Interrupts for the procedure to set software configurable interrupts.

Table 15.4 Interrupt Sources for Software Configurable Interrupt A (1/2)

Interrupt Source No.	Category	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
0	—	None	No interrupt selected (initial value)	N/A	N/A	PIAR0.PIR0
1	Edge	MTU0	TGIA0 (TGRA input capture/compare match)	✓	✓	PIAR0.PIR1
2			TGIB0 (TGRB input capture/compare match)	✓	✓	PIAR0.PIR2
3			TGIC0 (TGRC input capture/compare match)	✓	✓	PIAR0.PIR3
4			TGID0 (TGRD input capture/compare match)	✓	✓	PIAR0.PIR4
5			TCIV0 (TCNT overflow)	N/A	N/A	PIAR0.PIR5
6			TGIE0 (TGRE compare match)	N/A	N/A	PIAR0.PIR6
7			TGIF0 (TGRF compare match)	N/A	N/A	PIAR0.PIR7
8*1		MTU1	TGIA1 (TGRA input capture/compare match)	✓	✓	PIAR1.PIR0
9			TGIB1 (TGRB input capture/compare match)	✓	✓	PIAR1.PIR1
10			TCIV1 (TCNT overflow)	N/A	N/A	PIAR1.PIR2
11			TCIU1 (TCNT underflow)	N/A	N/A	PIAR1.PIR3
12		MTU2	TGIA2 (TGRA input capture/compare match)	✓	✓	PIAR1.PIR4
13			TGIB2 (TGRB input capture/compare match)	✓	✓	PIAR1.PIR5
14			TCIV2 (TCNT overflow)	N/A	N/A	PIAR1.PIR6
15			TCIU2 (TCNT underflow)	N/A	N/A	PIAR1.PIR7
16		MTU3	TGIA3 (TGRA input capture/compare match)	✓	✓	PIAR2.PIR0
17			TGIB3 (TGRB input capture/compare match)	✓	✓	PIAR2.PIR1
18			TGIC3 (TGRC input capture/compare match)	✓	✓	PIAR2.PIR2
19			TGID3 (TGRD input capture/compare match)	✓	✓	PIAR2.PIR3
20			TCIV3 (TCNT overflow)	N/A	N/A	PIAR2.PIR4
21		MTU4	TGIA4 (TGRA input capture/compare match)	✓	✓	PIAR2.PIR5
22			TGIB4 (TGRB input capture/compare match)	✓	✓	PIAR2.PIR6
23			TGIC4 (TGRC input capture/compare match)	✓	✓	PIAR2.PIR7
24			TGID4 (TGRD input capture/compare match)	✓	✓	PIAR3.PIR0
25			TCIV4 (TCNT overflow/underflow (only for complementary PWM mode))	✓	✓	PIAR3.PIR1
26		Reserved	—	N/A	N/A	PIAR3.PIR2
27		MTU5	TGIU5 (TGRU input capture/compare match)	✓	✓	PIAR3.PIR3
28			TGIV5 (TGRV input capture/compare match)	✓	✓	PIAR3.PIR4
29			TGIW5 (TGRW input capture/compare match)	✓	✓	PIAR3.PIR5
30		MTU6	TGIA6 (TGRA input capture/compare match)	✓	✓	PIAR3.PIR6
31			TGIB6 (TGRB input capture/compare match)	✓	✓	PIAR3.PIR7
32			TGIC6 (TGRC input capture/compare match)	✓	✓	PIAR4.PIR0
33			TGID6 (TGRD input capture/compare match)	✓	✓	PIAR4.PIR1
34			TCIV6 (TCNT overflow)	N/A	N/A	PIAR4.PIR2
35		MTU7	TGIA7 (TGRA input capture/compare match)	✓	✓	PIAR4.PIR3
36			TGIB7 (TGRB input capture/compare match)	✓	✓	PIAR4.PIR4
37			TGIC7 (TGRC input capture/compare match)	✓	✓	PIAR4.PIR5
38			TGID7 (TGRD input capture/compare match)	✓	✓	PIAR4.PIR6
39			TCIV7 (TCNT overflow/underflow (only for complementary PWM mode))	✓	✓	PIAR4.PIR7
40		Reserved	—	N/A	N/A	PIAR5.PIR0

Table 15.4 Interrupt Sources for Software Configurable Interrupt A (2/2)

Interrupt Source No.	Category	Interrupt Request Source	Name	Start the DTC	Start the DMAC	Interrupt Status Flag
41	Edge	MTU8	TGIA8 (TGRA input capture/compare match)	✓	✓	PIAR5.PIR1
42			TGIB8 (TGRB input capture/compare match)	✓	✓	PIAR5.PIR2
43			TGIC8 (TGRC input capture/compare match)	✓	✓	PIAR5.PIR3
44			TGID8 (TGRD input capture/compare match)	✓	✓	PIAR5.PIR4
45			TCIV8 (TCNT overflow)	N/A	N/A	PIAR5.PIR5
46 to 89		Reserved	—	N/A	N/A	—
90		RSPIA0	SPCI (communication end)	N/A	N/A	PIARB.PIR2
91		RSPI0	SPCI0 (communication end)	N/A	N/A	PIARB.PIR3
92		RSPI1	SPCI1 (communication end)	N/A	N/A	PIARB.PIR4
93		RSPI2	SPCI2 (communication end)	N/A	N/A	PIARB.PIR5
94		RSCI10	AED (active edge detected)	N/A	N/A	PIARB.PIR6
95		RSCI11	AED (active edge detected)	N/A	N/A	PIARB.PIR7
96 to 254		Reserved	—	N/A	N/A	—
255	—	None	No interrupt selected	N/A	N/A	—

Note 1. No. 8 can be used as a trigger for EXDMAC0 or EXDMAC1.

15.2.31 EXDMAC Trigger Select Register (SELEXDR)

Address(es): ICU.SELEXDR 0008 7A01h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SELEXD1	SELEXD0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SELEXD0	EXDMAC0 Trigger Select	0: Interrupt source selected in the SLIBR144 register is used as the EXDMAC0 trigger. 1: Interrupt source selected in the SLIAR208 register is used as the EXDMAC0 trigger.	R/W*1
b1	SELEXD1	EXDMAC1 Trigger Select	0: Interrupt source selected in the SLIBR145 register is used as the EXDMAC1 trigger. 1: Interrupt source selected in the SLIAR209 register is used as the EXDMAC1 trigger.	R/W*1
b7 to b2	—	Reserved	The read value is 0. The write value should be 0.	R/W

Note 1. Writing to this bit is ignored while the SLIPRCR.WPRC bit is 1.

The SELEXDR register selects an interrupt source for software configurable interrupts as the EXDMAC trigger. Refer to section 15.4.5.3, EXDMAC Start Trigger by Software Configurable Interrupts for details on starting the EXDMAC by an interrupt request.

SELEXD0 Bit (EXDMAC0 Trigger Select)

The SELEXD0 bit selects an interrupt source set in the SLIBR144 register or SLIAR208 register as the EXDMAC0 trigger.

SELEXD1 Bit (EXDMAC1 Trigger Select)

The SELEXD1 bit selects an interrupt source set in the SLIBR145 register or SLIAR209 register as the EXDMAC1 trigger.

15.2.32 Software Configurable Interrupt Source Select Register Write Protect Register (SLIPRCR)

Address(es): ICU.SLIPRCR 0008 7A00h

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

Bit	Symbol	Bit Name	Description	R/W
b0	WPRC	Software Configurable Interrupt Source Select Register Write Protect	0: Write enabled. 1: Write disabled.	R/(W) *1
b7 to b1	—	Reserved	The read value is 0. The write value should be 0.	R

Note 1. Once this bit is set to 1, it cannot be set to 0 by software.

The SLIPRCR register protects registers that control assignment of software configurable interrupts from being written to.

WPRC Bit (Software Configurable Interrupt Source Select Register Write Protect)

The WPRC bit disables writing to the SLIBXRn, SLIBRn, SLIARn, and SELEXDR registers.
Once this bit is set to 1, it cannot be set to 0 by software.
After assigning software configurable interrupts, confirm that the WPRC bit is 1 before the corresponding interrupt request is generated. Refer to section 15.7.7, Setting Software Configurable Interrupts for the procedure to set software configurable interrupts.

15.3 Vector Table

There are two types of exceptions detected by the ICU: maskable interrupts (hereinafter referred to as “interrupts”) and non-maskable interrupts.
When the CPU accepts an interrupt or non-maskable interrupt, it acquires a 4-byte vector address from the vector table.

15.3.1 Interrupt Vector Table

The vector table that is used for maskable interrupts is called the interrupt vector table.
The interrupt vector table is allocated to a 1024-byte area (4 bytes × 256 sources) beginning with the address set in the INTB register in the CPU. Set the INTB register before enabling interrupts. Set a multiple of 4 in the INTB register.
An unconditional trap is generated when the INT or BRK instruction is executed. Interrupt vectors for unconditional traps use the same area as the interrupt vector table. The BRK instruction is assigned to interrupt vector number 0. The INT instruction is assigned to the interrupt vector number corresponding to the value set as the operand (0 to 255).
Table 15.5 lists details of the interrupt vectors. Details of the headings in Table 15.5 are listed below.

Heading	Description
Interrupt request generated by	Name of the source that generates the interrupt request (module symbol)
Name	Name of the interrupt source (symbol)
Vector no.	Interrupt vector number
Vector address offset	Offset from the address set in the INTB register
Interrupt detection method	“Edge” indicates that the interrupt is detected by edge detection. “Level” indicates that the interrupt is detected by level detection.
CPU interrupt	Interrupt source indicated by “✓” can be used as an interrupt source to the CPU.
Start the DTC	Interrupt source indicated by “✓” can be used as the DTC trigger.
Start the DMAC	Interrupt source indicated by “✓” can be used as the DMAC trigger.
Exit from SSBY	Interrupt source indicated by “✓” can be used as a source to exit software standby mode.
Exit from ACS	Interrupt source indicated by “✓” can be used as a source to exit all-module clock stop mode.
IER	Name of the bit in the IER register corresponding to the interrupt vector number
IPR	Name of the IPR register corresponding to the interrupt source
DTCER	Name of the DTCER register corresponding to the DTC trigger

Table 15.5 Interrupt Vector Table (1/6)

Interrupt Request is Generated By	Name	Vector No.	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	Exit from ACS	IER	IPR	DTCER
—	For an unconditional trap	0	0000h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	1	0004h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	2	0008h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	3	000Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	4	0010h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	5	0014h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	6	0018h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	7	001Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	8	0020h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	9	0024h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	10	0028h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	11	002Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	12	0030h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	13	0034h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	14	0038h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	15	003Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
BSC	BUSERR	16	0040h	Level	✓	N/A	N/A	N/A	N/A	IER02.IEN0	IPR000	—
ICU*1	GROUPIE0	17	0044h	Level	✓	N/A	N/A	N/A	N/A	IER02.IEN1	IPR000	—
RAM	RAMERR*2	18	0048h	Level	✓	N/A	N/A	N/A	N/A	IER02.IEN2	IPR000	—
—	Reserved	19	004Ch	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	20	0050h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
FCU	FIFERR	21	0054h	Level	✓	N/A	N/A	N/A	N/A	IER02.IEN5	IPR001	—
—	Reserved	22	0058h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
FCU	FRDYI	23	005Ch	Edge	✓	N/A	N/A	N/A	N/A	IER02.IEN7	IPR002	—
—	Reserved	24	0060h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	25	0064h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
ICU	SWINT2	26	0068h	Edge	✓	✓	N/A	N/A	N/A	IER03.IEN2	IPR003	DTCER026
	SWINT	27	006Ch	Edge	✓	✓	N/A	N/A	N/A	IER03.IEN3		DTCER027
CMT0	CMI0 (for OS)	28	0070h	Edge	✓	✓	✓	N/A	N/A	IER03.IEN4	IPR004	DTCER028
CMT1	CMI1	29	0074h	Edge	✓	✓	✓	N/A	N/A	IER03.IEN5	IPR005	DTCER029
CMTW0	CMWI0	30	0078h	Edge	✓	✓	✓	N/A	N/A	IER03.IEN6	IPR006	DTCER030
CMTW1	CMWI1	31	007Ch	Edge	✓	✓	✓	N/A	N/A	IER03.IEN7	IPR007	DTCER031
RSCI10	RXI	32	0080h	Edge	✓	✓	✓	N/A	N/A	IER04.IEN0	IPR032	DTCER032
	TXI	33	0084h	Edge	✓	✓	✓	N/A	N/A	IER04.IEN1	IPR033	DTCER033
USB0	D0FIFO0	34	0088h	Edge	✓	✓	✓	N/A	N/A	IER04.IEN2	IPR034	DTCER034
	D1FIFO0	35	008Ch	Edge	✓	✓	✓	N/A	N/A	IER04.IEN3	IPR035	DTCER035
USB1	D0FIFO1	36	0090h	Edge	✓	✓	✓	N/A	N/A	IER04.IEN4	IPR036	DTCER036
	D1FIFO1	37	0094h	Edge	✓	✓	✓	N/A	N/A	IER04.IEN5	IPR037	DTCER037
RSPI0	SPRI0	38	0098h	Edge	✓	✓	✓	N/A	N/A	IER04.IEN6	IPR038	DTCER038
	SPTI0	39	009Ch	Edge	✓	✓	✓	N/A	N/A	IER04.IEN7	IPR039	DTCER039
RSPI1	SPRI1	40	00A0h	Edge	✓	✓	✓	N/A	N/A	IER05.IEN0	IPR040	DTCER040
	SPTI1	41	00A4h	Edge	✓	✓	✓	N/A	N/A	IER05.IEN1	IPR041	DTCER041
RSCI11	RXI	42	00A8h	Edge	✓	✓	✓	N/A	N/A	IER05.IEN2	IPR042	DTCER042
	TXI	43	00ACh	Edge	✓	✓	✓	N/A	N/A	IER05.IEN3	IPR043	DTCER043

Table 15.5 Interrupt Vector Table (2/6)

Interrupt Request is Generated By	Name	Vector No.	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	Exit from ACS	IER	IPR	DTCER
SDHI	SBFAI	44	00B0h	Edge	✓	✓	✓	N/A	N/A	IER05.IEN4	IPR044	DTCER044
—	Reserved	45	00B4h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
SSIE0	SSITXI0	46	00B8h	Edge	✓	✓	✓	N/A	N/A	IER05.IEN6	IPR046	DTCER046
	SSIRXI0	47	00BCh	Edge	✓	✓	✓	N/A	N/A	IER05.IEN7	IPR047	DTCER047
RSPIA0	SPRI	48	00C0h	Edge	✓	✓	✓	N/A	N/A	IER06.IEN0	IPR048	DTCER048
	SPTI	49	00C4h	Edge	✓	✓	✓	N/A	N/A	IER06.IEN1	IPR049	DTCER049
RIIC1	RXI1	50	00C8h	Edge	✓	✓	✓	N/A	N/A	IER06.IEN2	IPR050	DTCER050
	TXI1	51	00CCh	Edge	✓	✓	✓	N/A	N/A	IER06.IEN3	IPR051	DTCER051
RIIC0	RXI0	52	00D0h	Edge	✓	✓	✓	N/A	N/A	IER06.IEN4	IPR052	DTCER052
	TXI0	53	00D4h	Edge	✓	✓	✓	N/A	N/A	IER06.IEN5	IPR053	DTCER053
RIIC2	RXI2	54	00D8h	Edge	✓	✓	✓	N/A	N/A	IER06.IEN6	IPR054	DTCER054
	TXI2	55	00DCh	Edge	✓	✓	✓	N/A	N/A	IER06.IEN7	IPR055	DTCER055
—	Reserved	56	00E0h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
—	Reserved	57	00E4h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
SCI0	RXI0	58	00E8h	Edge	✓	✓	✓	N/A	N/A	IER07.IEN2	IPR058	DTCER058
	TXI0	59	00ECh	Edge	✓	✓	✓	N/A	N/A	IER07.IEN3	IPR059	DTCER059
SCI1	RXI1	60	00F0h	Edge	✓	✓	✓	N/A	N/A	IER07.IEN4	IPR060	DTCER060
	TXI1	61	00F4h	Edge	✓	✓	✓	N/A	N/A	IER07.IEN5	IPR061	DTCER061
SCI2	RXI2	62	00F8h	Edge	✓	✓	✓	N/A	N/A	IER07.IEN6	IPR062	DTCER062
	TXI2	63	00FCh	Edge	✓	✓	✓	N/A	N/A	IER07.IEN7	IPR063	DTCER063
ICU	IRQ0	64	0100h	Edge/ Level	✓	✓	✓	✓	✓	IER08.IEN0	IPR064	DTCER064
	IRQ1	65	0104h	Edge/ Level	✓	✓	✓	✓	✓	IER08.IEN1	IPR065	DTCER065
	IRQ2	66	0108h	Edge/ Level	✓	✓	✓	✓	✓	IER08.IEN2	IPR066	DTCER066
	IRQ3	67	010Ch	Edge/ Level	✓	✓	✓	✓	✓	IER08.IEN3	IPR067	DTCER067
	IRQ4	68	0110h	Edge/ Level	✓	✓	✓	✓	✓	IER08.IEN4	IPR068	DTCER068
	IRQ5	69	0114h	Edge/ Level	✓	✓	✓	✓	✓	IER08.IEN5	IPR069	DTCER069
	IRQ6	70	0118h	Edge/ Level	✓	✓	✓	✓	✓	IER08.IEN6	IPR070	DTCER070
	IRQ7	71	011Ch	Edge/ Level	✓	✓	✓	✓	✓	IER08.IEN7	IPR071	DTCER071
	IRQ8	72	0120h	Edge/ Level	✓	✓	✓	✓	✓	IER09.IEN0	IPR072	DTCER072
	IRQ9	73	0124h	Edge/ Level	✓	✓	✓	✓	✓	IER09.IEN1	IPR073	DTCER073
	IRQ10	74	0128h	Edge/ Level	✓	✓	✓	✓	✓	IER09.IEN2	IPR074	DTCER074
	IRQ11	75	012Ch	Edge/ Level	✓	✓	✓	✓	✓	IER09.IEN3	IPR075	DTCER075
	IRQ12	76	0130h	Edge/ Level	✓	✓	✓	✓	✓	IER09.IEN4	IPR076	DTCER076
	IRQ13	77	0134h	Edge/ Level	✓	✓	✓	✓	✓	IER09.IEN5	IPR077	DTCER077
	IRQ14	78	0138h	Edge/ Level	✓	✓	✓	✓	✓	IER09.IEN6	IPR078	DTCER078
	IRQ15	79	013Ch	Edge/ Level	✓	✓	✓	✓	✓	IER09.IEN7	IPR079	DTCER079

Table 15.5 Interrupt Vector Table (3/6)

Interrupt Request is Generated By	Name	Vector No.	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	Exit from ACS	IER	IPR	DTCER
SCI3	RXI3	80	0140h	Edge	✓	✓	✓	N/A	N/A	IER0A.IEN0	IPR080	DTCER080
	TXI3	81	0144h	Edge	✓	✓	✓	N/A	N/A	IER0A.IEN1	IPR081	DTCER081
SCI4	RXI4	82	0148h	Edge	✓	✓	✓	N/A	N/A	IER0A.IEN2	IPR082	DTCER082
	TXI4	83	014Ch	Edge	✓	✓	✓	N/A	N/A	IER0A.IEN3	IPR083	DTCER083
SCI5	RXI5	84	0150h	Edge	✓	✓	✓	N/A	N/A	IER0A.IEN4	IPR084	DTCER084
	TXI5	85	0154h	Edge	✓	✓	✓	N/A	N/A	IER0A.IEN5	IPR085	DTCER085
SCI6	RXI6	86	0158h	Edge	✓	✓	✓	N/A	N/A	IER0A.IEN6	IPR086	DTCER086
	TXI6	87	015Ch	Edge	✓	✓	✓	N/A	N/A	IER0A.IEN7	IPR087	DTCER087
LVD1	LVD1	88	0160h	Edge	✓	N/A	N/A	✓	✓	IER0B.IEN0	IPR088	—
LVD2	LVD2	89	0164h	Edge	✓	N/A	N/A	✓	✓	IER0B.IEN1	IPR089	—
USB0	USBR0	90	0168h	Level	✓	N/A	N/A	✓	✓	IER0B.IEN2	IPR090	—
VBATT	TAMPDI	91	016Ch	Level	✓	N/A	N/A	✓	✓	IER0B.IEN3	IPR091	—
RTC	ALM	92	0170h	Edge	✓	N/A	N/A	✓	✓	IER0B.IEN4	IPR092	—
	PRD	93	0174h	Edge	✓	N/A	N/A	✓	✓	IER0B.IEN5	IPR093	—
REMC0	REMCI0	94	0178h	Edge	✓	N/A	N/A	✓	✓	IER0B.IEN6	IPR094	—
IWDT	IWUNI*2	95	017Ch	Edge	✓	N/A	N/A	✓	✓	IER0B.IEN7	IPR095	—
WDT	WUNI*2	96	0180h	Edge	✓	N/A	N/A	N/A	N/A	IER0C.IEN0	IPR096	—
—	Reserved	97	0184h	—	N/A	N/A	N/A	N/A	N/A	—	—	—
SCI7	RXI7	98	0188h	Edge	✓	✓	✓	N/A	N/A	IER0C.IEN2	IPR098	DTCER098
	TXI7	99	018Ch	Edge	✓	✓	✓	N/A	N/A	IER0C.IEN3	IPR099	DTCER099
SCI8	RXI8	100	0190h	Edge	✓	✓	✓	N/A	N/A	IER0C.IEN4	IPR100	DTCER100
	TXI8	101	0194h	Edge	✓	✓	✓	N/A	N/A	IER0C.IEN5	IPR101	DTCER101
SCI9	RXI9	102	0198h	Edge	✓	✓	✓	N/A	N/A	IER0C.IEN6	IPR102	DTCER102
	TXI9	103	019Ch	Edge	✓	✓	✓	N/A	N/A	IER0C.IEN7	IPR103	DTCER103
SCI10	RXI10	104	01A0h	Edge	✓	✓	✓	N/A	N/A	IER0D.IEN0	IPR104	DTCER104
	TXI10	105	01A4h	Edge	✓	✓	✓	N/A	N/A	IER0D.IEN1	IPR105	DTCER105
ICU*1	GROUPBE0	106	01A8h	Level	✓	N/A	N/A	N/A	N/A	IER0D.IEN2	IPR106	—
	GROUPBL2	107	01ACh	Level	✓	N/A	N/A	N/A	N/A	IER0D.IEN3	IPR107	—
RSPI2	SPRI2	108	01B0h	Edge	✓	✓	✓	N/A	N/A	IER0D.IEN4	IPR108	DTCER108
	SPTI2	109	01B4h	Edge	✓	✓	✓	N/A	N/A	IER0D.IEN5	IPR109	DTCER109
ICU*1	GROUPBL0	110	01B8h	Level	✓	N/A	N/A	N/A	N/A	IER0D.IEN6	IPR110	—
	GROUPBL1	111	01BCh	Level	✓	N/A	N/A	N/A	N/A	IER0D.IEN7	IPR111	—
	GROUPAL0	112	01C0h	Level	✓	N/A	N/A	N/A	N/A	IER0E.IEN0	IPR112	—
	GROUPAL1	113	01C4h	Level	✓	N/A	N/A	N/A	N/A	IER0E.IEN1	IPR113	—
SCI11	RXI11	114	01C8h	Edge	✓	✓	✓	N/A	N/A	IER0E.IEN2	IPR114	DTCER114
	TXI11	115	01CCh	Edge	✓	✓	✓	N/A	N/A	IER0E.IEN3	IPR115	DTCER115
SCI12	RXI12	116	01D0h	Edge	✓	✓	✓	N/A	N/A	IER0E.IEN4	IPR116	DTCER116
	TXI12	117	01D4h	Edge	✓	✓	✓	N/A	N/A	IER0E.IEN5	IPR117	DTCER117
RIICHS0	RXI	118	01D8h	Edge	✓	✓	✓	N/A	N/A	IER0E.IEN6	IPR118	DTCER118
	TXI	119	01DCh	Edge	✓	✓	✓	N/A	N/A	IER0E.IEN7	IPR119	DTCER119
DMAC	DMAC0I	120	01E0h	Edge	✓	✓	N/A	N/A	N/A	IER0F.IEN0	IPR120	DTCER120
	DMAC1I	121	01E4h	Edge	✓	✓	N/A	N/A	N/A	IER0F.IEN1	IPR121	DTCER121
	DMAC2I	122	01E8h	Edge	✓	✓	N/A	N/A	N/A	IER0F.IEN2	IPR122	DTCER122
	DMAC3I	123	01ECh	Edge	✓	✓	N/A	N/A	N/A	IER0F.IEN3	IPR123	DTCER123
	DMAC74I	124	01F0h	Level	✓	N/A	N/A	N/A	N/A	IER0F.IEN4	IPR124	—

Table 15.5 Interrupt Vector Table (4/6)

Interrupt Request is Generated By	Name	Vector No.	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	Exit from ACS	IER	IPR	DTCER
OST	OSTDI*2	125	01F4h	Edge	✓	N/A	N/A	N/A	N/A	IER0F.IEN5	IPR125	—
EXDMAC	EXDMAC0I	126	01F8h	Edge	✓	✓	N/A	N/A	N/A	IER0F.IEN6	IPR126	DTCER126
	EXDMAC1I	127	01FCh	Edge	✓	✓	N/A	N/A	N/A	IER0F.IEN7	IPR127	DTCER127
PERIB (software configurable interrupt B *3)	INTB128	128	0200h	Edge	✓	✓	✓	N/A	N/A	IER10.IEN0	IPR128	DTCER128
	INTB129	129	0204h	Edge	✓	✓	✓	N/A	N/A	IER10.IEN1	IPR129	DTCER129
	INTB130	130	0208h	Edge	✓	✓	✓	N/A	N/A	IER10.IEN2	IPR130	DTCER130
	INTB131	131	020Ch	Edge	✓	✓	✓	N/A	N/A	IER10.IEN3	IPR131	DTCER131
	INTB132	132	0210h	Edge	✓	✓	✓	N/A	N/A	IER10.IEN4	IPR132	DTCER132
	INTB133	133	0214h	Edge	✓	✓	✓	N/A	N/A	IER10.IEN5	IPR133	DTCER133
	INTB134	134	0218h	Edge	✓	✓	✓	N/A	N/A	IER10.IEN6	IPR134	DTCER134
	INTB135	135	021Ch	Edge	✓	✓	✓	N/A	N/A	IER10.IEN7	IPR135	DTCER135
	INTB136	136	0220h	Edge	✓	✓	✓	N/A	N/A	IER11.IEN0	IPR136	DTCER136
	INTB137	137	0224h	Edge	✓	✓	✓	N/A	N/A	IER11.IEN1	IPR137	DTCER137
	INTB138	138	0228h	Edge	✓	✓	✓	N/A	N/A	IER11.IEN2	IPR138	DTCER138
	INTB139	139	022Ch	Edge	✓	✓	✓	N/A	N/A	IER11.IEN3	IPR139	DTCER139
	INTB140	140	0230h	Edge	✓	✓	✓	N/A	N/A	IER11.IEN4	IPR140	DTCER140
	INTB141	141	0234h	Edge	✓	✓	✓	N/A	N/A	IER11.IEN5	IPR141	DTCER141
	INTB142	142	0238h	Edge	✓	✓	✓	N/A	N/A	IER11.IEN6	IPR142	DTCER142
	INTB143	143	023Ch	Edge	✓	✓	✓	N/A	N/A	IER11.IEN7	IPR143	DTCER143
	INTB144	144*4	0240h	Edge	✓	✓	✓	N/A	N/A	IER12.IEN0	IPR144	DTCER144
	INTB145	145*5	0244h	Edge	✓	✓	✓	N/A	N/A	IER12.IEN1	IPR145	DTCER145
	INTB146	146	0248h	Edge	✓	✓	✓	N/A	✓	IER12.IEN2	IPR146	DTCER146
	INTB147	147	024Ch	Edge	✓	✓	✓	N/A	✓	IER12.IEN3	IPR147	DTCER147
	INTB148	148	0250h	Edge	✓	✓	✓	N/A	✓	IER12.IEN4	IPR148	DTCER148
	INTB149	149	0254h	Edge	✓	✓	✓	N/A	✓	IER12.IEN5	IPR149	DTCER149
	INTB150	150	0258h	Edge	✓	✓	✓	N/A	✓	IER12.IEN6	IPR150	DTCER150
	INTB151	151	025Ch	Edge	✓	✓	✓	N/A	✓	IER12.IEN7	IPR151	DTCER151
	INTB152	152	0260h	Edge	✓	✓	✓	N/A	✓	IER13.IEN0	IPR152	DTCER152
	INTB153	153	0264h	Edge	✓	✓	✓	N/A	✓	IER13.IEN1	IPR153	DTCER153
	INTB154	154	0268h	Edge	✓	✓	✓	N/A	✓	IER13.IEN2	IPR154	DTCER154
	INTB155	155	026Ch	Edge	✓	✓	✓	N/A	✓	IER13.IEN3	IPR155	DTCER155
	INTB156	156	0270h	Edge	✓	✓	✓	N/A	✓	IER13.IEN4	IPR156	DTCER156
	INTB157	157	0274h	Edge	✓	✓	✓	N/A	✓	IER13.IEN5	IPR157	DTCER157
	INTB158	158	0278h	Edge	✓	✓	✓	N/A	N/A	IER13.IEN6	IPR158	DTCER158
	INTB159	159	027Ch	Edge	✓	✓	✓	N/A	N/A	IER13.IEN7	IPR159	DTCER159
	INTB160	160	0280h	Edge	✓	✓	✓	N/A	N/A	IER14.IEN0	IPR160	DTCER160
	INTB161	161	0284h	Edge	✓	✓	✓	N/A	N/A	IER14.IEN1	IPR161	DTCER161
	INTB162	162	0288h	Edge	✓	✓	✓	N/A	N/A	IER14.IEN2	IPR162	DTCER162
	INTB163	163	028Ch	Edge	✓	✓	✓	N/A	N/A	IER14.IEN3	IPR163	DTCER163
	INTB164	164	0290h	Edge	✓	✓	✓	N/A	N/A	IER14.IEN4	IPR164	DTCER164
	INTB165	165	0294h	Edge	✓	✓	✓	N/A	N/A	IER14.IEN5	IPR165	DTCER165
	INTB166	166	0298h	Edge	✓	✓	✓	N/A	N/A	IER14.IEN6	IPR166	DTCER166
	INTB167	167	029Ch	Edge	✓	✓	✓	N/A	N/A	IER14.IEN7	IPR167	DTCER167
	INTB168	168	02A0h	Edge	✓	✓	✓	N/A	N/A	IER15.IEN0	IPR168	DTCER168
	INTB169	169	02A4h	Edge	✓	✓	✓	N/A	N/A	IER15.IEN1	IPR169	DTCER169

Table 15.5 Interrupt Vector Table (5/6)

Interrupt Request is Generated By	Name	Vector No.	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	Exit from ACS	IER	IPR	DTCER
PERIB (software configurable interrupt B *3)	INTB170	170	02A8h	Edge	✓	✓	✓	N/A	N/A	IER15.IEN2	IPR170	DTCER170
	INTB171	171	02ACh	Edge	✓	✓	✓	N/A	N/A	IER15.IEN3	IPR171	DTCER171
	INTB172	172	02B0h	Edge	✓	✓	✓	N/A	N/A	IER15.IEN4	IPR172	DTCER172
	INTB173	173	02B4h	Edge	✓	✓	✓	N/A	N/A	IER15.IEN5	IPR173	DTCER173
	INTB174	174	02B8h	Edge	✓	✓	✓	N/A	N/A	IER15.IEN6	IPR174	DTCER174
	INTB175	175	02BCh	Edge	✓	✓	✓	N/A	N/A	IER15.IEN7	IPR175	DTCER175
	INTB176	176	02C0h	Edge	✓	✓	✓	N/A	N/A	IER16.IEN0	IPR176	DTCER176
	INTB177	177	02C4h	Edge	✓	✓	✓	N/A	N/A	IER16.IEN1	IPR177	DTCER177
	INTB178	178	02C8h	Edge	✓	✓	✓	N/A	N/A	IER16.IEN2	IPR178	DTCER178
	INTB179	179	02CCh	Edge	✓	✓	✓	N/A	N/A	IER16.IEN3	IPR179	DTCER179
	INTB180	180	02D0h	Edge	✓	✓	✓	N/A	N/A	IER16.IEN4	IPR180	DTCER180
	INTB181	181	02D4h	Edge	✓	✓	✓	N/A	N/A	IER16.IEN5	IPR181	DTCER181
	INTB182	182	02D8h	Edge	✓	✓	✓	N/A	N/A	IER16.IEN6	IPR182	DTCER182
	INTB183	183	02DCh	Edge	✓	✓	✓	N/A	N/A	IER16.IEN7	IPR183	DTCER183
	INTB184	184	02E0h	Edge	✓	✓	✓	N/A	N/A	IER17.IEN0	IPR184	DTCER184
	INTB185	185	02E4h	Edge	✓	✓	✓	N/A	N/A	IER17.IEN1	IPR185	DTCER185
	INTB186	186	02E8h	Edge	✓	✓	✓	N/A	N/A	IER17.IEN2	IPR186	DTCER186
	INTB187	187	02ECh	Edge	✓	✓	✓	N/A	N/A	IER17.IEN3	IPR187	DTCER187
	INTB188	188	02F0h	Edge	✓	✓	✓	N/A	N/A	IER17.IEN4	IPR188	DTCER188
	INTB189	189	02F4h	Edge	✓	✓	✓	N/A	N/A	IER17.IEN5	IPR189	DTCER189
	INTB190	190	02F8h	Edge	✓	✓	✓	N/A	N/A	IER17.IEN6	IPR190	DTCER190
	INTB191	191	02FCh	Edge	✓	✓	✓	N/A	N/A	IER17.IEN7	IPR191	DTCER191
	INTB192	192	0300h	Edge	✓	✓	✓	N/A	N/A	IER18.IEN0	IPR192	DTCER192
	INTB193	193	0304h	Edge	✓	✓	✓	N/A	N/A	IER18.IEN1	IPR193	DTCER193
	INTB194	194	0308h	Edge	✓	✓	✓	N/A	N/A	IER18.IEN2	IPR194	DTCER194
	INTB195	195	030Ch	Edge	✓	✓	✓	N/A	N/A	IER18.IEN3	IPR195	DTCER195
	INTB196	196	0310h	Edge	✓	✓	✓	N/A	N/A	IER18.IEN4	IPR196	DTCER196
	INTB197	197	0314h	Edge	✓	✓	✓	N/A	N/A	IER18.IEN5	IPR197	DTCER197
	INTB198	198	0318h	Edge	✓	✓	✓	N/A	N/A	IER18.IEN6	IPR198	DTCER198
	INTB199	199	031Ch	Edge	✓	✓	✓	N/A	N/A	IER18.IEN7	IPR199	DTCER199
	INTB200	200	0320h	Edge	✓	✓	✓	N/A	N/A	IER19.IEN0	IPR200	DTCER200
	INTB201	201	0324h	Edge	✓	✓	✓	N/A	N/A	IER19.IEN1	IPR201	DTCER201
	INTB202	202	0328h	Edge	✓	✓	✓	N/A	N/A	IER19.IEN2	IPR202	DTCER202
	INTB203	203	032Ch	Edge	✓	✓	✓	N/A	N/A	IER19.IEN3	IPR203	DTCER203
	INTB204	204	0330h	Edge	✓	✓	✓	N/A	N/A	IER19.IEN4	IPR204	DTCER204
	INTB205	205	0334h	Edge	✓	✓	✓	N/A	N/A	IER19.IEN5	IPR205	DTCER205
	INTB206	206	0338h	Edge	✓	✓	✓	N/A	N/A	IER19.IEN6	IPR206	DTCER206
	INTB207	207	033Ch	Edge	✓	✓	✓	N/A	N/A	IER19.IEN7	IPR207	DTCER207
PERIA (software configurable interrupt A *6)	INTA208	208*4	0340h	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN0	IPR208	DTCER208
	INTA209	209*5	0344h	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN1	IPR209	DTCER209
	INTA210	210	0348h	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN2	IPR210	DTCER210
	INTA211	211	034Ch	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN3	IPR211	DTCER211
	INTA212	212	0350h	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN4	IPR212	DTCER212
	INTA213	213	0354h	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN5	IPR213	DTCER213
	INTA214	214	0358h	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN6	IPR214	DTCER214

Table 15.5 Interrupt Vector Table (6/6)

Interrupt Request is Generated By	Name	Vector No.	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start the DTC	Start the DMAC	Exit from SSBY	Exit from ACS	IER	IPR	DTCER
PERIA (software configurable interrupt A *6)	INTA215	215	035Ch	Edge	✓	✓	✓	N/A	N/A	IER1A.IEN7	IPR215	DTCER215
	INTA216	216	0360h	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN0	IPR216	DTCER216
	INTA217	217	0364h	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN1	IPR217	DTCER217
	INTA218	218	0368h	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN2	IPR218	DTCER218
	INTA219	219	036Ch	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN3	IPR219	DTCER219
	INTA220	220	0370h	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN4	IPR220	DTCER220
	INTA221	221	0374h	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN5	IPR221	DTCER221
	INTA222	222	0378h	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN6	IPR222	DTCER222
	INTA223	223	037Ch	Edge	✓	✓	✓	N/A	N/A	IER1B.IEN7	IPR223	DTCER223
	INTA224	224	0380h	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN0	IPR224	DTCER224
	INTA225	225	0384h	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN1	IPR225	DTCER225
	INTA226	226	0388h	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN2	IPR226	DTCER226
	INTA227	227	038Ch	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN3	IPR227	DTCER227
	INTA228	228	0390h	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN4	IPR228	DTCER228
	INTA229	229	0394h	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN5	IPR229	DTCER229
	INTA230	230	0398h	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN6	IPR230	DTCER230
	INTA231	231	039Ch	Edge	✓	✓	✓	N/A	N/A	IER1C.IEN7	IPR231	DTCER231
	INTA232	232	03A0h	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN0	IPR232	DTCER232
	INTA233	233	03A4h	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN1	IPR233	DTCER233
	INTA234	234	03A8h	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN2	IPR234	DTCER234
	INTA235	235	03ACh	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN3	IPR235	DTCER235
	INTA236	236	03B0h	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN4	IPR236	DTCER236
	INTA237	237	03B4h	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN5	IPR237	DTCER237
	INTA238	238	03B8h	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN6	IPR238	DTCER238
	INTA239	239	03BCh	Edge	✓	✓	✓	N/A	N/A	IER1D.IEN7	IPR239	DTCER239
	INTA240	240	03C0h	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN0	IPR240	DTCER240
	INTA241	241	03C4h	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN1	IPR241	DTCER241
	INTA242	242	03C8h	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN2	IPR242	DTCER242
	INTA243	243	03CCh	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN3	IPR243	DTCER243
	INTA244	244	03D0h	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN4	IPR244	DTCER244
	INTA245	245	03D4h	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN5	IPR245	DTCER245
	INTA246	246	03D8h	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN6	IPR246	DTCER246
	INTA247	247	03DCh	Edge	✓	✓	✓	N/A	N/A	IER1E.IEN7	IPR247	DTCER247
	INTA248	248	03E0h	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN0	IPR248	DTCER248
	INTA249	249	03E4h	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN1	IPR249	DTCER249
	INTA250	250	03E8h	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN2	IPR250	DTCER250
	INTA251	251	03ECh	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN3	IPR251	DTCER251
	INTA252	252	03F0h	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN4	IPR252	DTCER252
	INTA253	253	03F4h	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN5	IPR253	DTCER253
	INTA254	254	03F8h	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN6	IPR254	DTCER254
	INTA255	255	03FCh	Edge	✓	✓	✓	N/A	N/A	IER1F.IEN7	IPR255	DTCER255

Note: This table lists the interrupt vectors for the maximum specification. The interrupt vectors for individual products correspond to the functions listed in Table 1.2. For details, refer to Table 1.2, Comparison of Functions for Different Packages.
Note 1. For the group interrupt sources, refer to Table 15.7, Group Interrupt Requests.
Note 2. This is the case where the corresponding non-maskable interrupt enable bit is set to 0 (disabled).
Note 3. For the software configurable interrupt B sources, refer to Table 15.3, Interrupt Sources for Software Configurable Interrupt B.

Note that some interrupt sources cannot start the DTC, DMAC, or EXDMAC.

Note 4. This can be used to start EXDMAC0. Set the SELEXDR.SELEXD0 bit to 0 when using the interrupt source assigned to interrupt vector number 144 as a trigger. Set the SELEXDR.SELEXD0 bit to 1 when using the interrupt source assigned to interrupt vector number 208 as a trigger. Refer to section 15.2.29, Software Configurable Interrupt B Source Select Register n (SLIBRn) (n = 144 to 207), section 15.2.30, Software Configurable Interrupt A Source Select Register n (SLIARn) (n = 208 to 255), and section 15.2.31, EXDMAC Trigger Select Register (SELEXDR).

Note 5. This can be used to start EXDMAC1. Set the SELEXDR.SELEXD1 bit to 0 when using the interrupt source assigned to interrupt vector number 145 as a trigger. Set the SELEXDR.SELEXD1 bit to 1 when using the interrupt source assigned to interrupt vector number 209 as a trigger.

Note 6. For the software configurable interrupt A sources, refer to Table 15.4, Interrupt Sources for Software Configurable Interrupt A. Note that some interrupt sources cannot start the DTC, DMAC, or EXDMAC.

15.3.2 Fast Interrupt Vector Area

The interrupt set as the fast interrupt uses the FINTV register in the CPU. Set the FINTV register before enabling the fast interrupt.

15.3.3 Non-maskable Interrupt Vector Area

Non-maskable interrupts use the vector area in the exception vector table.

The exception vector table is allocated to the 128-byte area (4 bytes × 32 sources) beginning with the address set in the EXTB register in the CPU. Set the EXTB register before enabling non-maskable interrupts. Set a multiple of 4 in the EXTB register.

15.4 Types of Interrupts

Interrupts are divided into maskable interrupts and non-maskable interrupts. Maskable interrupts can be masked by the PSW.I bit or IPL[3:0] bits of the processor status word in the CPU. Non-maskable interrupts can be accepted by the CPU regardless of those bits. While interrupt sources assigned to vector numbers 0 to 127 are fixed, an interrupt source assigned to each vector number from 128 to 255 (software configurable interrupt) can be selected from multiple sources. Note that maskable interrupts are referred to as interrupts in this chapter.
Figure 15.2 shows types of interrupts.

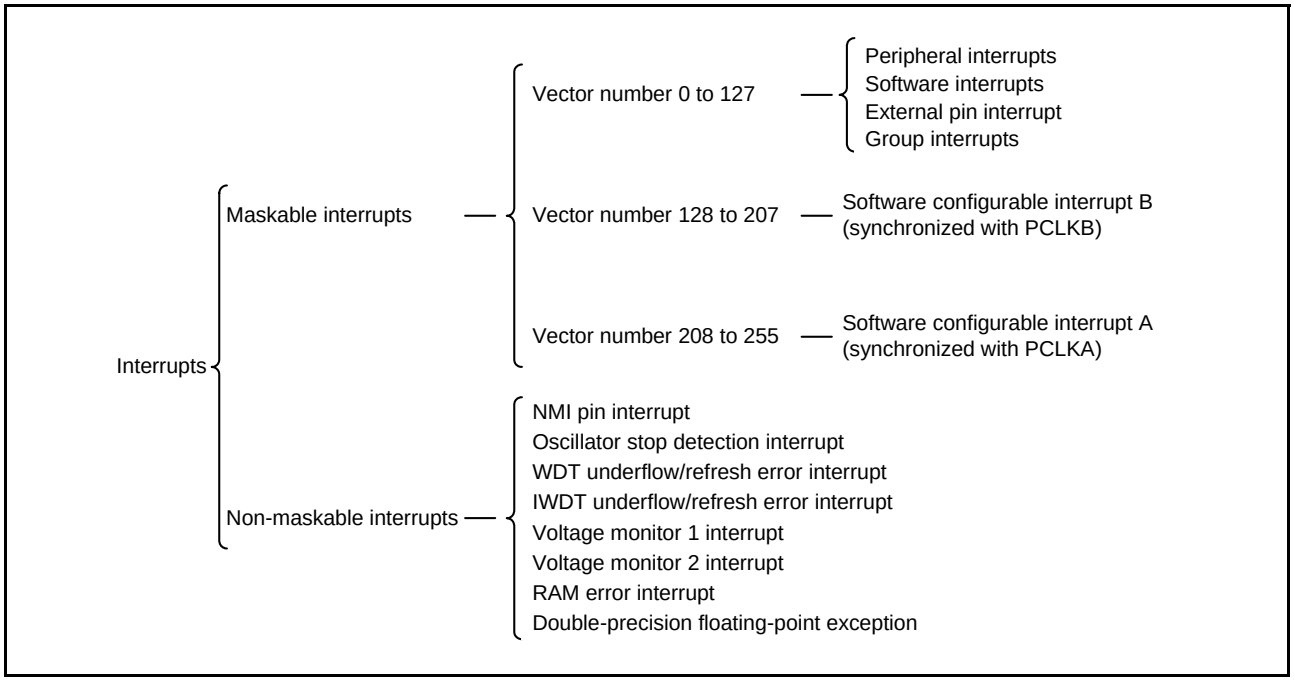


Figure 15.2 Types of Interrupts

15.4.1 Peripheral Interrupts

Peripheral interrupt is generated by peripherals. Peripheral interrupts sources assigned to vector numbers 0 to 127 cannot be assigned to the software configurable interrupts. Refer to section 15.4.5, Software Configurable Interrupts for details on software configurable interrupts.

15.4.2 Software Interrupts

When the SWINTR.SWINT bit and SWINT2R.SWINT2 bit are set to 1, the SWINT interrupt and SWINT2 interrupt occur, respectively.

15.4.3 External Pin Interrupt

An external pin interrupt is generated by signals input to the IRQi pin (i = 0 to 15). Refer to section 15.7.4, Setting the External Pin Interrupt for the procedure to set the external pin interrupt.

15.4.4 Group Interrupts

Multiple peripheral interrupt requests (up to 32 requests) are grouped together as one interrupt request. Interrupts are grouped depending on the peripheral operating clock (ICLK, PCLKB, or PCLKA) and method to detect interrupt requests (edge detection or level detection).

(1) Types of Group Interrupts

Table 15.6 lists types of group interrupts.

Table 15.6 Types of Group Interrupts

Interrupt Vector Number	Interrupt Name	Group Interrupt Source	
		Peripheral operating clock	Interrupt detection method
17	GROUPIE0	ICLK	Edge detection
106	GROUPBE0	PCLKB	
110	GROUPBL0		Level detection
111	GROUPBL1		
107	GROUPBL2		
112	GROUPAL0	PCLKA	
113	GROUPAL1		

(2) Configuration of Group Interrupts

When an interrupt request is generated while the corresponding ENj bit in the group interrupt request enable register (GENIE0, GENBE0, GENBL0, GENBL1, GENBL2, GENAL0, GENAL1*1) is 1, the ISj flag in the group interrupt request register (GRPIE0, GRPBE0, GRPBL0, GRPBL1, GRPBL2, GRPAL0, GRPAL1*1) becomes 1 (j = 0 to 31). Figure 15.3 shows the configuration of group interrupts.

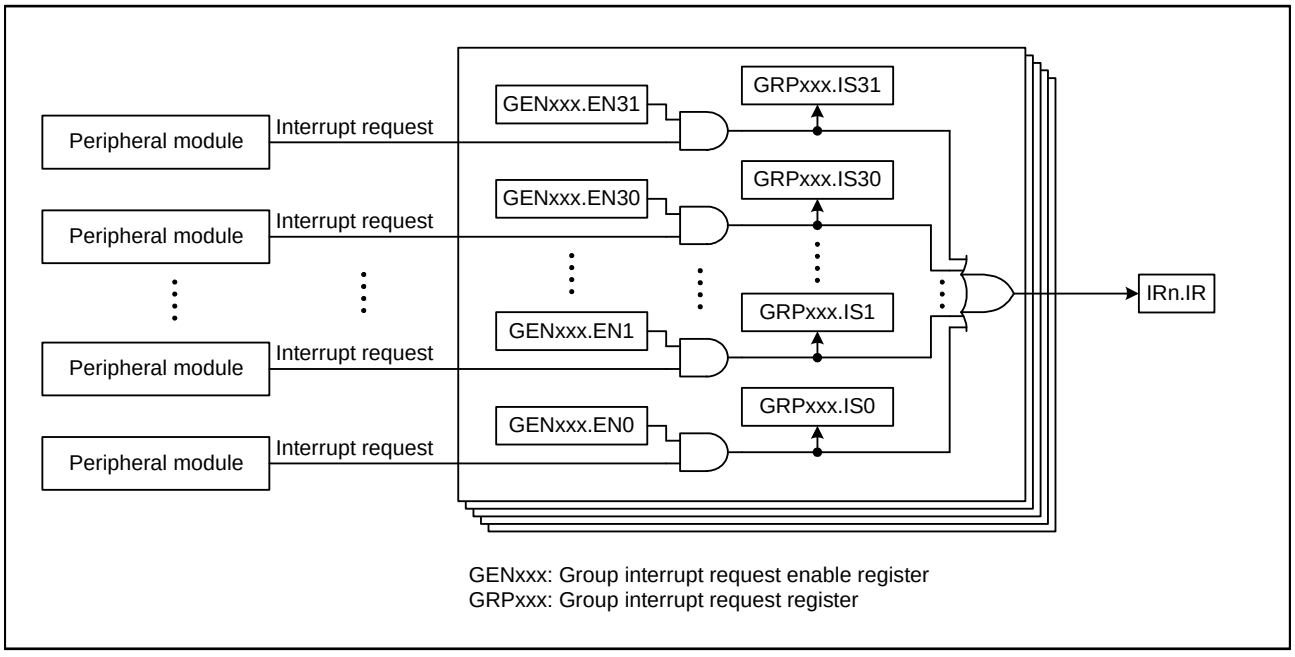


Figure 15.3 Group Interrupt Configuration (n = 17, 106, 107, 110 to 113)

Note 1. There is no register in the group to which no interrupt source is assigned.

(3) Group Interrupt Sources

Table 15.7 lists peripheral interrupt sources that are assigned to group interrupts.

Table 15.7 Group Interrupt Requests (1/3)

Group	No.	Interrupt Request Source	Name	Interrupt Request Enable Bit	Interrupt Status Flag	Interrupt Source Clear Bit	Vector No. (IRn.IR)
IE0	0	DPFPU	DPFPUEX (double-precision floating-point exception)	GENIE0.EN0	GRPIE0.IS0	GCRIE0.CLR0	17
	1 to 31	Reserved	—	—	—	—	
BE0	0	CAN0	ERS0 (error interrupt)	GENBE0.EN0	GRPBE0.IS0	GCRBE0.CLR0	106
	1	CAN1	ERS1 (error interrupt)	GENBE0.EN1	GRPBE0.IS1	GCRBE0.CLR1	
	2 to 31	Reserved	—	—	—	—	
BL0	0	SCI0	TEI0 (transmission end)	GENBL0.EN0	GRPBL0.IS0	—	110
	1		ERI0 (receive error)	GENBL0.EN1	GRPBL0.IS1	—	
	2	SCI1	TEI1 (transmission end)	GENBL0.EN2	GRPBL0.IS2	—	
	3		ERI1 (receive error)	GENBL0.EN3	GRPBL0.IS3	—	
	4	SCI2	TEI2 (transmission end)	GENBL0.EN4	GRPBL0.IS4	—	
	5		ERI2 (receive error)	GENBL0.EN5	GRPBL0.IS5	—	
	6	SCI3	TEI3 (transmission end)	GENBL0.EN6	GRPBL0.IS6	—	
	7		ERI3 (receive error)	GENBL0.EN7	GRPBL0.IS7	—	
	8	SCI4	TEI4 (transmission end)	GENBL0.EN8	GRPBL0.IS8	—	
	9		ERI4 (receive error)	GENBL0.EN9	GRPBL0.IS9	—	
	10	SCI5	TEI5 (transmission end)	GENBL0.EN10	GRPBL0.IS10	—	
	11		ERI5 (receive error)	GENBL0.EN11	GRPBL0.IS11	—	
	12	SCI6	TEI6 (transmission end)	GENBL0.EN12	GRPBL0.IS12	—	
	13		ERI6 (receive error)	GENBL0.EN13	GRPBL0.IS13	—	
	14	SCI7	TEI7 (transmission end)	GENBL0.EN14	GRPBL0.IS14	—	
	15		ERI7 (receive error)	GENBL0.EN15	GRPBL0.IS15	—	
	16	SCI12	TEI12 (transmission end)	GENBL0.EN16	GRPBL0.IS16	—	
	17		ERI12 (receive error)	GENBL0.EN17	GRPBL0.IS17	—	
	18		SCIX0 (Break Field Low width detection)	GENBL0.EN18	GRPBL0.IS18	—	
	19		SCIX1 (Control Field 0 match) (Control Field 1 match) (priority interrupt bit detection)	GENBL0.EN19	GRPBL0.IS19	—	
	20		SCIX2 (bus collision detection)	GENBL0.EN20	GRPBL0.IS20	—	
	21		SCIX3 (valid edge detection)	GENBL0.EN21	GRPBL0.IS21	—	
	22 to 25	Reserved	—	—	—	—	
	26	CAC	FERRI (frequency error)	GENBL0.EN26	GRPBL0.IS26	—	
	27		MENDI (measurement end)	GENBL0.EN27	GRPBL0.IS27	—	
	28		OVFI (overflow interrupt)	GENBL0.EN28	GRPBL0.IS28	—	
	29	DOC	DOPCI (data operation circuit interrupt)	GENBL0.EN29	GRPBL0.IS29	—	
	30, 31	Reserved	—	—	—	—	

Table 15.7 Group Interrupt Requests (2/3)

Group	No.	Interrupt Request Source	Name	Interrupt Request Enable Bit	Interrupt Status Flag	Interrupt Source Clear Bit	Vector No. (IRn.IR)
BL1	0 to 2	Reserved	—	—	—	—	111
	3	SDHI	CDETI (card detection interrupt)	GENBL1.EN3	GRPBL1.IS3	—	
	4		CACI (card access interrupt)	GENBL1.EN4	GRPBL1.IS4	—	
	5		SDACI (SDIO access interrupt)	GENBL1.EN5	GRPBL1.IS5	—	
	6 to 8	Reserved	—	—	—	—	
	9	POE3	OEI1 (output enable interrupt 1)	GENBL1.EN9	GRPBL1.IS9	—	
	10		OEI2 (output enable interrupt 2)	GENBL1.EN10	GRPBL1.IS10	—	
	11		OEI3 (output enable interrupt 3)	GENBL1.EN11	GRPBL1.IS11	—	
	12		OEI4 (output enable interrupt 4)	GENBL1.EN12	GRPBL1.IS12	—	
	13	RIIC0	TEI0 (transmission end)	GENBL1.EN13	GRPBL1.IS13	—	
	14		EEI0 (communication error/ communication event)	GENBL1.EN14	GRPBL1.IS14	—	
	15	RIIC2	TEI2 (transmission end)	GENBL1.EN15	GRPBL1.IS15	—	
	16		EEI2 (communication error/ communication event)	GENBL1.EN16	GRPBL1.IS16	—	
	17	SSIE0	SSIF0 (status interrupt)	GENBL1.EN17	GRPBL1.IS17	—	
	18, 19	Reserved	—	—	—	—	
	20	S12AD	S12CMPAI (compare interrupt)	GENBL1.EN20	GRPBL1.IS20	—	
	21		S12CMPBI (compare interrupt)	GENBL1.EN21	GRPBL1.IS21	—	
	22	S12AD1	S12CMPAI1 (compare interrupt)	GENBL1.EN22	GRPBL1.IS22	—	
	23		S12CMPBI1 (compare interrupt)	GENBL1.EN23	GRPBL1.IS23	—	
	24	SCI8	TEI8 (transmission end)	GENBL1.EN24	GRPBL1.IS24	—	
	25		ERI8 (receive error)	GENBL1.EN25	GRPBL1.IS25	—	
	26	SCI9	TEI9 (transmission end)	GENBL1.EN26	GRPBL1.IS26	—	
	27		ERI9 (receive error)	GENBL1.EN27	GRPBL1.IS27	—	
	28	RIIC1	TEI1 (transmission end)	GENBL1.EN28	GRPBL1.IS28	—	
	29		EEI1 (communication error/ communication event)	GENBL1.EN29	GRPBL1.IS29	—	
	30, 31	Reserved	—	—	—	—	
BL2	0 to 31	Reserved	—	—	—	—	107
AL0	0 to 7	Reserved	—	—	—	—	112
	8	SCI10	TEI10 (transmission end)	GENAL0.EN8	GRPAL0.IS8	—	
	9		ERI10 (receive error)	GENAL0.EN9	GRPAL0.IS9	—	
	10, 11	Reserved	—	—	—	—	
	12	SCI11	TEI11 (transmission end)	GENAL0.EN12	GRPAL0.IS12	—	
	13		ERI11 (receive error)	GENAL0.EN13	GRPAL0.IS13	—	
	14, 15	Reserved	—	—	—	—	
	16	RSPI0	SPII0 (idle interrupt)	GENAL0.EN16	GRPAL0.IS16	—	
	17		SPEI0 (error interrupt)	GENAL0.EN17	GRPAL0.IS17	—	
	18	RSPI1	SPII1 (idle interrupt)	GENAL0.EN18	GRPAL0.IS18	—	
	19		SPEI1 (error interrupt)	GENAL0.EN19	GRPAL0.IS19	—	
	20	RSPI2	SPII2 (idle interrupt)	GENAL0.EN20	GRPAL0.IS20	—	
	21		SPEI2 (error interrupt)	GENAL0.EN21	GRPAL0.IS21	—	
	22	RSPIA0	SPII (idle interrupt)	GENAL0.EN22	GRPAL0.IS22	—	
	23		SPEI (error interrupt)	GENAL0.EN23	GRPAL0.IS23	—	

Table 15.7 Group Interrupt Requests (3/3)

Group	No.	Interrupt Request Source	Name	Interrupt Request Enable Bit	Interrupt Status Flag	Interrupt Source Clear Bit	Vector No. (IRn.IR)
AL0	24	RSCI10	TEI (transmission end/condition generated)	GENAL0.EN24	GRPAL0.IS24	—	112
	25		ERI (receive error)	GENAL0.EN25	GRPAL0.IS25	—	
	26		BFD (break field detect)	GENAL0.EN26	GRPAL0.IS26	—	
	27	RSCI11	TEI (transmission end/condition generated)	GENAL0.EN27	GRPAL0.IS27	—	
	28		ERI (receive error)	GENAL0.EN28	GRPAL0.IS28	—	
	29		BFD (break field detect)	GENAL0.EN29	GRPAL0.IS29	—	
	30	QSPIX	ERI (ROM access error)	GENAL0.EN30	GRPAL0.IS30	—	
	31	Reserved	—	—	—	—	
AL1	0 to 11	Reserved	—	—	—	—	113
	12	RIICHS0	TEI (transmission end)	GENAL1.EN12	GRPAL1.IS12	—	
	13		EEI (communication error/ communication event)	GENAL1.EN13	GRPAL1.IS13	—	
	14 to 31	Reserved	—	—	—	—	

15.4.5 Software Configurable Interrupts

An interrupt source assigned to each interrupt vector number from 128 to 255 can be selected from multiple sources. They are divided into software configurable interrupt B and software configurable interrupt A depending on the peripheral operating clock. Figure 15.4 shows the software configurable interrupt configuration.

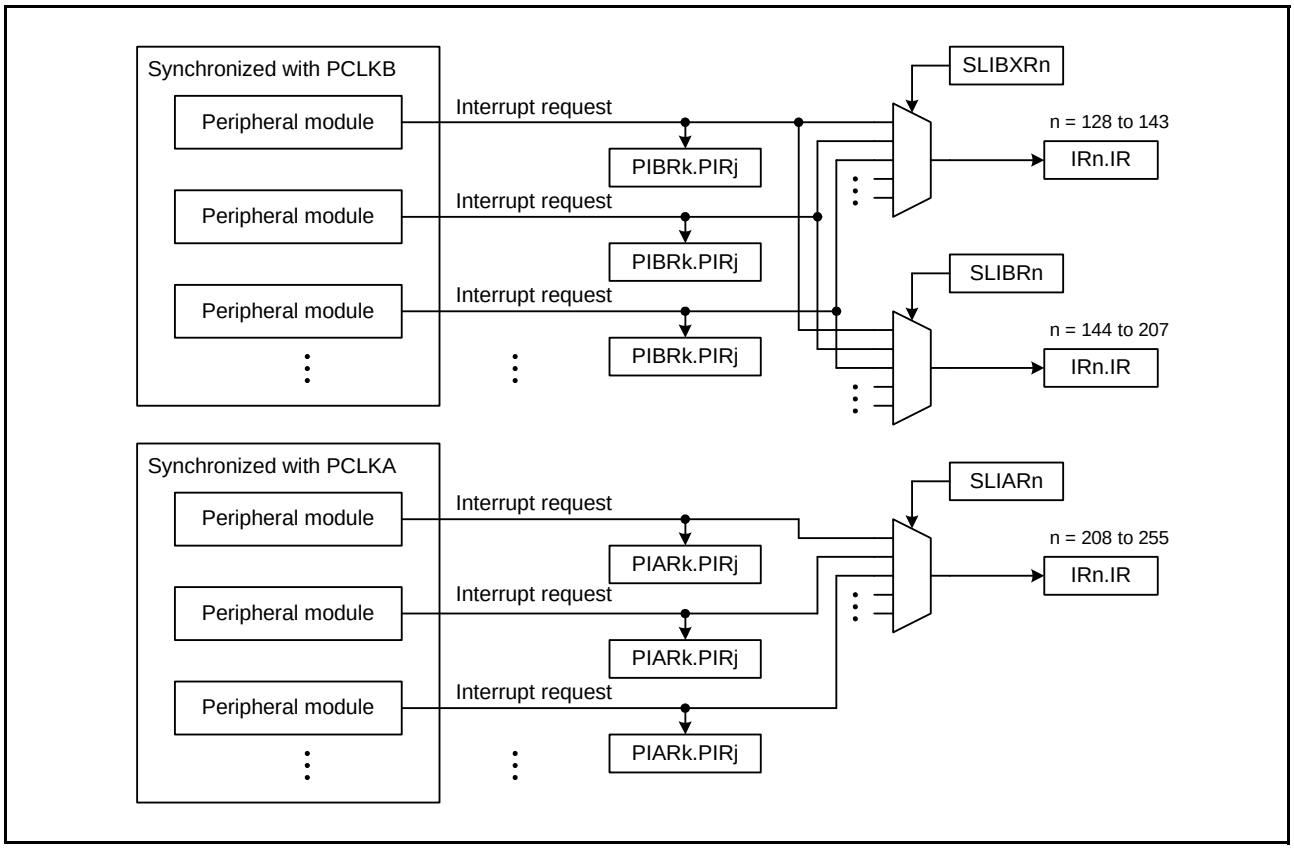


Figure 15.4 Software Configurable Interrupt Configuration

15.4.5.1 Software Configurable Interrupt B

For interrupt sources assigned to software configurable interrupts, interrupt sources of peripherals that operates in synchronization with PCLKB can be assigned to interrupt number from 128 to 207. The abbreviation for software configurable interrupt B is PERIB. Interrupt names are indicated by INTB128 to INTB207.
Refer to Table 15.3, Interrupt Sources for Software Configurable Interrupt B for interrupt sources that can be assigned to software configurable interrupt B.

15.4.5.2 Software Configurable Interrupt A

For interrupt sources assigned to software configurable interrupts, interrupt sources of peripherals that operates in synchronization with PCLKA can be assigned to interrupt number from 208 to 255. The abbreviation for software configurable interrupt A is PERIA. Interrupt names are indicated by INTA208 to INTA255.
Refer to Table 15.4, Interrupt Sources for Software Configurable Interrupt A for interrupt sources that can be assigned for software configurable interrupt A.

15.4.5.3 EXDMAC Start Trigger by Software Configurable Interrupts

Some interrupt sources assigned to software configurable interrupt B and software configurable interrupt A can be used as a trigger for EXDMAC0 or EXDMAC1.
To use an interrupt source as the EXDMAC0 trigger, assign the interrupt source to interrupt vector number 144 (INTB144) or 208 (INTA208). To use an interrupt source as the EXDMAC1 trigger, assign the interrupt source to interrupt vector number 145 (INTB145) or 209 (INTA209).
Only the TPU1.TGI1A interrupt of software configurable interrupt B and the MTU1.TGIA1 interrupt of software configurable interrupt A can be used as the EXDMAC trigger.
Triggers for EXDMAC0 and EXDMAC1 can be selected by setting bits SELEXDR.SELEXD0 and SELEXD1, respectively. Table 15.8 lists correspondences between the SELEXDR register settings and interrupt sources to start the EXDMAC.

Table 15.8 EXDMAC Start Triggers by Software Configurable Interrupts

SELEXDR Register	Bit Value	Interrupt Vector Number	Interrupt Request
SELEXD0 bit (EXDMAC0)	0	144 (INTB144)	TPU1.TGI1A (input capture/compare match)
	1	208 (INTA208)	MTU1.TGIA1 (input capture/compare match)
SELEXD1 bit (EXDMAC1)	0	145 (INTB145)	TPU1.TGI1A (input capture/compare match)
	1	209 (INTA209)	MTU1.TGIA1 (input capture/compare match)

15.4.6 Non-Maskable Interrupts

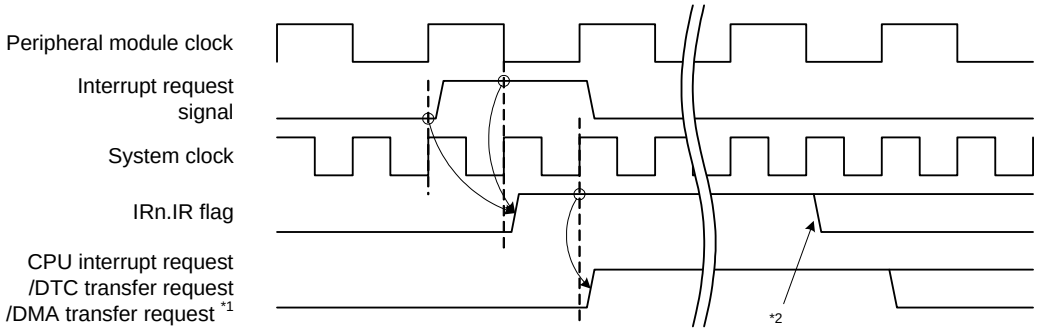
Non-maskable interrupts include the NMI pin interrupt, oscillation stop detection interrupt, WDT underflow/refresh error interrupt, IWDT underflow/refresh error interrupt, voltage monitoring 1 interrupt, voltage monitoring 2 interrupt, RAM error interrupt, and double-precision floating-point exception.
Non-maskable interrupts have the highest priority in all interrupts, including the fast interrupt, so are accepted regardless of the settings of the PSW.I bit (interrupt enable) and IPL[3:0] bits (processor interrupt priority level) in the CPU.
The NMISR register can be used to confirm whether a non-maskable interrupt is generated.
Only the CPU can be selected as the interrupt request destination of non-maskable interrupt. The DTC and DMAC cannot be selected.

15.5 Interrupt Detection

Level detection or edge detection can be used for detecting an interrupt request.
For interrupt requests from the peripherals, edge detection or level detection is fixed for each interrupt source. For interrupt requests of the external pin interrupt, edge detection or level detection can be selected with the IRQCRi.IRQMD[1:0] bits (i = 0 to 15).
Refer to Table 15.5, Interrupt Vector Table for the method to detect each interrupt request.
For group interrupts, interrupt sources are grouped depending on the method to detect interrupt requests.
Interrupt requests by interrupt sources assigned to groups IE0 and BE0 are detected by edge detection. Interrupt requests by interrupt sources assigned to groups BL0, BL1, BL2, AL0, and AL1 are detected by level detection. Note that group interrupts (GROUPIE0, GROUPBE0, GROUPBL0, GROUPBL1, GROUPBL2, GROUPAL0, GROUPAL1) are detected by level detection.
Refer to section 15.4.4, Group Interrupts for group interrupts. Refer to section 15.5.3, Group Interrupts Using Edge Detection and section 15.5.4, Group Interrupts Using Level Detection for interrupt requests of group interrupts.

15.5.1 Edge Detection

Figure 15.5 shows the operation of the IRn.IR flag at edge detection (n = 023 to 255).
The IRn.IR flag becomes 1 when the rising edge of the interrupt request signal is detected. Then, the IRn.IR flag does not become 0 by disabling the interrupt request of the peripheral module. When the CPU accepts the interrupt request or the DTC/DMAC accepts the transfer request, the IRn.IR flag automatically becomes 0. It is not required to set the IRn.IR flag to 0 by software. Refer to Table 15.9, Operations When Starting the DTC/DMAC for details on clearing the IRn.IR flag by DTC/DMAC.
For the external pin interrupts of interrupt vector number 64 to 79 and interrupt sources of interrupt vector number 88 to 95, the timing when the IRn.IR flag becomes 1 after an interrupt signal occurs is different from the other interrupts. For the external pin interrupts, the timing is delayed for internal delay plus two cycles of PCLKB after a signal is input to the IRQ pin (i = 0 to 15). For interrupts of interrupt vector number 88 to 95, the timing is delayed for two cycles of PCLKB.



Note 1. A CPU interrupt request, DTC transfer request, or DMA transfer request is generated. Refer to section 15.7.3.1, Interrupt Request Destination Setting Procedure.
Note 2. For a CPU interrupt request, the IRn.IR flag becomes 0 when the CPU accepts an interrupt request. For the timing for a DTC transfer request or DMA transfer request, refer to Table 15.9, Operations When Starting the DTC/DMAC.

Figure 15.5 IRn.IR Flag Operation for Interrupts Detected by Edge Detection

(1) Detecting Consecutive Interrupt Request Signals

When interrupt request signals occur every cycle, the latter interrupt signal cannot be detected. To accept consecutive interrupt request signals, the interval of at least two cycles of the system clock or peripheral module clock, whichever is lower, is required between interrupt signals. Figure 15.6 shows the interval for accepting consecutive interrupt request signals.

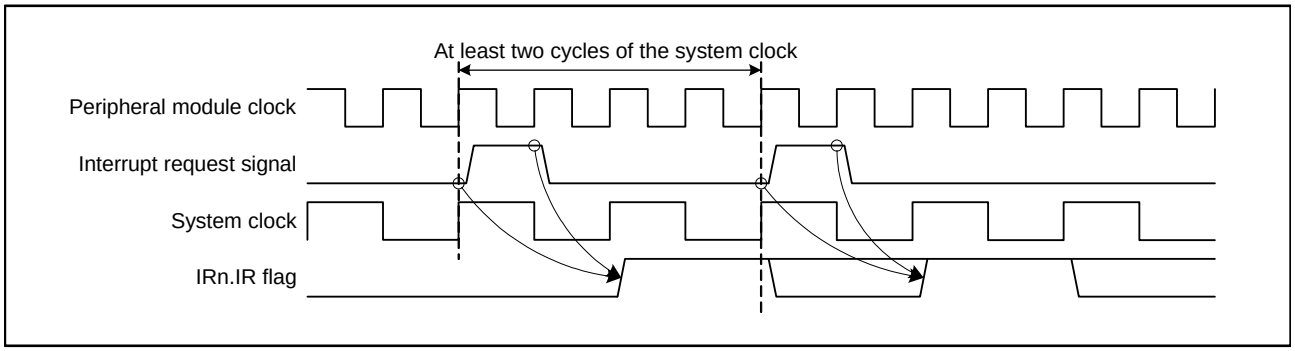


Figure 15.6 Accepting Consecutive Interrupt Signals (when the system clock frequency is lower than the peripheral clock frequency)

When an interrupt request is generated again while the IRn.IR flag is 1, the interrupt request is ignored (n = 023 to 255). However, for transmit interrupt requests, receive interrupt requests, and buffer access interrupt requests of the SCI, RSCI, RIIC, RIICHS, RSPI, RSPIA, QSPIX, SSIE, and SDHI, when an interrupt request occurs while the IRn.IR flag is 1, the interrupt request is retained in the module. After the IRn.IR flag becomes 0, the IRn.IR flag is set to 1 again by the retained request. Refer to the descriptions for interrupts in each chapter of peripheral modules for details.
Figure 15.7 shows the timing when the IRn.IR flag is set again.

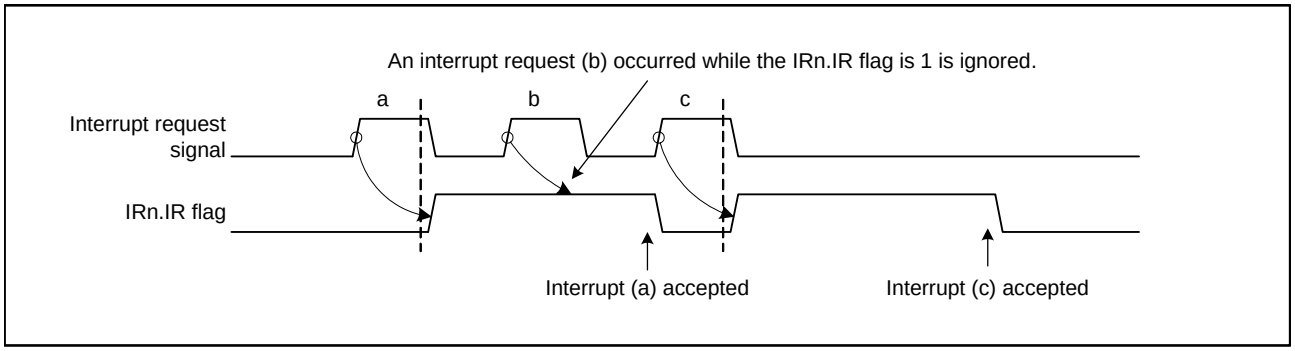


Figure 15.7 Timing to Set the IRn.IR Flag Again

(2) Relation between the IRn.IR flag and Interrupt Request Enable Bits

After the IRn.IR flag becomes 1, the IRn.IR flag does not become 0 even when an interrupt request enable bit in the corresponding peripheral module is set to 0.

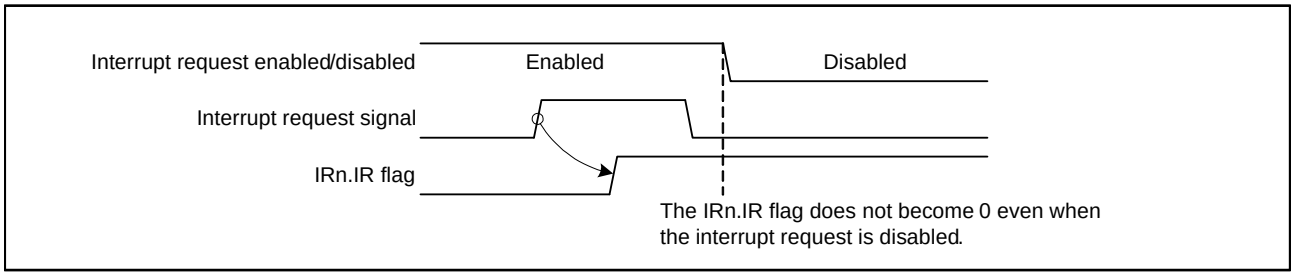


Figure 15.8 Relation Between Disabling the Interrupt Request and the IRn.IR Flag

15.5.2 Level Detection

Figure 15.9 shows operation of the interrupt request signal and the IRn.IR flag for level detection (n = 016 to 124). The IRn.IR flag is 1 while the interrupt request signal is 1. To set the IRn.IR flag to 0, set the corresponding interrupt request signal of the peripheral module to 0. Set the corresponding interrupt status flag of the peripheral module to 0 and wait for the time until the value is reflected in the IRn.IR flag before exiting the interrupt handler. Refer to (2) Notes on writing to I/O registers in section 5, I/O Registers for details on waiting for the reflection.

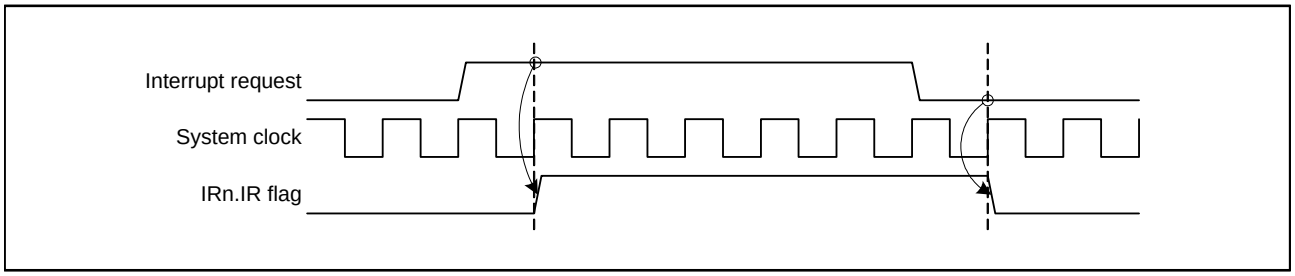


Figure 15.9 IRn.IR Flag Operation for Level Detection

Figure 15.10 shows an example of the procedure to handle interrupts for level detection.

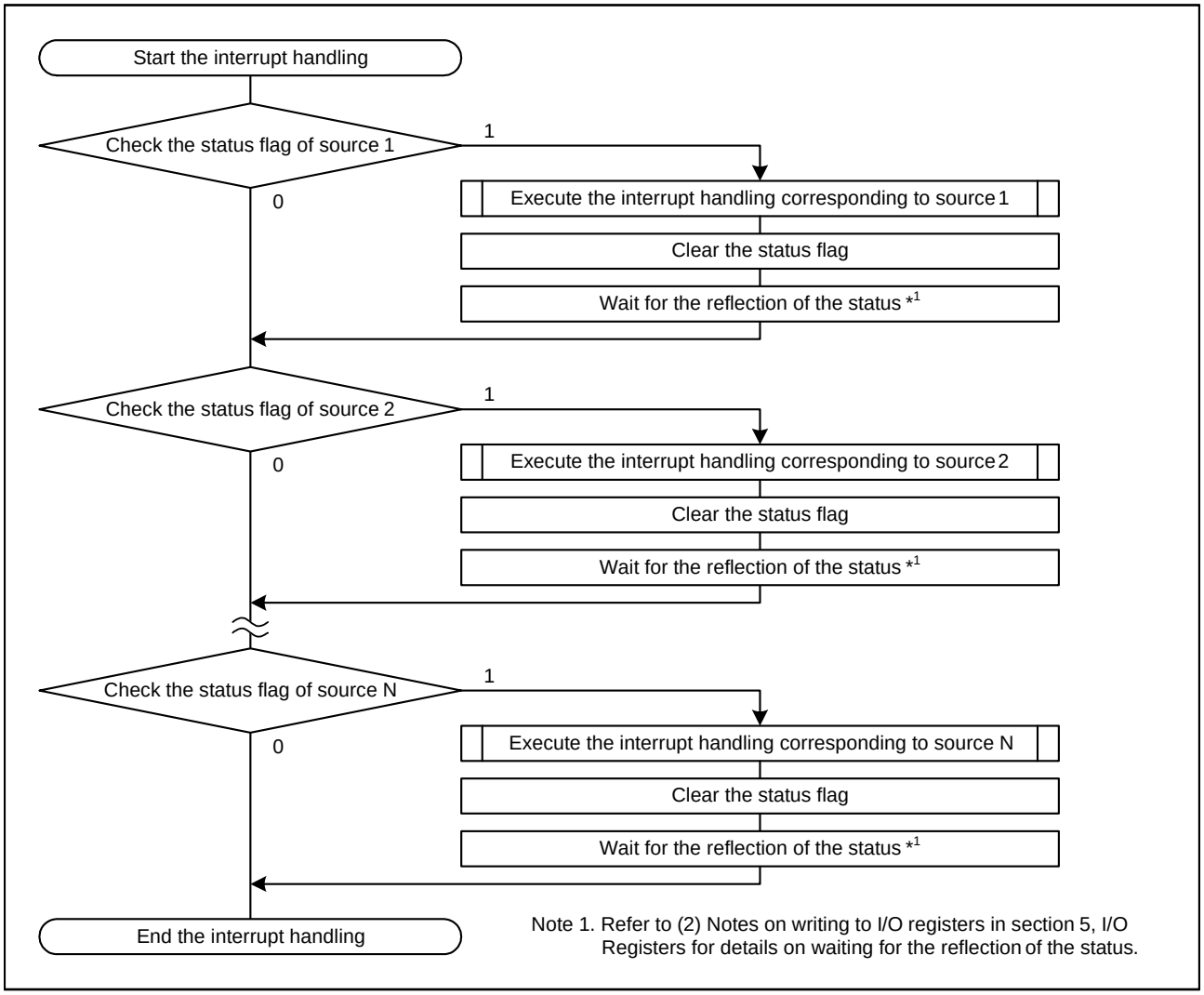


Figure 15.10 Example of Level Detection Interrupt Handling Procedure (N indicates the number of status flags)

15.5.3 Group Interrupts Using Edge Detection

Groups IE0 and BE0 of group interrupts include interrupt sources that are detected by edge detection.
While the IR017.IR flag corresponding to the GROUPIE0 interrupt and the IR106.IR flag corresponding to the GROUPBE0 interrupt become 1 under the same conditions as edge detection, the IR017.IR and IR106.IR flags become 0 under the same conditions as level detection.
When the rising edge of an interrupt request signal is detected while the corresponding GENIE0/GENBE0.ENj bit is 1, both the GRPIE0/GRPBE0.ISj flag and IR017/IR106.IR flag become 1 (j = 0 to 31). Then, GRPIE0/GRPBE0.ISj flag and IR017/IR106.IR flag do not become 0 by disabling the interrupt request of the peripheral module or setting the GENIE0/GENBE0.ENj bit to 0.
When the GCRIE0/GCRBE0.CLRj bit is set to 1, the GRPIE0/GRPBE0.ISj flag becomes 0, and consequently the IR017/IR106.IR flag becomes 0.
Figure 15.11 and Figure 15.12 show operation examples of group interrupts using edge detection. Figure 15.13 shows an example of operation when interrupt requests are generated by multiple interrupt sources in the same group.

Note: • There is no register in the group to which no interrupt source is assigned.

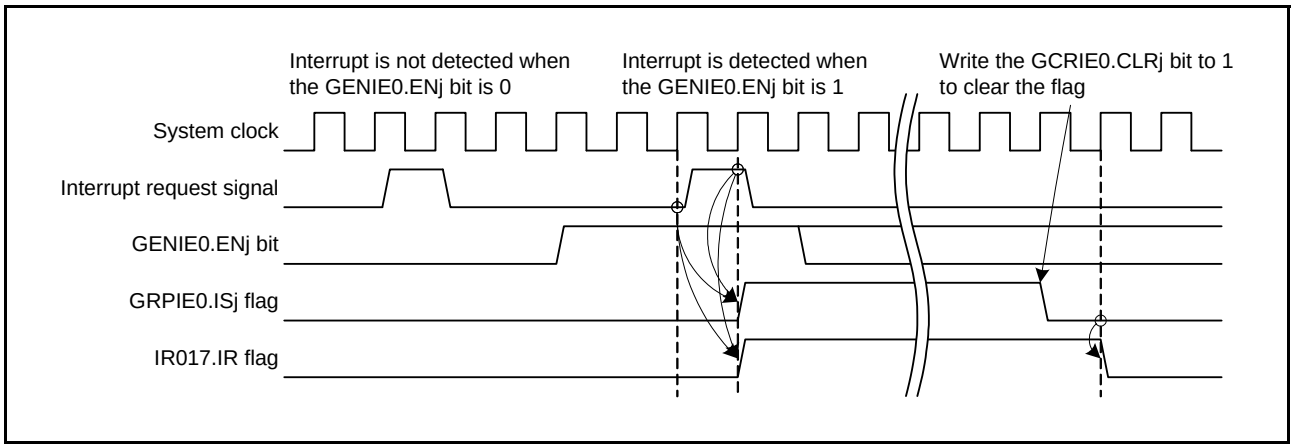


Figure 15.11 Example of Interrupt Request for Group Interrupt Using Edge Detection (Group IE0)

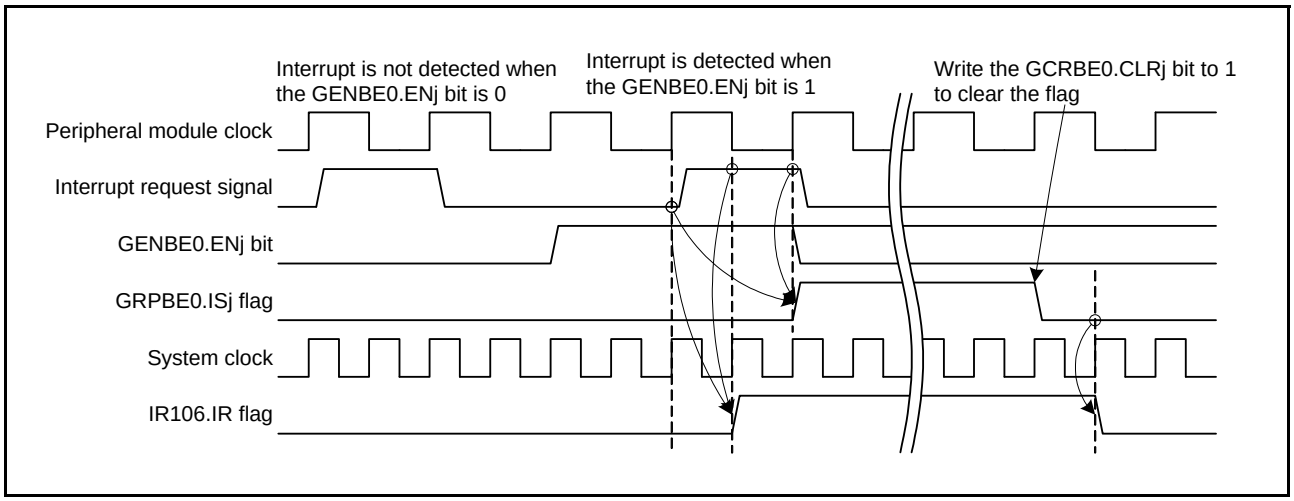


Figure 15.12 Example of Interrupt Request for Group Interrupt Using Edge Detection (Group BE0)

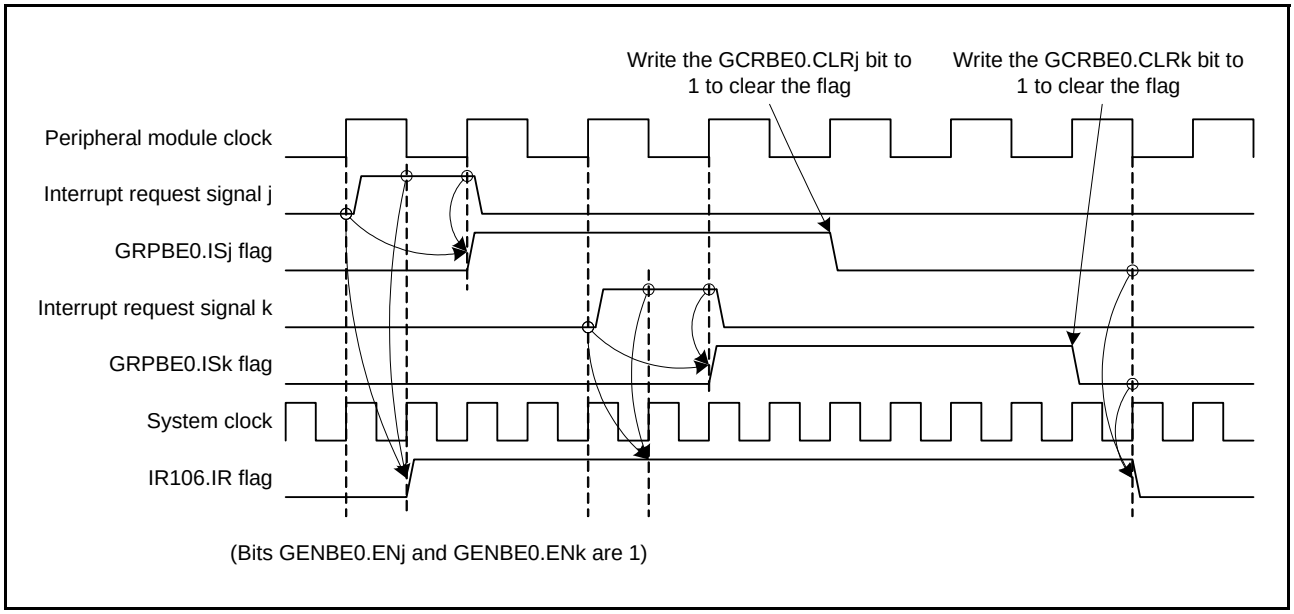


Figure 15.13 Example of Operation When Multiple Edge Detection Interrupt Requests Are Generated in the Same Group (Group BE0)

Figure 15.14 and Figure 15.15 show examples of the procedure to handle group interrupts using edge detection.

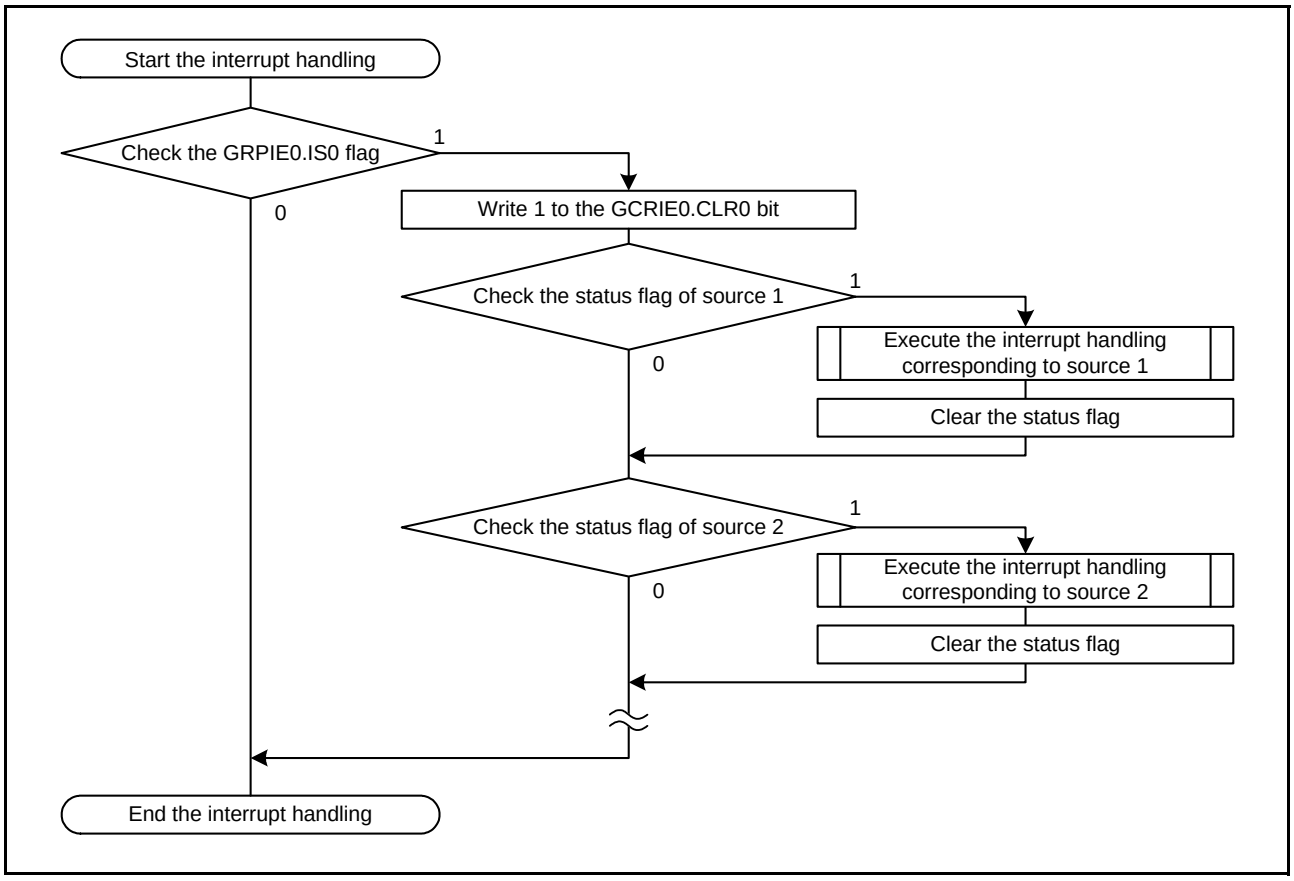


Figure 15.14 Example of Procedure to Handle Group Interrupts Using Edge Detection (Group IE0)

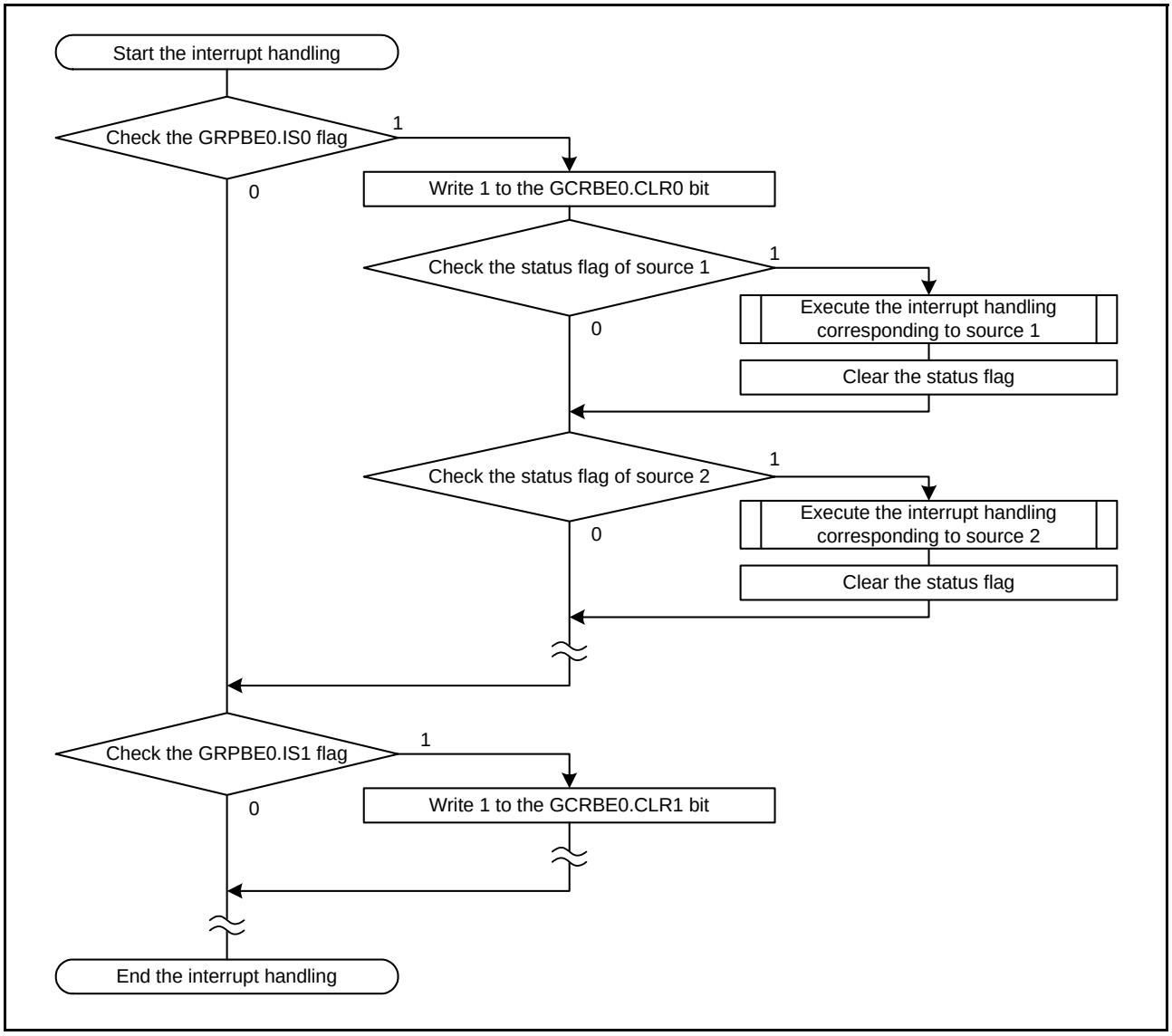


Figure 15.15 Example of Procedure to Handle Group Interrupts Using Edge Detection (Group BE0)

15.5.4 Group Interrupts Using Level Detection

Groups BL0, BL1, BL2, AL0, and AL1 of group interrupts includes interrupt sources that are detected by level detection. The IR110.IR flag corresponding to the GROUPBL0 interrupt, the IR111.IR flag corresponding to the GROUPBL1 interrupt, the IR107.IR flag corresponding to the GROUPBL2 interrupt, the IR112.IR flag corresponding to the GROUPAL0 interrupt, and the IR113.IR flag corresponding to the GROUPAL1 interrupt change under the same conditions as level detection.

When an interrupt signal becomes 1 while the corresponding GENBL0/GENBL1/GENBL2/GENAL0/GENAL1.ENj bit is 1, the GRPBL0/GRPBL1/GRPBL2/GRPAL0/GRPAL1.ISj flag and the IRn.IR flag become 1 (j = 0 to 31). Then, the GRPBL0/GRPBL1/GRPBL2/GRPAL0/GRPAL1.ISj flag and IRn.IR flag becomes 0 when the corresponding interrupt request signal becomes 0. Also, when the GENBL0/GENBL1/GENBL2/GENAL0/GENAL1.ENj bit is set to 0, the corresponding GRPBL0/GRPBL1/GRPBL2/GRPAL0/GRPAL1.ISj flag and IRn.IR flag become 0.

Figure 15.16 shows an operation example of group interrupts using edge detection. Figure 15.17 shows an example of operation when interrupt requests are generated by multiple interrupt sources in the same group.

Note: • There is no register in the group to which no interrupt source is assigned.

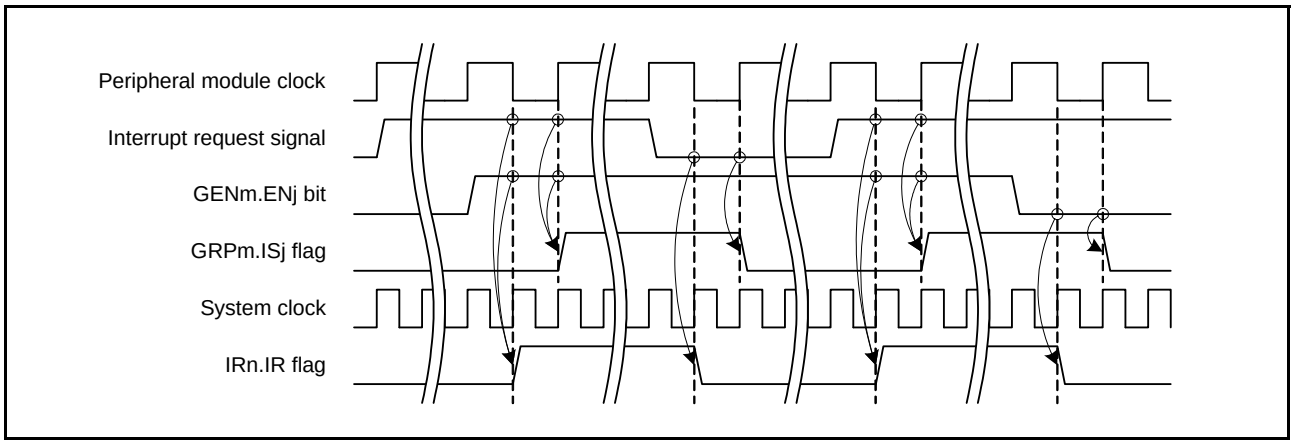


Figure 15.16 Operation Example of Group Interrupt Using Level Detection (m = BL0, BL1, BL2, AL0, AL1)

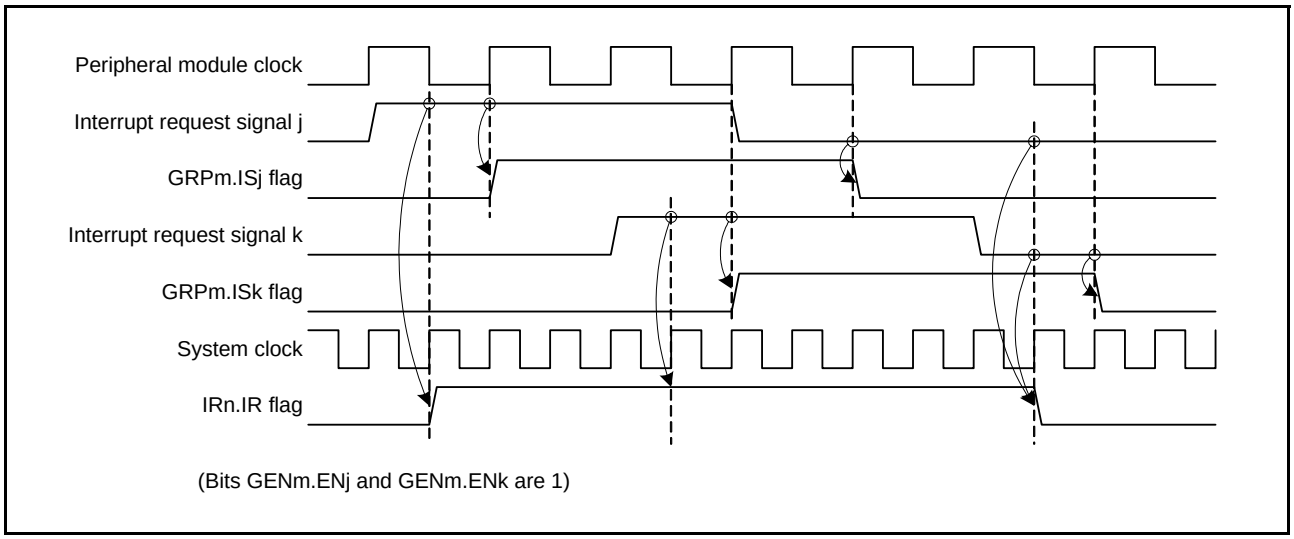


Figure 15.17 Operation Example When Multiple Interrupt Requests are Generated in the Same Group (m = BL0, BL1, BL2, AL0, AL1)

Figure 15.18 shows the procedure to handle group interrupts for level detection.

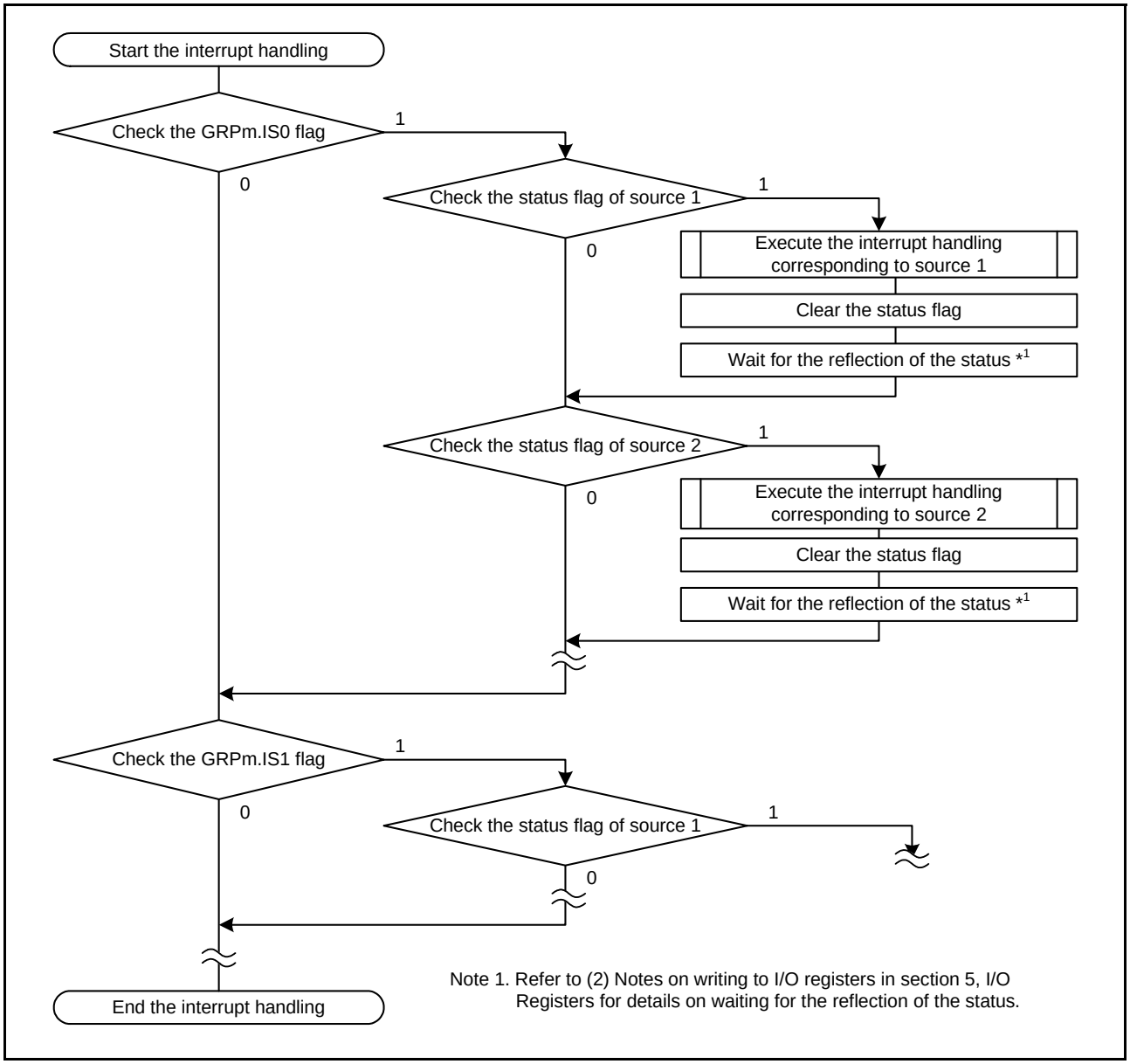


Figure 15.18 Example of Group Interrupt Handling Procedure for Level Detection (m = BL0, BL1, BL2, AL0, AL1)

15.5.5 Software Configurable Interrupts

Interrupt sources and interrupt requests for software configurable interrupts are detected by edge detection.
Figure 15.19 shows an operation example of the interrupt request and interrupt status flag for software configurable interrupts.

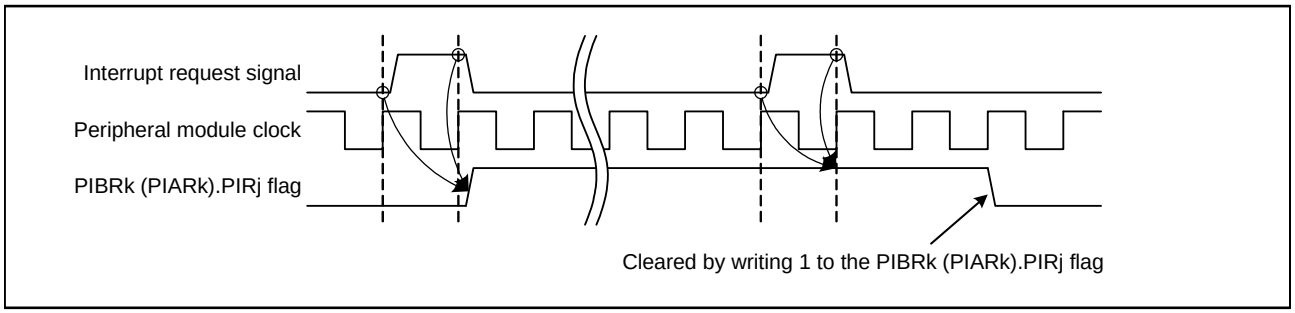


Figure 15.19 Operation Example of the Interrupt Request and Interrupt Status Flag for Software Configurable Interrupts

15.6 Determining Priority of Interrupt Requests

The ICU determines the priority for each interrupt request destination. The priority for each interrupt request destination is determined as follows.

(1) Determining Priority when the CPU is the Interrupt Request Destination

A source selected for the fast interrupt has the highest priority. Next to the fast interrupt, the priority is determined by the IPRr.IPR[3:0] bit value, and an interrupt source with a larger value has priority (r = 000 to 255). If multiple sources has the same IPRr.IPR[3:0] bit value, the priority is determined by the interrupt vector number, and the source with the smaller number has priority.

(2) Determining Priority when the DTC is the Interrupt Request Destination

The IPRr.IPR[3:0] bits have no effect (r = 000 to 255). The priority is determined by only the interrupt vector number, and an interrupt source with a smaller number has priority.

(3) Determining Priority when the DMAC is the Interrupt Request Destination

The IPRr.IPR[3:0] bits have no effect. The priority is determined by the DMAC channel number. Refer to section 18, DMA Controller (DMACAb) for details on the DMAC channel priority.

15.7 Interrupt Setting Procedure

15.7.1 Enabling Interrupt Requests

The following describes the procedure to enable interrupt requests.

(1) Set the interrupt request enable bit of the peripheral modules to enable output of the interrupt request.
(2) For group interrupts, set the corresponding ENj bit in the group interrupt request enable register to 1 to enable output of the interrupt request to the ISj flag in the group interrupt request register (j = 0 to 31).
(3) Set the corresponding IERm.IENj bit to 1 to enable output of the interrupt request to the interrupt request destination (m = 02h to 1Fh; j = 0 to 7).

After the above procedure is completed, when a peripheral interrupt occurs, the IRn.IR flag corresponding to the interrupt source becomes 1 (n = 016 to 255).
For group interrupts, the ISj flag in the group interrupt request register becomes 1, the IRn.IR flag corresponding to the group becomes 1, and an interrupt request is output to the interrupt request destination.
When the IERm.IENj bit is 0, the interrupt request corresponding to the interrupt source is not output to the interrupt request destination.

15.7.2 Disabling Interrupt Requests

The following describes the procedure to disable interrupt requests.

(1) Set the corresponding IERm.IENj bit to 0 (m = 02h to 1Fh; j = 0 to 7).
(2) For group interrupts, set the corresponding ENj bit in the group interrupt request enable register to 0 to disable output of the interrupt request to the ISj flag in the group interrupt request register (j = 0 to 31).
(3) Set the interrupt request enable bit of the peripheral modules to disable output of the interrupt request. Read the register that has been set to confirm that the value is reflected.
(4) As needed, read the IRn.IR flag or set the IR flag to 0.*1
For group interrupts, confirm that the ISj flag in the group interrupt request register is 0 or set the ISj flag to 0.

Note 1. When disabling the transmit interrupt request, receive interrupt request, or buffer access interrupt requests of the SCI, RSCI, RIIC, RIICHS, RSPI, RSPIA, QSPIX, SSIE, or SDHI, set the IRn.IR flag to 0 according to the above procedure. Refer to the description of interrupts in the corresponding section of peripheral modules for details.

15.7.3 Selecting Interrupt Request Destination

15.7.3.1 Interrupt Request Destination Setting Procedure

The destination of an interrupt request can be selected from the CPU, DTC, or DMAC for each interrupt source. Destinations that can be selected differ depending on the interrupt source. Refer to Table 15.5, Interrupt Vector Table for details on the destinations. Do not select a destination that is not indicated as "✓" in Table 15.5.
When set the external pin interrupt as the DTC or DMAC trigger, set the IRQCRi.IRQMD[1:0] bits to select edge detection (i = 0 to 15).
The following describes the procedure to select a destination of an interrupt request.

(1) Setting Interrupt Sources as the DMAC trigger

Perform the following settings while the IERm.IENj bit is 0 of the interrupt source that is selected as the DMAC trigger (m = 02h to 1Fh; j = 0 to 7).

(1) Set the interrupt vector number of the interrupt source used as the DMAC trigger in the DMRSRm register corresponding to the DMAC channel (m = DMAC channel number).*1
(2) Set the DMTMD.DCTG[1:0] bits corresponding to the DMAC channel to 01b in order to select the peripheral interrupt or external pin interrupt as the DMAC trigger.
(3) Set the DMCNT.DTE bit corresponding to the DMAC channel to 1.

After the above settings are completed, set the corresponding IERm.IENj bit to 1.
Also, set the DMAST.DMST bit to 1 before or after the above settings.
Refer to section 18.3.7, Activating the DMAC in section 18, DMA Controller (DMACAb) for the procedure to set the DMAC.

(2) Setting Interrupt Sources as the DTC trigger

Perform the following setting while the IERm.IENj bit of the interrupt source that is selected as the DTC trigger is 0.

(1) Set the DTCERn.DTCE bit corresponding to the interrupt vector number n used for the DTC trigger to 1 (n = 026 to 255).*1

After the above setting is completed, set the IERm.IENj bit to 1.
Also set the DTCST.DTCST bit to 1 before or after the above settings.
Refer to section 20.5, DTC Setting Procedure in section 20, Data Transfer Controller (DTCb) for the procedure to set the DTC.

Note 1. Do not set the same interrupt source as DTC and DMAC triggers. Also, do not set the same interrupt source as triggers of multiple DMAC channels.

(3) Setting Interrupt Sources for the CPU

When an interrupt source is not selected as the DTC or DMAC trigger, the interrupt request is output to the CPU.
Set the IERm.IENj bit to 1 while the interrupt source is not selected as a DTC or DMAC trigger.

15.7.3.2 Operations When the DTC/DMAC Selected

Table 15.9 lists operations when the DTC or DMAC is set as an interrupt request destination.

Table 15.9 Operations When Starting the DTC/DMAC

Interrupt Request Destination	DISEL *1	Number of Remaining Transfers	Operation per Request	IR Flag Clear Timing *2	Interrupt Request Destination after Transfer
DTC *3	1	≠ 0	DTC transfer → CPU interrupt	Cleared when the CPU accepts an interrupt request.	DTC
		= 0	DTC transfer → CPU interrupt	Cleared when the CPU accepts an interrupt request.	CPU (DTCERn.DTCE bit becomes 0)
	0	≠ 0	DTC transfer	Cleared when the DTC starts data transfer.	DTC
		= 0	DTC transfer → CPU interrupt *4	Cleared when the CPU accepts an interrupt request. *4	CPU (DTCERn.DTCE bit becomes 0)
DMAC	1	≠ 0	DMA transfer → CPU interrupt	Cleared when the CPU accepts an interrupt request.	DMAC
		= 0	DMA transfer → CPU interrupt	Cleared when the CPU accepts an interrupt request.	CPU (DMACm.DMCNT.DTE bit becomes 0)
	0	≠ 0	DMA transfer	Cleared when the DMAC starts data transfer.	DMAC
		= 0	DMA transfer *4	Cleared when the DMAC starts data transfer. *4	CPU (DMACm.DMCNT.DTE bit becomes 0)

Note 1. For the DTC, set the DTC.MRB.DISEL bit; For the DMAC, set the DMACm.DMCSL.DISEL bit.
Note 2. When the IRn.IR flag is 1, an interrupt request (DTC or DMA transfer request) that is generated again is ignored.
Note 3. For chain transfer, the DTC transfer continues until the chain transfer ends. After chain transfer ends, whether a CPU interrupt occurs, the IRn.IR flag clear timing, and the interrupt request destination differ depending on the DISEL bit value the number of remaining transfers. For the chain transfer, refer to Table 20.4, Chain Transfer Conditions in section 20, Data Transfer Controller (DTCb).
Note 4. When the DISEL bit is 0 and the number of remaining transfers is 0, operations in the DTC and DMAC are different.

15.7.3.3 Changing the Interrupt Request Destination

Set the IERm.IENj bit to 0 before changing the interrupt request destination (m = 02h to 1Fh; j = 0 to 7).

(1) When the current interrupt request destination is the DMAC

To change interrupt request destinations or change the DMAC trigger to another interrupt source while the DMA transfer is not completed (DMCNT.DTE bit is not cleared) after the procedure described in (1) Setting Interrupt Sources as the DMAC trigger of section 15.7.3.1, Interrupt Request Destination Setting Procedure, follow the procedure below.

(1) Set the IERm.IENj bit of both the current and new triggers to 0.
(2) Check the DMAC transfer status. If transfer is not completed, wait until the completion of transfer.
(3) Perform the procedure described in 15.7.3.1 Interrupt Request Destination Setting Procedure.

(2) When the current interrupt request destination is the DTC

To change interrupt request destinations or change the DTC transfer information while the DTC transfer is not completed (the DTCERn.DTCE bit is not cleared) after the procedure described in (2) Setting Interrupt Sources as the DTC trigger of section 15.7.3.1, Interrupt Request Destination Setting Procedure, follow the procedure below (n = 026 to 255).

(1) Set the IERm.IENj bit of both the current and new triggers to 0 (m = 02h to 1Fh; j = 0 to 7).
(2) Check the DTC transfer status. If transfer is not completed, wait until the completion of transfer.
(3) Perform the procedure described in 15.7.3.1 Interrupt Request Destination Setting Procedure.

15.7.4 Setting the External Pin Interrupt

The following describes the procedure to use the external pin interrupt.

(1) Set the IERm.IENj bit corresponding to the IRQi pin to 0 (interrupt request is disabled) (m = 02h to 1Fh; j = 0 to 7; i = 0 to 15).
(2) Set the IRQFLTE0.FLTENi or IRQFLTE1.FLTENi bit to 0 (digital filter is disabled).
(3) Set the IRQFLTC0.FCLKSELi[1:0] or IRQFLTC1.FCLKSELi[1:0] bits to select the sampling clock of the digital filter.
(4) Set the I/O port and confirm the setting.
(5) Set the IRQCRi.IRQMD[1:0] bits to select the detection method.
(6) When edge detection is selected, set the corresponding IRn.IR flag to 0 (n = 016 to 255).
(7) Set the IRQFLTE0.FLTENi or IRQFLTE1.FLTENi bit to 1 (digital filter is enabled).
(8) To select the DTC as the interrupt request destination, set the DTCERn.DTCE bit. To select the DMAC as the interrupt request destination, set the DMRSRm register. When neither the DTCERn.DTCE bit nor the DMRSRm register is set, the interrupt request is sent to the CPU (m = DMAC channel number; n = 026 to 255).
(9) Set the corresponding IERm.IENj bit to 1 (interrupt request is enabled).

15.7.5 Setting Non-Maskable Interrupts

After reset, non-maskable interrupts are disabled. To use non-maskable interrupts, follow the procedure below.

(1) Set the stack pointer (SP).
(2) When using the NMI pin, set the NMIFLTE.NFLTEN bit to 0 (digital filter is disabled).
(3) When using the NMI pin, set the NMIFLTC.NFCLKSEL[1:0] bits to select the sampling clock of the digital filter.
(4) When using the NMI pin, set the NMICR.NMIMD bit to select the edge for detection.
(5) When using the NMI pin, write 1 to the NMICLR.NMICLR bit to set the NMISR.NMIST flag to 0.
(6) When using the NMI pin, set the NMIFLTE.NFLTEN bit to 1 (digital filter is enabled).
(7) To enable generation of the non-maskable interrupt, set the bit in the NMIER register corresponding to the used interrupt source to 1.

Once a bit in the NMIER register is set to 1 (enabled), the bit cannot be rewritten so it cannot be set to 0 (disabled). To disable a non-maskable interrupt that has been enabled, reset the MCU.
Refer to section 14, Exception Handling for details on the flow of non-maskable interrupt handling.
Excluding the EXNMIST flag, each flag in the NMISR register becomes 0 by writing 1 to the corresponding bit in the NMICLR register. To set the EXNMIST flag to 0, set each flag in the EXNMISR register that has become 1 to 0.
Confirm that all flags in the NMISR register are 0 before exiting the interrupt handler of non-maskable interrupts.
Non-maskable interrupts, excluding the NMI pin interrupt, can be used as a maskable interrupt. When using as a maskable interrupt, do not change the NMIER and EXNMIER register values from the value after reset. In addition, set the LVD1CR1.LVD1IRQSEL bit and LVD2CR1.LVD2IRQSEL bit to 1 when using voltage monitoring 1 interrupt and voltage monitoring 2 interrupt as a maskable interrupt.

15.7.6 Digital Filter

Noise included in signals input to pins IRQi and NMI can be reduced by enabling the digital filter (i = 0 to 15).
The digital filter samples signals input to pins using the sampling clock (PCLKB, PCLKB/8, PCLKB/32, PCLKB/64) for the digital filter, and passes the input signal only when three consecutive sampled signals are the same level.
When using the digital filter for the IRQi pin, refer to section 15.7.4, Setting the External Pin Interrupt to set the associated registers. When using the digital filter for the NMI pin, refer to section 15.7.5, Setting Non-Maskable Interrupts to set the associated registers.
When the external pin interrupt or the NMI pin interrupt is used as a source to exit software standby mode, the digital filter cannot be used. Set the IRQFLTE0.FLTENi bit, IRQFLTE1.FLTENi bit or NMIFLTE.NFLTEN bit to 0 before entering software standby mode. To enable the digital filter again, set the IRQFLTE0.FLTENi bit, IRQFLTE1.FLTENi bit, or NMIFLTE.NFLTEN bit to 1.
Figure 15.20 shows an example of digital filter operation.

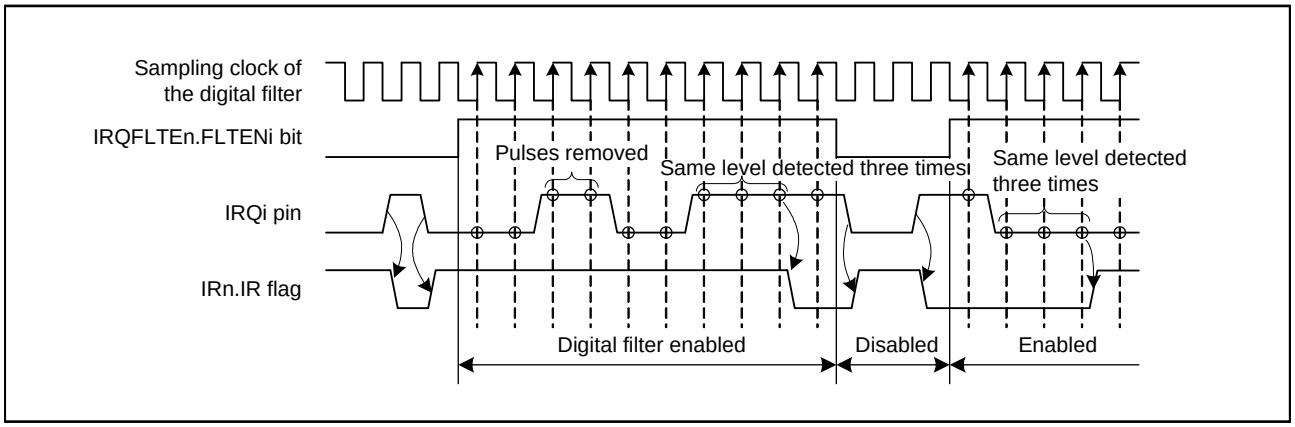


Figure 15.20 Digital Filter Operation Example (when the IRQCRi.IRQMD[1:0] bits are 00b (low level))

15.7.7 Setting Software Configurable Interrupts

The following describes the procedure to assign interrupt sources to software configurable interrupts.

(1) Set the IERm.IENj bit to 0 (m = 02h to 1Fh; j = 0 to 7). This setting is not required when the value does not change from the value after reset.
(2) For software configurable interrupt B, set the interrupt source number in registers SLIBXRn (n = 128 to 143) and SLIBRn (n = 144 to 207). Refer to Table 15.3, Interrupt Sources for Software Configurable Interrupt B for details on interrupt source numbers that are assigned to software configurable interrupt B.
(3) For software configurable interrupt A, set the interrupt source number in the SLIARn (n = 208 to 255) register. Refer to Table 15.4, Interrupt Sources for Software Configurable Interrupt A for details on interrupt source numbers that are assigned to software configurable interrupt A.
(4) When starting the EXDMAC by a software configurable interrupt, set each bit in the SELEXDR register.
(5) Set the SLIPRCR.WPRC bit to 1.
(6) Confirm that the SLIPRCR.WPRC bit is 1.
(7) Select the interrupt request destination from the CPU, DTC, or DMAC. Refer to section 15.7.3.1, Interrupt Request Destination Setting Procedure for details on the setting procedure.
(8) Write 0 to the IRn.IR flag only when edge detection is selected (n = 128 to 255).
(9) Set the IERm.IENj bit to 1.

15.7.7.1 Polling for Software Configurable Interrupts

When polling an interrupt request by reading the PIBRk.PIRj (k = 0h to Ch) or PIARk.PIRj (k = 0h to 5h, Bh), follow the procedure below (j = 0 to 7).

(1) Set the peripheral interrupt used.
(2) Clear the PIBRk.PIRj or PIARk.PIRj flag for polling by writing 1 to the flag.*1
(3) Enable output of the peripheral interrupt request.
(4) As needed, read the PIBRk.PIRj or PIARk.PIRj flag to check the value.
(5) When clearing the PIBRk.PIRj or PIARk.PIRj flag, write 1 to the targeted flag.*1
(6) As needed, repeat step (4) and (5).

Note 1. Do not use bit manipulation instructions. Multiple status flags may be cleared if a bit manipulation instruction is used. To clear a flag, write the PIBRk or PIARk register in 8-bit units as follows: set the flag that is to be cleared to 1 and set the other flags to 0.

15.8 Multiple Interrupt

To enable another interrupt while processing an interrupt (multiple interrupt), set the PSW.I bit to 1 (interrupt enabled) in the interrupt handler of an accepted interrupt.
The PSW.IPL[3:0] bits in the interrupt handler are the same value as the priority level of the accepted interrupt request. In this case, when an interrupt request with the higher priority level than the PSW.IPL[3:0] bit value is generated, the interrupt request is accepted.
The PSW.I bit can be rewritten only in supervisor mode. Since the PSW.PM bit becomes 0 (supervisor mode is selected) when an interrupt is accepted, the PSW.I bit can be rewritten in the interrupt handler.

15.9 Fast interrupt

The fast interrupt is an interrupt that the CPU can respond to fast. Only one interrupt source can be assigned to the fast interrupt.
The priority level of the fast interrupt is 15 (highest) regardless of the IPRr.IPR[3:0] bit setting (r = 000 to 255). Also, the fast interrupt has higher priority than the other interrupt sources of which the priority level is 15. Note that the fast interrupt cannot be accepted when the PSW.IPL[3:0] bits are 1111b (priority level 15).
To assign an interrupt source to the fast interrupt, set the FIR.FVCT[7:0] bits to select the vector number of the interrupt source, and set the FIR.FIEN bit to 1 (fast interrupt is enabled).
The fast interrupt is enabled only when the CPU is selected as the interrupt request destination. When the DTC or DMAC is selected as the destination, the fast interrupt is disabled.
Refer to section 2, CPU and section 14, Exception Handling for details on the fast interrupt.

15.10 Exiting Low Power Consumption State

Interrupts can be used for exiting sleep mode, all-module clock stop mode, and software standby mode.
Refer to section 11, Low Power Consumption for details. This section describes the procedure to set an interrupt source for exiting each low power consumption mode.
Refer to section 11.6.4, Deep Software Standby Mode for details on exiting deep software standby mode.

15.10.1 Exiting Sleep Mode

Non-maskable interrupts and all interrupt sources can be used for exiting sleep mode. The following conditions must be satisfied.

(1) Non-maskable interrupts

- Generation of the interrupt that is used for exiting is enabled in the NMIER register.

(2) Interrupts

- The CPU is selected as the interrupt request destination.
- The interrupt request is enabled by the IERm.IENj bit (m = 02h to 1Fh; j = 0 to 7).
- The priority level is higher than the PSW.IPL[3:0] bit value in the CPU.
- For group interrupts, the interrupt request is enabled by the corresponding ENj bit in the group interrupt request enable register (GENIE0, GENBE0, GENBL0, GENBL1, GENAL0, GENAL1) (j = 0 to 31).

15.10.2 Exiting All-Module Clock Stop Mode

Non-maskable interrupts and interrupt sources that have a “✓” in the Exit from ACS column in Table 15.5, Interrupt Vector Table can be used for exiting all-module clock stop mode. The conditions below must be satisfied.

(1) Non-maskable interrupts

- Generation of the interrupt that is used for exiting is enabled in the NMIER register.

(2) Interrupts

- The interrupt source can be used for exiting all-module clock stop mode.
- The CPU is selected as the interrupt request destination.
- The interrupt request is enabled by the IERm.IENj bit (m = 02h to 1Fh; j = 0 to 7).
- The priority level is higher than the PSW.IPL[3:0] bit value in the CPU.

15.10.3 Exiting Software Standby Mode

Non-maskable interrupts (excluding oscillation stop detection interrupt) and interrupt sources that have a "✓" in the Exit from SSBY column in Table 15.5, Interrupt Vector Table can be used for exiting software standby mode. The conditions below must be satisfied.

(1) Non-maskable interrupts

- Generation of the interrupt that is used for exiting is enabled in the NMIER register.
- When using the NMI pin interrupt, the digital filter is disabled.

(2) Interrupts

- The interrupt source can be used for exiting software standby mode.
- The CPU is selected as the interrupt request destination.
- The interrupt request is enabled by the IERm.IENj bit (m = 02h to 1Fh; j = 0 to 7).
- The priority level is higher than the PSW.IPL[3:0] bit value in the CPU.
  When using the fast interrupt, set not only the FIR register but also the corresponding IPRr.IPR[3:0] bits (r = 000 to 255). Set the IPRr.IPR[3:0] bits to a higher level than the PSW.IPL[3:0] bit value in the CPU.
- When using the external pin interrupt, the digital filter of the IRQi pin used is disabled.

Refer to section 15.7.6, Digital Filter for details on the procedure to set the digital filter.

15.11 Usage Notes

15.11.1 Notes on the WAIT instruction When Using the Non-Maskable Interrupt

Confirm that all status flags in the NMISR register are 0 before executing the WAIT instruction.

15.11.2 Software Configurable Interrupts in All-Module Clock Stop Mode

When using an interrupt source assigned to a software configurable interrupt for exiting all-module clock stop mode, assign the interrupt source to software configurable interrupt B (INTB146 to INTB157) of interrupt vector numbers 146 to 157.

15.11.3 Interrupt Requests in Software Standby Mode

When an interrupt request occurs in software standby mode but the interrupt source is not set as a source for exiting software standby mode, the request is held in the ICU. The request is handled after exiting by another interrupt source. Note that the interrupt request for the external pin interrupt is not held.

16. Buses

16.1 Overview

Table 16.1 lists the bus specifications, Figure 16.1 shows the bus configuration, and Table 16.2 lists the addresses assigned for each bus.

Table 16.1 Bus Specifications

Bus Type		Description
CPU bus	Instruction bus	• Connected to the CPU (for instructions) • Connected to on-chip memory (RAM, code flash memory) • Operates in synchronization with the system clock (ICLK)
	Operand bus	• Connected to the CPU (for operands) • Connected to on-chip memory (RAM, code flash memory) • Operates in synchronization with the system clock (ICLK)
Memory bus	Memory bus 1	• Connected to RAM
	Memory bus 2	• Connected to code flash memory
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 and DTC • Connected to on-chip memory (RAM, code flash memory) • Operates in synchronization with the system clock (ICLK)
Internal peripheral bus	Internal peripheral bus 1	• Connected to peripheral modules (DTC, DMAC, EXDMAC, interrupt controller, and bus error monitoring section) • Operates in synchronization with the system clock (ICLK) (EXDMAC 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, DOC, CTSU, REMC, and standby RAM) • Operates in synchronization with the peripheral-module clock (PCLKB)
	Internal peripheral bus 4	• Connected to peripheral modules (MTU, SCIm, and RSPI) • Operates in synchronization with the peripheral-module clock (PCLKA)
	Internal peripheral bus 5	• Connected to peripheral modules (RSCI, RSPIA, RIICHS) • Operates in synchronization with the peripheral-module clock (PCLKA)
	Internal peripheral bus 6	• Connected to code flash (in P/E) and data flash memory • Operates in synchronization with the FlashIF clock (FCLK)
External bus	CS area	• Connected to the external devices • Operates in synchronization with the external-bus clock (BCLK)
	SDRAM area	• Connected to the SDRAM • Operates in synchronization with the SDRAM clock (SDCLK)
Internal expansion bus	QSPI area	• Connected to external SPI devices • Operates in synchronization with the system clock (ICLK)

P/E: Programming/Erasure

BCLK (external-bus clock): 120 MHz (max.) The CSC (CS area controller) and the EXDMAC operate in synchronization with the BCLK.

SDCLK (SDRAM clock): 60 MHz (max.) The SDRAMC (SDRAM area controller) operates in synchronization with the SDCLK.

BCLK pin output: The frequency is the same as the BCLK as default. 1/2 BCLK can be supplied by setting the BCLK pin output select bit (BCKCR.BCLKDIV) in the external bus clock control register. For details, refer to section 9, Clock Generation Circuit.

Note: The BCLK and the SDCLK should be operated with the same frequency when the SDRAM is in use.

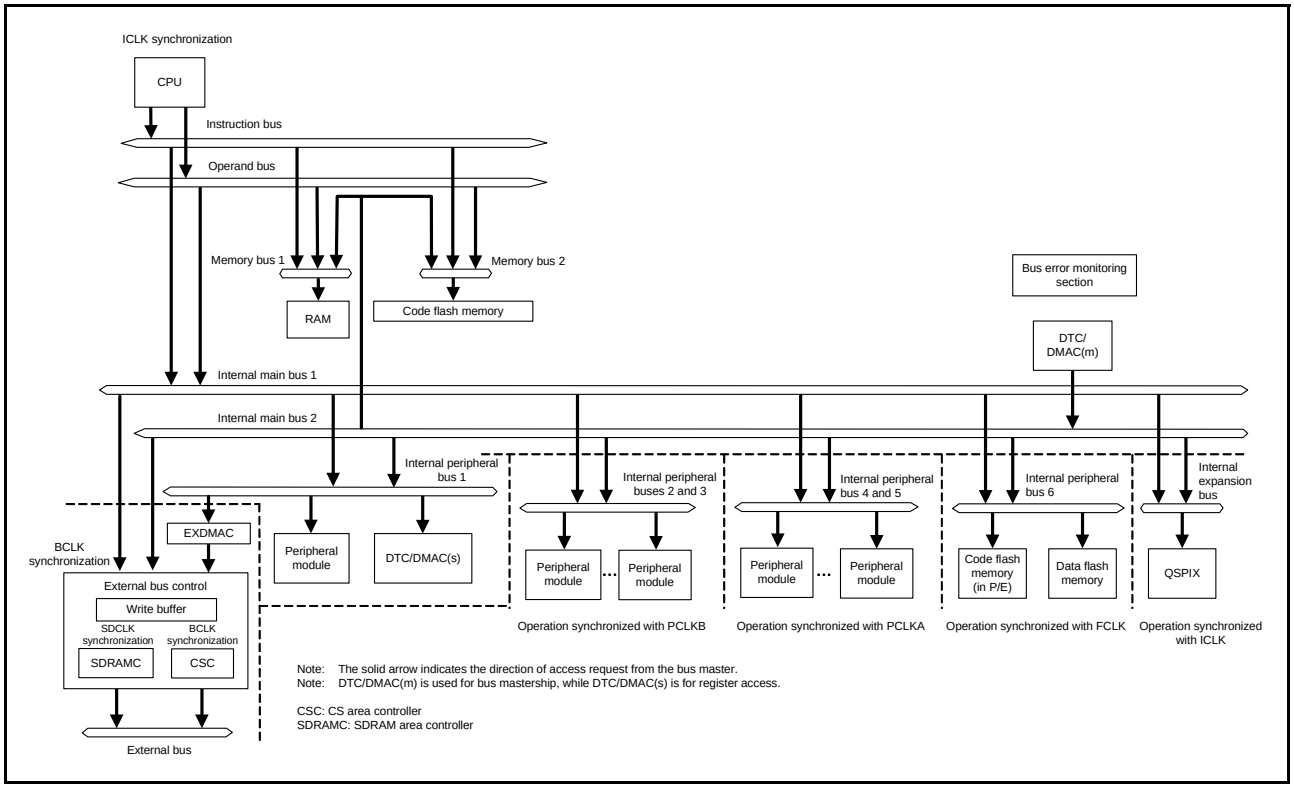


Figure 16.1 Bus Configuration

Table 16.2 Addresses Assigned for Each Bus

Address	Bus		Area	
	On-Chip ROM Enabled	On-Chip ROM Disabled	On-Chip ROM Enabled	On-Chip ROM Disabled
0000 0000h to 0007 FFFFh	Memory bus 1		RAM	
0008 0000h to 0008 7FFFh	Internal peripheral bus 1		Peripheral I/O registers	
0008 8000h to 0009 FFFFh	Internal peripheral bus 2			
000A 0000h to 000B FFFFh	Internal peripheral bus 3			
000C 0000h to 000D FFFFh	Internal peripheral bus 4			
000E 0000h to 000F FFFFh	Internal peripheral bus 5			
0010 0000h to 007F FFFFh	Internal peripheral bus 6	Reserved area	Code flash memory (for programming/erasure), data flash memory	Reserved area
0080 0000h to 00FF FFFFh	Reserved area		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 6FFF FFFFh	Reserved area		Reserved area	
7000 0000h to 77FF FFFFh	Internal expansion bus		QSPI area	
7800 0000h to 7FFF FFFFh	Reserved area		Reserved area	
8000 0000h to FEFF FFFFh	Memory bus 2	Reserved area	Code flash memory	Reserved area
FF00 0000h to FFFF FFFFh		External bus		External address space (CS0)

16.2 Description of Buses

16.2.1 CPU Buses

The CPU buses consist of the instruction and operand buses, which are connected to internal main bus 1. As the names suggest, the instruction bus is used to fetch instructions for the CPU, while the operand bus is used for operand access. The instruction bus is 64 bits while the operand bus is 64 bits.
Connection of the instruction and operand buses to RAM and code flash memory provides the CPU with direct access to these areas, i.e. access is not via internal main bus 1. However, only reading is possible in direct access to code flash memory by the CPU; programming and erasure are handled via an internal peripheral bus.
Bus requests for instruction fetching and operand access are arbitrated through internal main bus 1. The order of priority is operand access then instruction fetching.
If instruction fetching and operand access are requested for different buses (memory bus 1, memory bus 2, and internal main bus 1), the bus-access operations can proceed simultaneously. For example, parallel access to code flash memory and RAM or to code flash memory and external space is possible.

16.2.2 Memory Buses

The memory bus has memory bus 1 and memory bus 2. The RAM is connected to memory bus 1 and code flash memory is connected to memory bus 2. The memory buses are 64 bits. Requests for bus mastership from the CPU buses (instruction fetching and operand) and internal main bus 2 are arbitrated through memory buses 1 and 2.
The priority order of the buses can be set using the memory bus 1 (RAM) priority control bits (BPRA[1:0]) and memory bus 2 (code flash memory) priority control bits (BPRO[1:0]) in the bus priority control register (BUSPRI) for the corresponding memory buses. 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 and DMAC (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 and DMAC are arbitrated by internal main bus 2. The order of priority is DMAC and then DTC, as indicated in Table 16.3.
Between the DTC and DMAC, only the one that accepted the transfer request issues the bus mastership request. The priority order of transfer requests between the DTC and DMAC is DMAC0, DMAC1, DMAC2, DMAC3, DMAC4, DMAC5, DMAC6, DMAC7, and then DTC, regardless of the BUSPRI setting.
If the CPU and another bus master are requesting access to different buses (on-chip memory, internal peripheral buses 1 to 6, and external bus), the respective bus-access operations can proceed simultaneously.
However, when the CPU executes the XCHG instruction, requests for bus access from masters other than the CPU are not accepted until data transfer for the XCHG instruction is completed regardless of the bus priority control register (BUSPRI) setting. Furthermore, requests for bus access from masters other than the DTC are not accepted during reading and writing-back of transfer control information for the DTC.

Table 16.3 Order of Priority for Bus Masters

Priority	Internal main bus	Bus Master
High	—	EXDMAC
↑	2	DMAC
		DTC
Low	1	CPU

Note: The above applies when the priority order of the buses is fixed.
The priority order of the internal main bus 1 and other buses (internal main bus 2 and EXDMAC) can be toggled by using the bus priority control register (BUSPRI) (round-robin method).
However, the priority order of EXDMAC to be connected only to the external bus has priority over internal main bus 2 regardless of the setting of the bus priority control register (BUSPRI) (EXDMAC > internal main bus 2).

16.2.4 Internal Peripheral Buses

Connection of peripheral modules to the internal peripheral buses is as described in Table 16.4.

Table 16.4 Connection of Peripheral Modules to the Internal Peripheral Buses

Type of Bus	Peripheral Modules
Internal peripheral bus 1	DTC, DMAC, EXDMAC, interrupt controller, and bus error monitoring section
Internal peripheral bus 2	Peripheral modules other than those connected to internal peripheral buses 1, 3, 4, and 5
Internal peripheral bus 3	USB, DOC, CTSU, REMC, and standby RAM
Internal peripheral bus 4	MTU, SCIm, and RSPI
Internal peripheral bus 5	RSCI, RSPIA, RIICHS
Internal peripheral bus 6	Code flash memory (in P/E) or data flash memory

Requests for bus mastership from the CPU (internal main bus 1) and other bus masters (internal main bus 2) are arbitrated through internal peripheral buses 1 to 6.
The priority order of two internal main buses can be set using the bus priority control register (BUSPRI). The priority order can be set with the internal peripheral bus 1 priority control bits (BUSPRI.BPIB[1:0]), internal peripheral bus 2 and 3 priority control bits (BUSPRI.BPGB[1:0]), internal peripheral bus 4 and 5 priority control bits (BUSPRI.BPHB[1:0]), and internal peripheral bus 6 priority control bits (BUSPRI.BPFB[1:0]) for the corresponding internal peripheral buses.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1. When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted (round-robin method).
The order of accepting requests may change depending on the BUSPRI setting (Refer to Figure 16.2).

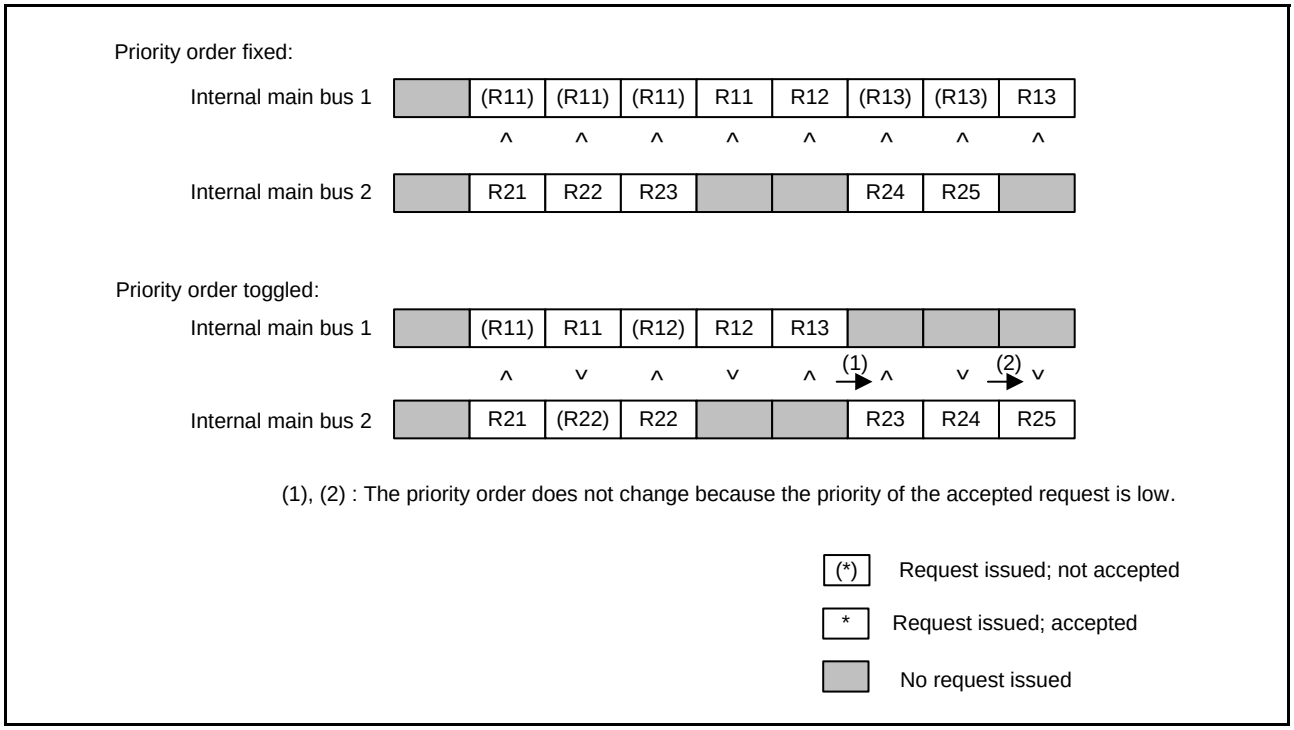


Figure 16.2 Priority Order between Internal Peripheral Bus Accesses

16.2.5 Write Buffer Function (Internal Peripheral Bus)

The internal peripheral bus has the write buffer function, which allows the next round of bus access to start, before the current write access is completed, in write access. However, if the following round of bus access is from the same bus master but to the different internal peripheral bus, it is suspended until the bus operations already in progress are completed. When access to the internal memory is scheduled after the write access to the internal peripheral bus from the CPU, the following round of bus access can be started before the current bus operation is completed and thus the order of accesses may be changed (Refer to Figure 16.3).

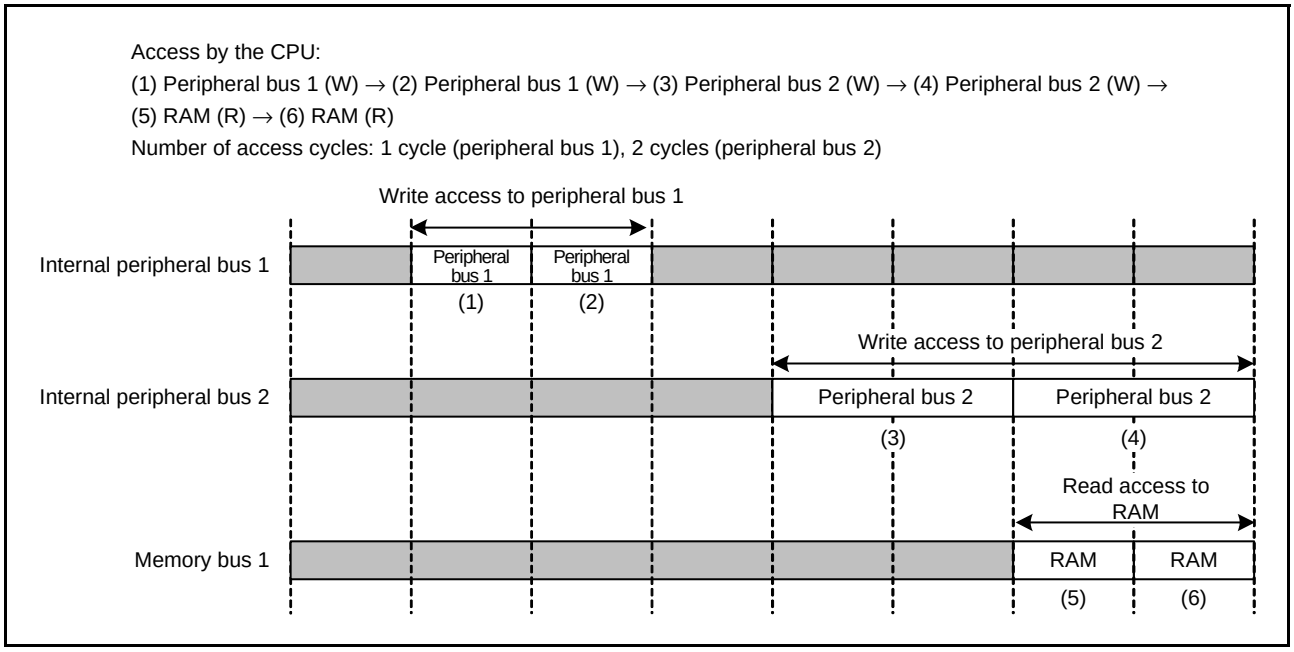


Figure 16.3 Write Buffer Function

16.2.6 Internal Expansion Bus

Table 16.5 lists peripheral module connected to the internal expansion bus.

Table 16.5 Peripheral Module Connected to the Internal Expansion Bus

Type of Bus	Description	Endian
Internal expansion bus	• The QSPIX interface is connected to the internal expansion bus. • The internal expansion bus operates in synchronization with the ICLK.	Little endian

The internal expansion bus arbitrates requests for bus mastership in access through internal main bus 1 for the CPU and internal main bus 2 for bus masters other than the CPU.
The internal expansion bus is 32 bits wide. Operation of the internal expansion bus is little endian.
The method of controlling priority for the two buses can be set by the bus priority control register (BUSPRI). The priority order can be set by the internal expansion bus priority control bits (BUSPRI.BPXB[1:0]). If the fixed order of priority is to be used, access through internal main bus 2 takes priority over access through internal main bus1. If the priority order is to be toggled, the priority of the bus from which a bus request was most recently accepted becomes lower.
Note that changing the setting of the bus priority control register may change the order of the acceptance of requests. Refer to Figure 16.4 for examples.

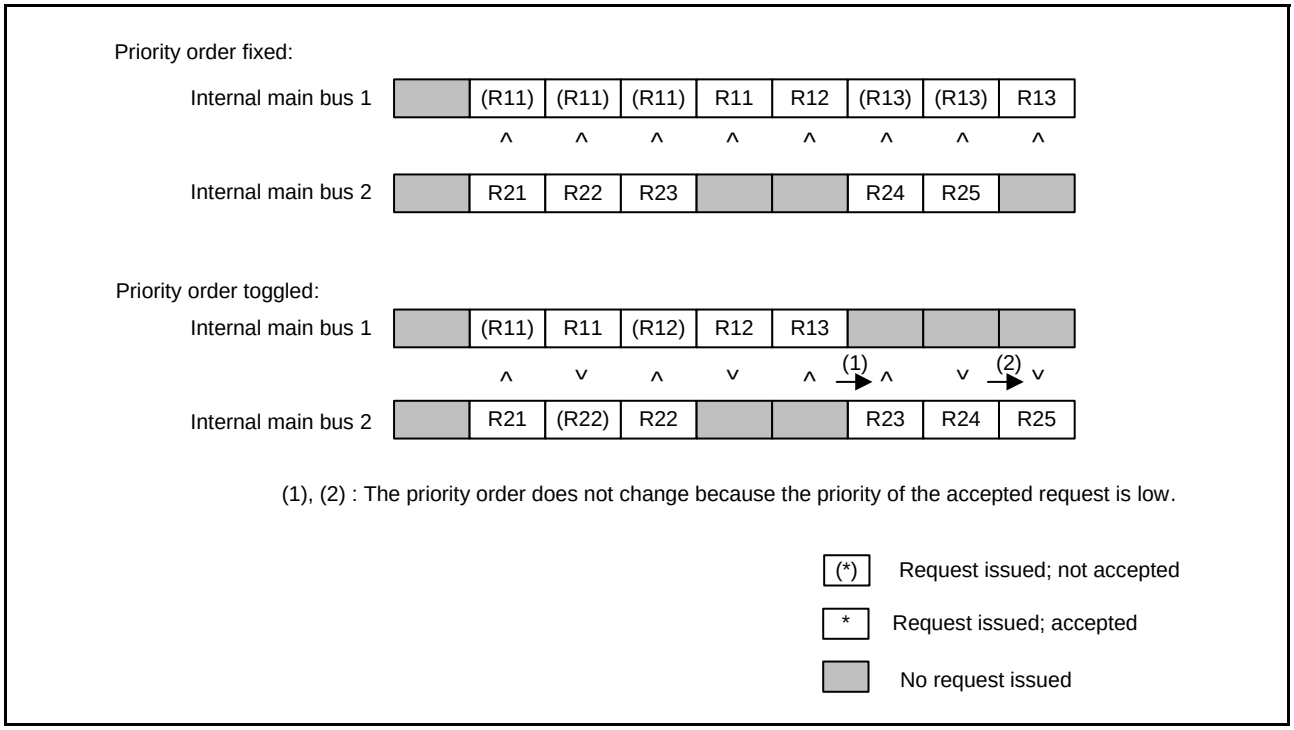


Figure 16.4 Priority Order of Bus Accesses

The internal expansion bus has a write buffer so a next data for writing can be accepted without waiting for the completion of the current writing. In general, however, if writing from the same internal main bus is to a different bus, note that writing must be held pending until the current writing is completed, but if writing to the on-chip memory is to proceed after writing to the internal expansion bus by the CPU, the next data for writing can be accepted, and that may change the order of access. Refer to Figure 16.5 for an example.

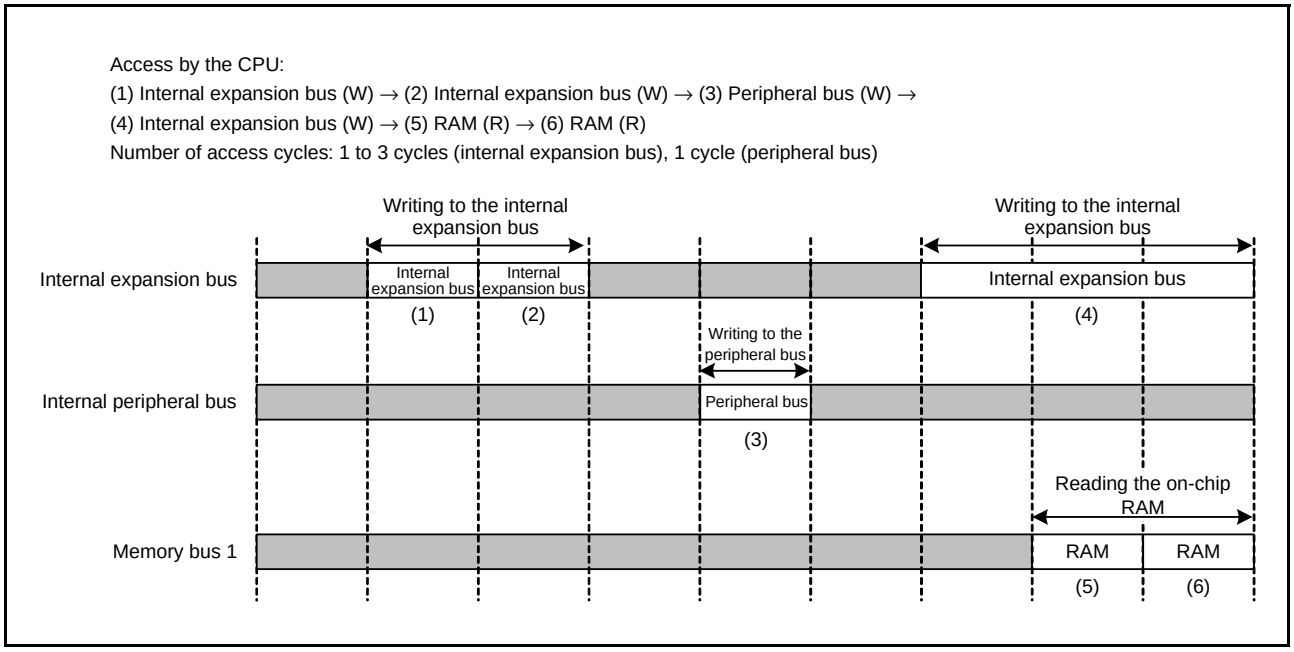


Figure 16.5 Write Buffer Function

16.2.7 External Bus

Table 16.6 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 16.6).

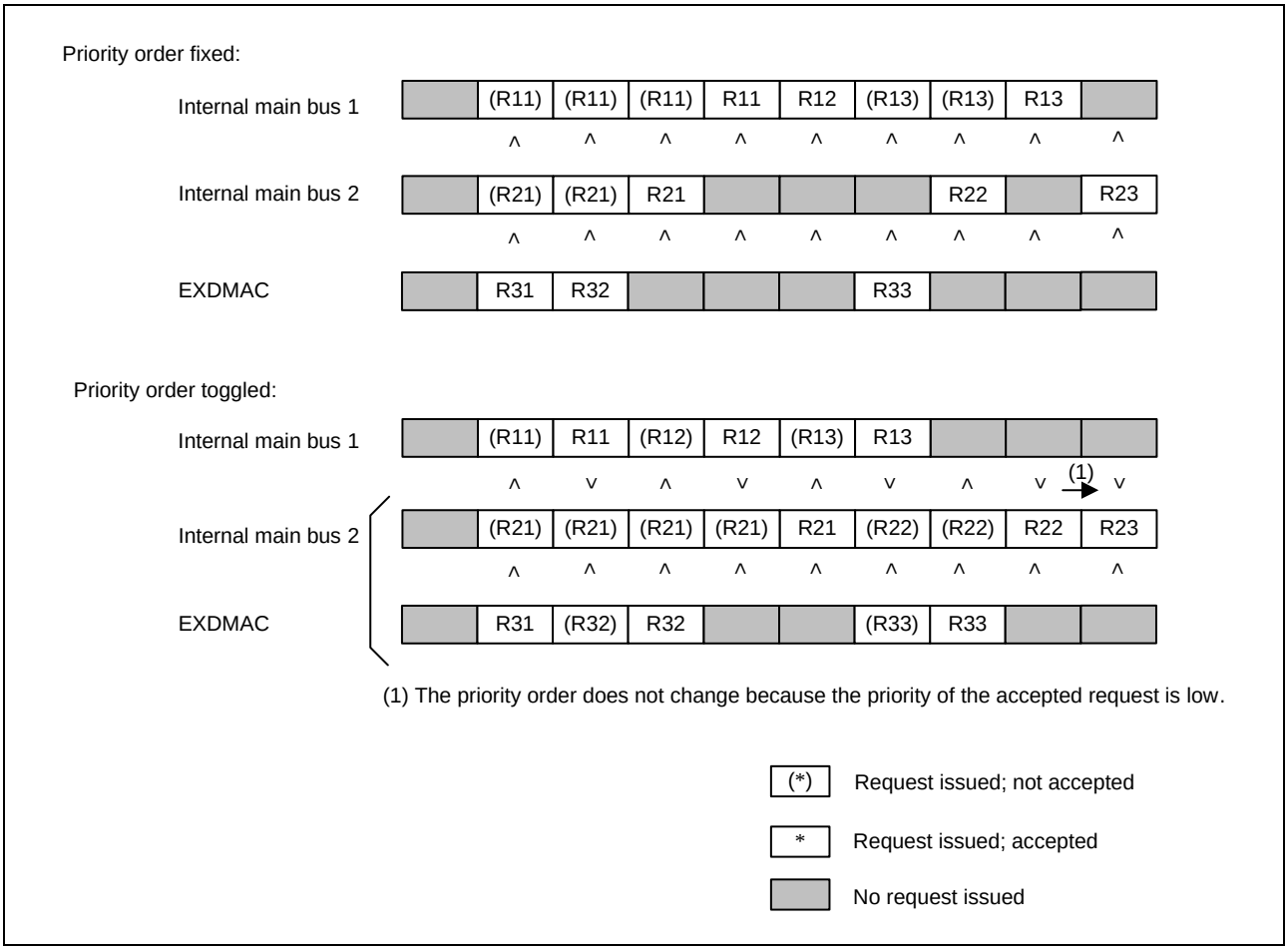


Figure 16.6 Priority Order of Internal Peripheral Bus Accesses

Table 16.6 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 or 16-bit bus space is selectable. Address/data multiplexed bus: An 8 or 16-bit bus space is selectable. • An endian mode can be specified for each area.
CS area controller	• Recovery cycles can be inserted. Read recovery: Up to 15 cycles Write recovery: Up to 15 cycles • Cycle wait function: Wait for up to 31 cycles (page access: up to 7 cycles) • Wait control can be used to set up the following. Timing of assertion and negation for chip-select signals (CS0# to CS7#) The timing of assertion of the read signal (RD#) and write signals (WR0#/WR#, and WR1#) 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 16.7 lists the input/output pins of the external bus.

Table 16.7 Pin Configuration of the External Bus (1/2)

Pin Name	I/O	Description
A23 to A0*1	Output	Address output pins
D15 to D0	I/O	Data input/output pins 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.
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

Table 16.7 Pin Configuration of the External Bus (2/2)

Pin Name	I/O	Description
WR0#/WR#	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.
ALE	Output	Address latch signal when address/data multiplexed bus is selected.
WAIT#	Input	A wait request signal when accessing the external address space (CS0 to CS7) (Low: Wait request)
SDCLK	Output	SDRAM clock
CKE	Output	SDRAM clock enable signal
SDCS#	Output	SDRAM chip select signal
RAS#	Output	SDRAM low address strobe signal
CAS#	Output	SDRAM column address strobe signal
WE#	Output	SDRAM write enable signal
DQM0	Output	SDRAM I/O data mask enable signal for D7 to D0
DQM1	Output	SDRAM I/O data mask enable signal for D15 to D8

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, refer to section 22, I/O Ports.

16.2.8 Parallel Operation

Parallel operation is possible when different bus-master modules are requesting access to different slave modules. For example, if the CPU is fetching an instruction from code flash memory and an operand from RAM, the DMAC is able to handle transfer between a peripheral bus and the external bus at the same time.
An example of parallel operations is shown in Figure 16.7. In this example, the CPU is able to employ the instruction and operand buses for simultaneous access to code flash memory and RAM, respectively. Furthermore, the DMAC simultaneously employs internal main bus 2 for access to a peripheral bus or the external bus during access to RAM and code flash memory by the CPU.

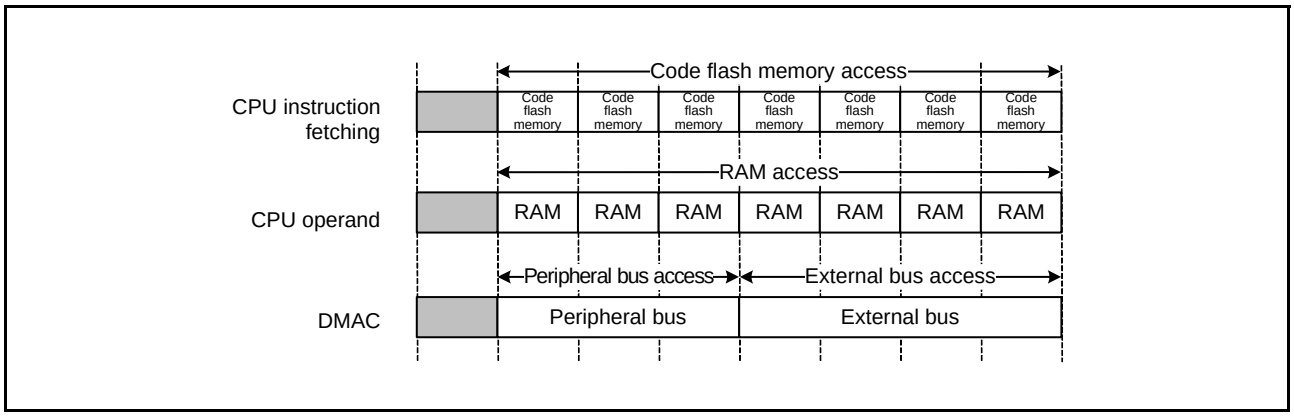


Figure 16.7 Example of Parallel Operations

16.2.9 Bus Settings

(1) Set the mode of the external bus in the CSn mode register (CSnMOD), CSn wait-control register 1 (CSnWCR1), CSn wait-control register 2 (CSnWCR2), CSn control register (CSnCR), CSn recovery-cycle setting register (CSnREC), CS recovery cycle insertion enable register (CSRECEN), bus error monitoring enable register (BEREN), and bus priority control register (BUSPRI).
(2) Make settings for pins in the CS output enable register (PFCSE), CS output pin select register 0 (PFCSS0), CS output pin select register 1 (PFCSS1), address output enable register 0 (PFAOE0), address output enable register 1 (PFAOE1), external-bus control register 0 (PFBCR0), and external-bus control register 1 (PFBCR1).
(3) Set up pins to be used as input port pins.
(4) Set the external-bus enable bit (EXBE) in the system control register 0 (SYSCR0) to 1 (enabling the external bus).

16.2.10 Restrictions

(1) Prohibition of Access that Spans Areas of Address Space

Single access that spans two areas of the address space is prohibited, and operation of such an access is not guaranteed. Setting must be made so that two areas are not accessed at the same time by a single word or longword access.
Single access that spans two areas of the address space is also prohibited in the EXDMAC block transfer in single address mode or cluster transfer, and operation of such an access is not guaranteed. Setting must be made so that two areas are not accessed at the same time by a single access in the EXDMAC block transfer in single address mode or cluster transfer.

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

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

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

(3) Restriction on Endian

When the endian setting for each area is different from that for the chip, no instruction code can be allocated in the area. The instruction code should be allocated to the external space whose endian setting is the same as that for the chip.
For the CPU to handle 64-bit operand access, the area for access must have the same endian setting as that of the chip. 64-bit operand access is not possible in the case of areas with a different endian setting from that of the chip.
The internal expansion bus only supports little endian operation. If instruction codes are to be placed in the internal expansion bus area while big endian is selected for the chip, place the instruction codes in the bus area as binary values for use in big endian and copy them for execution to another area where big endian is selected.

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
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: Setting prohibited 1 0: An 8-bit bus space is selected 1 1: Setting prohibited	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	EMODE	Endian Mode	0: Endian of area n is the same as the endian of operating mode. 1: Endian of area n is not the endian of operating mode. (n = 0 to 7)	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	MPXEN	Address/Data Multiplexed I/O Interface Select	0: Separate bus interface is selected for area n. 1: Address/data multiplexed I/O interface is selected for area n. (n = 0 to 7)	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Do not write to the CSnCR register while access to the CSn area is in progress.

EXENB Bit (Operation Enable)

This bit enables or disables operation of the respective CS areas.

After this MCU 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.

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 write to the CSnREC register while access to the CSn area is in progress.
When the preceding bus access is a separate bus access, CSnREC is valid when the recovery cycle insertion is enabled with the separate bus recovery cycle insertion enable bit (RCVENj (j = 0 to 7)) in CSRECEN. When the preceding bus access is an address/data multiplexed bus access, CSnREC is valid when the recovery cycle insertion is enabled with the multiplexed bus recovery cycle insertion enable bit (RCVENMj) in CSRECEN.

RRCV[3:0] Bits (Read Recovery)

These bits specify the number of recovery cycles to be inserted after a read access to the external bus.
When the recovery cycle insertion is enabled and a value except 0000b is written to these bits, one to 15 recovery cycles are inserted in the following cases.

- After a read access to the external bus, a read access is made to the external bus in the same area.
- After a read access to the external bus, a read access is made to the external bus in a different area.
- After a read access to the external bus, a write access is made to the external bus in the same area.
- After a read access to the external bus, a write access is made to the external bus in a different area.

WRCV[3:0] Bits (Write Recovery)

These bits specify the number of recovery cycles to be inserted after a write access to the external bus.
When the recovery cycle insertion is enabled and a value except 0000b is written to these bits, one to 15 recovery cycles are inserted in the following cases.

- After a write access to the external bus, a read access is made to the external bus in the same area.
- After a write access to the external bus, a read access is made to the external bus in a different area.
- After a write access to the external bus, a write access is made to the external bus in the same area.
- After a write access to the external bus, a write access is made to the external bus in a different area.

16.3.3 CS Recovery Cycle Insertion Enable Register (CSRECEN)

Address(es): 0008 3880h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCVENM7	RCVENM6	RCVENM5	RCVENM4	RCVENM3	RCVENM2	RCVENM1	RCVENM0	RCVEN7	RCVEN6	RCVEN5	RCVEN4	RCVEN3	RCVEN2	RCVEN1	RCVEN0
Value after reset:	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	RCVEN0	Separate Bus Recovery Cycle Insertion Enable 0	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b1	RCVEN1	Separate Bus Recovery Cycle Insertion Enable 1	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b2	RCVEN2	Separate Bus Recovery Cycle Insertion Enable 2	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b3	RCVEN3	Separate Bus Recovery Cycle Insertion Enable 3	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b4	RCVEN4	Separate Bus Recovery Cycle Insertion Enable 4	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b5	RCVEN5	Separate Bus Recovery Cycle Insertion Enable 5	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b6	RCVEN6	Separate Bus Recovery Cycle Insertion Enable 6	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b7	RCVEN7	Separate Bus Recovery Cycle Insertion Enable 7	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b8	RCVENM0	Multiplexed Bus Recovery Cycle Insertion Enable 0	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b9	RCVENM1	Multiplexed Bus Recovery Cycle Insertion Enable 1	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b10	RCVENM2	Multiplexed Bus Recovery Cycle Insertion Enable 2	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b11	RCVENM3	Multiplexed Bus Recovery Cycle Insertion Enable 3	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b12	RCVENM4	Multiplexed Bus Recovery Cycle Insertion Enable 4	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b13	RCVENM5	Multiplexed Bus Recovery Cycle Insertion Enable 5	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b14	RCVENM6	Multiplexed Bus Recovery Cycle Insertion Enable 6	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b15	RCVENM7	Multiplexed Bus Recovery Cycle Insertion Enable 7	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W

Do not write to the CSRECEN register while access to the CSn area is in progress.

RCVEN0 Separate Bus Recovery Cycle Insertion Enable 0

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.

RCVEN1 Separate Bus Recovery Cycle Insertion Enable 1

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 different area.

RCVEN2 Separate Bus Recovery Cycle Insertion Enable 2

This bit enables or disables the insertion of read recovery cycles when, after a read access to the external bus, a write access is made to the external bus in the same area.

RCVEN3 Separate Bus Recovery Cycle Insertion Enable 3

This bit enables or disables the insertion of read recovery cycles when, after a read access to the external bus, a write access is made to the external bus in the different area.

RCVEN4 Separate Bus Recovery Cycle Insertion Enable 4

This bit enables or disables the insertion of write recovery cycles when, after a write access to the external bus, a read access is made to the external bus in the same area.

RCVEN5 Separate Bus Recovery Cycle Insertion Enable 5

This bit enables or disables the insertion of write recovery cycles when, after a write access to the external bus, a read access is made to the external bus in the different area.

RCVEN6 Separate Bus Recovery Cycle Insertion Enable 6

This bit enables or disables the insertion of write recovery cycles when, after a write access to the external bus, a write access is made to the external bus in the same area.

RCVEN7 Separate Bus Recovery Cycle Insertion Enable 7

This bit enables or disables the insertion of write recovery cycles when, after a write access to the external bus, a write access is made to the external bus in the different area.

RCVENM0 Multiplexed Bus Recovery Cycle Insertion Enable 0

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.

RCVENM1 Multiplexed Bus Recovery Cycle Insertion Enable 1

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 different area.

RCVENM2 Multiplexed Bus Recovery Cycle Insertion Enable 2

This bit enables or disables the insertion of read recovery cycles when, after a read access to the external bus, a write access is made to the external bus in the same area.

RCVENM3 Multiplexed Bus Recovery Cycle Insertion Enable 3

This bit enables or disables the insertion of read recovery cycles when, after a read access to the external bus, a write access is made to the external bus in the different area.

RCVENM4 Multiplexed Bus Recovery Cycle Insertion Enable 4

This bit enables or disables the insertion of write recovery cycles when, after a write access to the external bus, a read access is made to the external bus in the same area.

RCVENM5 Multiplexed Bus Recovery Cycle Insertion Enable 5

This bit enables or disables the insertion of write recovery cycles when, after a write access to the external bus, a read access is made to the external bus in the different area.

RCVENM6 Multiplexed Bus Recovery Cycle Insertion Enable 6

This bit enables or disables the insertion of write recovery cycles when, after a write access to the external bus, a write access is made to the external bus in the same area.

RCVENM7 Multiplexed Bus Recovery Cycle Insertion Enable 7

This bit enables or disables the insertion of write recovery cycles when, after a write access to the external bus, a write access is made to the external bus in the different area.

Table 16.8 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
Write access after reed 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
Reed access after write access	Same area	Recovery cycles specified by the WRCV[3:0] bits are inserted.	RCVEN4/RCVENM4
	Different area	Recovery cycles specified by the WRCV[3:0] bits are inserted.	RCVEN5/RCVENM5
Write access after write access	Same area	Recovery cycles specified by the WRCV[3:0] bits are inserted.	RCVEN6/RCVENM6
	Different area	Recovery cycles specified by the WRCV[3:0] bits are inserted.	RCVEN7/RCVENM7

16.3.4 CSn Mode Register (CSnMOD) (n = 0 to 7)

Address(es): CS0MOD 0008 3002h, CS1MOD 0008 3012h, CS2MOD 0008 3022h, CS3MOD 0008 3032h, CS4MOD 0008 3042h, CS5MOD 0008 3052h, CS6MOD 0008 3062h, CS7MOD 0008 3072h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PRMOD	—	—	—	—	—	PWENB	PRENB	—	—	—	—	EWENB	—	—	WRMOD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WRMOD	Write Access Mode Select	0: Byte strobe mode 1: Single write strobe mode	R/W
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	EWENB	External Wait Enable	0: External wait is disabled 1: External wait is enabled	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	PRENB	Page Read Access Enable	0: Page read access is disabled 1: Page read access is enabled	R/W
b9	PWENB	Page Write Access Enable	0: Page write access is disabled 1: Page write access is enabled	R/W
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	PRMOD	Page Read Access Mode Select	0: Normal access compatible mode 1: External data read continuous assertion mode	R/W

Do not write to the CSnMOD register while access to the CSn area is in progress.

WRMOD Bit (Write Access Mode Select)

This bit selects a write access operating mode.

Writing 0 to this bit selects byte strobe mode where data write operation is controlled by the WRn# (n = 0, 1) 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, 1) 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 16.9 Control Signals for Write Access Mode

Mode	Pin Name			
Write Access Mode	**WR1#**	**WR0#/WR#**	**BC1#**	**BC0#**
Byte strobe mode	✓	✓ (WR0#)	×	×
Single write strobe mode	×	✓ (WR#)	✓	✓

✓: Enabled, ×: Disabled

EWENB Bit (External Wait Enable)

This bit enables or disables external wait.

Writing 1 to this bit selects external wait and allows control of the number of waits in each cycle with the WAIT# signal. In this state, wait cycles are inserted while the WAIT# signal is at the low level.

Writing 0 to this bit disables the WAIT# signal.

PRENB Bit (Page Read Access Enable)

This bit enables or disables page read accesses.

Note: When the address/data multiplexed I/O interface is selected with the MPXEN bit in CSnCR, this bit should not be set to enable page read accesses. Page read accesses are not supported in the address/data multiplexed I/O interface.

PWENB Bit (Page Write Access Enable)

This bit enables or disables page write accesses.

Note: When the address/data multiplexed I/O interface is selected with the MPXEN bit in CSnCR, this bit should not be set to enable page write accesses. Page write accesses are not supported in the address/data multiplexed I/O interface.

PRMOD Bit (Page Read Access Mode Select)

This bit selects a page read access operating mode.

Writing 0 to this bit selects normal access compatible mode where the RD# signal is negated and RD assert wait is inserted each time a piece of data is read. However, when there is no RD assert wait, the RD# signal is negated only in the final transfer of the external bus access.

Writing 1 to this bit selects external data read continuous assertion mode where RD assert wait is inserted and the RD# signal is continuously asserted during this time period.

16.3.5 CSn Wait Control Register 1 (CSnWCR1) (n = 0 to 7)

Address(es): CS0WCR1 0008 3004h, CS1WCR1 0008 3014h, CS2WCR1 0008 3024h, CS3WCR1 0008 3034h, CS4WCR1 0008 3044h, CS5WCR1 0008 3054h, CS6WCR1 0008 3064h, CS7WCR1 0008 3074h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	CSRWAIT[4:0]					—	—	—	CSWWAIT[4:0]				
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CSPRWAIT[2:0]			—	—	—	—	—	CSPWWAIT[2:0]		
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CSPWWAIT[2:0]	Page Write Cycle Wait Select*1	b2 b0 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	CSPRWAIT[2:0]	Page Read Cycle Wait Select*2	b10 b8 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b20 to b16	CSWWAIT[4:0]	Normal Write Cycle Wait Select	b20 b16 0 0 0 0 0: No wait is inserted. 0 0 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 0 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 0 0 1 1: Wait with a length of 3 clock cycles are inserted. 0 0 1 0 0: Wait with a length of 4 clock cycles are inserted. 0 0 1 0 1: Wait with a length of 5 clock cycles are inserted. 0 0 1 1 0: Wait with a length of 6 clock cycles are inserted. 0 0 1 1 1: Wait with a length of 7 clock cycles are inserted. 0 1 0 0 0: Wait with a length of 8 clock cycles are inserted. 0 1 0 0 1: Wait with a length of 9 clock cycles are inserted. 0 1 0 1 0: Wait with a length of 10 clock cycles are inserted. 0 1 0 1 1: Wait with a length of 11 clock cycles are inserted. 0 1 1 0 0: Wait with a length of 12 clock cycles are inserted. 0 1 1 0 1: Wait with a length of 13 clock cycles are inserted. 0 1 1 1 0: Wait with a length of 14 clock cycles are inserted. 0 1 1 1 1: Wait with a length of 15 clock cycles are inserted. 1 0 0 0 0: Wait with a length of 16 clock cycles are inserted. 1 0 0 0 1: Wait with a length of 17 clock cycles are inserted. 1 0 0 1 0: Wait with a length of 18 clock cycles are inserted. 1 0 0 1 1: Wait with a length of 19 clock cycles are inserted. 1 0 1 0 0: Wait with a length of 20 clock cycles are inserted. 1 0 1 0 1: Wait with a length of 21 clock cycles are inserted. 1 0 1 1 0: Wait with a length of 22 clock cycles are inserted. 1 0 1 1 1: Wait with a length of 23 clock cycles are inserted. 1 1 0 0 0: Wait with a length of 24 clock cycles are inserted. 1 1 0 0 1: Wait with a length of 25 clock cycles are inserted. 1 1 0 1 0: Wait with a length of 26 clock cycles are inserted. 1 1 0 1 1: Wait with a length of 27 clock cycles are inserted. 1 1 1 0 0: Wait with a length of 28 clock cycles are inserted. 1 1 1 0 1: Wait with a length of 29 clock cycles are inserted. 1 1 1 1 0: Wait with a length of 30 clock cycles are inserted. 1 1 1 1 1: Wait with a length of 31 clock cycles are inserted.	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28 to b24	CSRWAIT[4:0]	Normal Read Cycle Wait Select	b28 b24 0 0 0 0 0: No wait is inserted. 0 0 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 0 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 0 0 1 1: Wait with a length of 3 clock cycles are inserted. 0 0 1 0 0: Wait with a length of 4 clock cycles are inserted. 0 0 1 0 1: Wait with a length of 5 clock cycles are inserted. 0 0 1 1 0: Wait with a length of 6 clock cycles are inserted. 0 0 1 1 1: Wait with a length of 7 clock cycles are inserted. 0 1 0 0 0: Wait with a length of 8 clock cycles are inserted. 0 1 0 0 1: Wait with a length of 9 clock cycles are inserted. 0 1 0 1 0: Wait with a length of 10 clock cycles are inserted. 0 1 0 1 1: Wait with a length of 11 clock cycles are inserted. 0 1 1 0 0: Wait with a length of 12 clock cycles are inserted. 0 1 1 0 1: Wait with a length of 13 clock cycles are inserted. 0 1 1 1 0: Wait with a length of 14 clock cycles are inserted. 0 1 1 1 1: Wait with a length of 15 clock cycles are inserted. 1 0 0 0 0: Wait with a length of 16 clock cycles are inserted. 1 0 0 0 1: Wait with a length of 17 clock cycles are inserted. 1 0 0 1 0: Wait with a length of 18 clock cycles are inserted. 1 0 0 1 1: Wait with a length of 19 clock cycles are inserted. 1 0 1 0 0: Wait with a length of 20 clock cycles are inserted. 1 0 1 0 1: Wait with a length of 21 clock cycles are inserted. 1 0 1 1 0: Wait with a length of 22 clock cycles are inserted. 1 0 1 1 1: Wait with a length of 23 clock cycles are inserted. 1 1 0 0 0: Wait with a length of 24 clock cycles are inserted. 1 1 0 0 1: Wait with a length of 25 clock cycles are inserted. 1 1 0 1 0: Wait with a length of 26 clock cycles are inserted. 1 1 0 1 1: Wait with a length of 27 clock cycles are inserted. 1 1 1 0 0: Wait with a length of 28 clock cycles are inserted. 1 1 1 0 1: Wait with a length of 29 clock cycles are inserted. 1 1 1 1 0: Wait with a length of 30 clock cycles are inserted. 1 1 1 1 1: Wait with a length of 31 clock cycles are inserted.	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The CSPWWAIT[2:0] value is valid only when the PWENB bit in CSnMOD is set to 1.
Note 2. The CSPRWAIT[2:0] value is valid only when the PRENB bit in CSnMOD is set to 1.

Do not write to the CSnWCR1 register while access to the CSn area is in progress.
Set each of these bits within a range of the restrictions described in section 16.5.7 (1) Limitations on Using Separate Bus Interface or section 16.5.7 (2) Limitations on Using Address/Data Multiplexed Bus Interface, according to the bus interface used. In addition, during the EXDMAC transfer in single address mode, set each of these bits within a range of the restrictions described in section 16.5.7 (5) Limitations on EXDMAC Single Address Transfer Mode.

CSPWWAIT[2:0] Bits (Page Write Cycle Wait Select)

These bits specify the number of wait cycles to be inserted into the second and subsequent accesses during a page write cycle.
This setting is enabled when the PWENB bit in CSnMOD is set to 1.

Note: Be sure to satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.

CSPRWAIT[2:0] Bits (Page Read Cycle Wait Select)

These bits specify the number of wait cycles to be inserted into the second and subsequent accesses during a page read cycle.
This setting is enabled when the PRENB bit in CSnMOD is set to 1.

Note: Be sure to satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSPRWAIT[2:0] value.

CSWWAIT[4:0] Bits (Normal Write Cycle Wait Select)

These bits specify the number of wait cycles to be inserted into the first access during a normal write cycle or page write cycle.

Note: Be sure to satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

CSRWAIT[4:0] Bits (Normal Read Cycle Wait Select)

These bits specify the number of wait cycles to be inserted into the first access during a normal read cycle or page read cycle.

Note: Be sure to satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.

16.3.6 CSn Wait Control Register 2 (CSnWCR2) (n = 0 to 7)

Address(es): CS0WCR2 0008 3008h, CS1WCR2 0008 3018h, CS2WCR2 0008 3028h, CS3WCR2 0008 3038h, CS4WCR2 0008 3048h, CS5WCR2 0008 3058h, CS6WCR2 0008 3068h, CS7WCR2 0008 3078h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	CSON[2:0]			—	WDON[2:0]			—	WRON[2:0]			—	RDON[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	AWAIT[1:0]		—	WDOFF[2:0]			—	CSWOFF[2:0]			—	CSROFF[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CSROFF[2:0]	Read-Access CS Extension Cycle Select	b2 b1 b0 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	CSWOFF[2:0]	Write-Access CS Extension Cycle Select	b6 b5 b4 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b8	WDOFF[2:0]	Write Data Output Extension Cycle Select	b10 b9 b8 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b13, b12	AWAIT[1:0]	Address Cycle Wait Select	b13 b12 0 0: No wait is inserted. 0 1: Wait with a length of 1 clock cycle is inserted. 1 0: Wait with a length of 2 clock cycles are inserted. 1 1: Wait with a length of 3 clock cycles are inserted.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	RDON[2:0]	RD Assert Wait Select	b18 b17 b16 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b22 to b20	WRON[2:0]	WR Assert Wait Select	b22 b20 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b26 to b24	WDON[2:0]	Write Data Output Wait Select	b26 b24 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b30 to b28	CSON[2:0]	CS Assert Wait Select	b30 b28 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Do not write to the CSnWCR2 register while access to the CSn area is in progress.
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, 1) 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, 1) negated) until the write data output is completed in write access mode.
When the EXDMAC is in single-address transfer mode, although the output of write-data from the chip does not proceed, the value for cycles of delay until output of the write data in divided-up page access over the bus becomes effective.

Note: Be sure to satisfy CSnWCR2.WDOFF[2:0] value ≤ CSnWCR2.CSWOFF[2:0] value.

AWAIT[1:0] Bits (Address Cycle Wait Select)

These bits specify the number of wait cycles to be inserted into an address output cycle with the address/data multiplexed I/O interface.

Note: CSnWCR2.CSON[2:0] value ≤ CSnWCR2.AWAIT[1:0] value
For read access, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.
For write access, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value and CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

RDON[2:0] Bits (RD Assert Wait Select)

These bits specify the number of wait cycles to be inserted before the RD# signal is asserted.

Note: For normal read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.
For page read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSPRWAIT[2:0] value.
Note: For read access by the EXDMAC in single-address transfer mode, as well as satisfying the above conditions, set the CSnWCR2.RDON[2:0] bits to one or a greater value.
Note: When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.

WRON[2:0] Bits (WR Assert Wait Select)

These bits specify the number of wait cycles to be inserted before the WRn# signal (n = 0, 1) is asserted.

Note: For normal write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.
For page write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value and CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.
Note: For write access by the EXDMAC in single-address transfer mode, as well as satisfying the above conditions, set the CSnWCR2.WRON[2:0] bits to one or a greater value.
Note: When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

WDON[2:0] Bits (Write Data Output Wait Select)

These bits specify the number of wait cycles to be inserted before the write data is output.

Note: For normal write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.
For page write access, satisfy 1 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.
Note: When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.AWAIT[1:0] value + 2 ≤ CSnWCR2.WDON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.

CSON[2:0] Bits (CS Assert Wait Select)

These bits specify the number of wait cycles to be inserted before the CSn# signal (n = 0 to 7) is asserted.

Note: For normal read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSRWAIT[4:0] value.
For page read access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.RDON[2:0] value ≤ CSnWCR1.CSPRWAIT[2:0] value.

For normal write access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSWWAIT[4:0] value.
For page write access, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.WRON[2:0] value ≤ CSnWCR1.CSPWWAIT[2:0] value.

Note: When the address/data multiplexed I/O interface is selected, satisfy CSnWCR2.CSON[2:0] value ≤ CSnWCR2.AWAIT[1:0] value.

16.3.7 SDC Control Register (SDCCR)

Address(es): 0008 3C00h

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

Bit	Symbol	Bit Name	Description	R/W
b0	EXENB	Operation Enable	0: Operation is disabled 1: Operation is enabled	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	BSIZE[1:0]	SDRAM Bus Width Select	b5 b4 0 0: A 16-bit bus space is selected 0 1: Setting prohibited 1 0: An 8-bit bus space is selected 1 1: Setting prohibited	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

EXENB Bit (Operation Enable)

This bit enables or disables the operation of the SDRAM address space. After reset, this bit is set to 0 (operation disabled). An attempt at access to an area for which operation has been disabled does not lead to SDRAM access. If the illegal address access detection enable bit (IGAEN) in the bus error monitoring enable register (BEREN) has been set to 1 (detection enabled), such an attempt will lead to a bus error.

16.3.8 SDC Mode Register (SDCMOD)

Address(es): 0008 3C01h

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

Bit	Symbol	Bit Name	Description	R/W
b0	EMODE	Endian Mode	0: Endian of SDRAM address space is the same as the endian of operating mode. 1: Endian of SDRAM address space is not the endian of operating mode.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Writing to this register is possible only once after release from the reset state. Operation is not guaranteed after write access more than once is attempted.

EMODE Bit (Endian Mode)

This bit specifies the endian of the SDRAM address space.
When the endian setting for each area is different from that for the chip, no instruction code can be allocated in the area. The instruction code should be allocated to the external space whose endian setting is the same as that for the chip.

16.3.9 SDRAM Access Mode Register (SDAMOD)

Address(es): 0008 3C02h

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

Bit	Symbol	Bit Name	Description	R/W
b0	BE	Continuous Access Enable	0: Continuous access is disabled 1: Continuous access is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Set SDAMOD while the conditions listed in Table 16.16, Conditions for Register Modification, are satisfied. The operation is not guaranteed if this register is set while these conditions are not satisfied.

BE Bit (Continuous Access Enable)

This bit enables or disables continuous access to the SDRAM access space.

Note: When the SDRAM area is accessed from bus masters other than EXDMAC, continuous access is always disabled regardless of the setting.

16.3.10 SDRAM Self-Refresh Control Register (SDSELF)

Address(es): 0008 3C10h

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

Bit	Symbol	Bit Name	Description	R/W
b0	SFEN	SDRAM Self-Refresh Enable	0: Self-refresh is disabled 1: Self-refresh is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Set SDSELF while the conditions listed in Table 16.16, Conditions for Register Modification, are satisfied. The operation is not guaranteed if this register is set while these conditions are not satisfied.

SFEN Bit (SDRAM Self-Refresh Enable)

This bit controls self-refresh operation.
Setting this bit to 1 performs auto-refresh cycle operation, after which self-refresh operation begins.
Clearing this bit to 0 ends self-refresh operation, and auto-refresh operation resumes afterwards.
If this bit was set to 1, the value written to this bit is reflected when self-refresh operation starts. If this bit was set to 0, the value written to this bit has already been reflected when auto-refresh operation starts following the end of self-refresh operation.

16.3.11 SDRAM Refresh Control Register (SDRFCR)

Address(es): 0008 3C14h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REFW[3:0]				RFC[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	RFC[11:0]	Auto-Refresh Request Interval Setting	b11 b0 0 0 0 0 0 0 0 0 0 0 0 0: Setting prohibited 0 0 0 0 0 0 0 0 0 0 0 1: 2 cycles 0 0 0 0 0 0 0 0 0 0 1 0: 3 cycles : 1 1 1 1 1 1 1 1 1 1 1 1: 4096 cycles	R/W
b15 to b12	REFW[3:0]	Auto-Refresh Cycle/ Self-Refresh Clearing Cycle Count Setting	b15 b12 0 0 0 0: 1 cycle 0 0 0 1: 2 cycles 0 0 1 0: 3 cycles 0 0 1 1: 4 cycles 0 1 0 0: 5 cycles 0 1 0 1: 6 cycles 0 1 1 0: 7 cycles 0 1 1 1: 8 cycles 1 0 0 0: 9 cycles 1 0 0 1: 10 cycles 1 0 1 0: 11 cycles 1 0 1 1: 12 cycles 1 1 0 0: 13 cycles 1 1 0 1: 14 cycles 1 1 1 0: 15 cycles 1 1 1 1: 16 cycles	R/W

RFC[11:0] Bits (Auto-Refresh Request Interval Setting)

These bits specify the auto-refresh request interval.

These bits can be written to at any time, regardless of the state of the auto-refresh operation enable (RFEN) bit in SDRFEN.

If auto-refresh is enabled, the value written to these bits is reflected after the end of auto-refresh cycles. The refresh counter operates in SDCLK.

REFW[3:0] Bits (Auto-Refresh Cycle/ Self-Refresh Clearing Cycle Count Setting)

These bits specify the number of auto-refresh cycles and the number of self-refresh clearing cycles.

These bits can be written to at any time, regardless of the state of the auto-refresh operation enable (RFEN) bit in SDRFEN.

If an auto-refresh cycle is in progress, the value written to these bits while auto-refresh is enabled takes effect after the cycle completes.

Note: Auto-refresh requests are not accepted while SDRAM is being accessed; they must wait until the access completes, so the auto-refresh interval may become enlarged in some cases. Set the RFC[11:0] bits to an auto-refresh request interval value that satisfies the auto-refresh interval specification of the SDRAM being used. Furthermore, make sure to set the auto-refresh request interval to a duration longer than the auto-refresh cycle. Note that the auto-refresh interval cannot be automatically adjusted when the frequency is changed during operation; in this case, perform self-refresh operation and set the auto-refresh interval appropriate for the frequency again.

- Auto-Refresh Request Interval and RFC Set Value

SDRAMC (SDRAM area controller) includes a 12-bit refresh counter that generates auto-refresh requests at fixed intervals. The following equation is used to calculate the set value for the RFC[11:0] bits from the auto-refresh request interval.

RFC = (Auto-refresh request interval / SDCLK cycle) – 1

Note: Auto-refresh requests are not accepted while SDRAM is being accessed; they must wait until the access completes. However, the counter value is updated regardless of whether or not the request was accepted. Note that if two or more auto-refresh requests are generated while SDRAM is being accessed, the second and subsequent requests are ignored.

16.3.12 SDRAM Auto-Refresh Control Register (SDRFEN)

Address(es): 0008 3C16h

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

Bit	Symbol	Bit Name	Description	R/W
b0	RFEN	Auto-Refresh Operation Enable	0: Auto-refresh operation is disabled 1: Auto-refresh operation is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RFEN Bit (Auto-Refresh Operation Enable)

Setting this bit to 0 while auto-refreshing is enabled causes RFEN to be set to 0 and auto-refresh operation to halt after the end of the auto-refresh cycle. However, if RFEN is again set to 1 before the end of the auto-refresh cycle, auto-refreshing continues and the RFEN bit is not set to 0. Setting the RFEN bit to 1 while auto-refresh is disabled starts auto-refresh operation, and refresh requests are then generated at fixed intervals determined by a counter. The interval at which refresh requests are generated is determined by the value of the auto-refresh request interval setting (RFC[11:0]) bits in the SDRAM refresh control register (SDRFCR).

Refresh requests are not accepted while SDRAM is being accessed; they must wait until the access completes.

If an SDRAM access and a refresh request are generated at the same time, the refresh request takes precedence.

16.3.13 SDRAM Initialization Sequence Control Register (SDICR)

Address(es): 0008 3C20h

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

Bit	Symbol	Bit Name	Description	R/W
b0	INIRQ	Initialization Sequence Start	0: Invalid 1: Initialization sequence starts	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Writing to this register is possible only once after release from the reset state. Operation is not guaranteed after write access more than once is attempted.

INIRQ Bit (Initialization Sequence Start)

Setting this bit to 1 causes the SDRAM initialization sequence to start and automatically sets the initialization status bit (INIST) in the SDRAM status register (SDSR) to 1. The INIST bit is cleared automatically after the initialization sequence ends.

The value written to the INIRQ bit is not retained.

If access to an external address space or an external bus controller register occurs after the initialization sequence is started, the access is suspended until the initialization sequence ends.

Note: Set the INIRQ bit to start the SDRAM initialization sequence while the conditions listed in Table 16.16, Conditions for Register Modification, are satisfied. The operation is not guaranteed if this bit is set while these conditions are not satisfied.

16.3.14 SDRAM Initialization Register (SDIR)

Address(es): 0008 3C24h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	PRC[2:0]			ARFC[3:0]				ARFI[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	ARFI[3:0]	Initialization Auto-Refresh Interval	b3 b0 0 0 0 0: 3 cycles 0 0 0 1: 4 cycles 0 0 1 0: 5 cycles 0 0 1 1: 6 cycles 0 1 0 0: 7 cycles 0 1 0 1: 8 cycles 0 1 1 0: 9 cycles 0 1 1 1: 10 cycles 1 0 0 0: 11 cycles 1 0 0 1: 12 cycles 1 0 1 0: 13 cycles 1 0 1 1: 14 cycles 1 1 0 0: 15 cycles 1 1 0 1: 16 cycles 1 1 1 0: 17 cycles 1 1 1 1: 18 cycles	R/W
b7 to b4	ARFC[3:0]	Initialization Auto-Refresh Count	b7 b4 0 0 0 0: Setting prohibited 0 0 0 1: 1 time 0 0 1 0: 2 times 0 0 1 1: 3 times 0 1 0 0: 4 times 0 1 0 1: 5 times 0 1 1 0: 6 times 0 1 1 1: 7 times 1 0 0 0: 8 times 1 0 0 1: 9 times 1 0 1 0: 10 times 1 0 1 1: 11 times 1 1 0 0: 12 times 1 1 0 1: 13 times 1 1 1 0: 14 times 1 1 1 1: 15 times	R/W
b10 to b8	PRC[2:0]	Initialization Precharge Cycle Count	b10 b8 0 0 0: 3 cycles 0 0 1: 4 cycles 0 1 0: 5 cycles 0 1 1: 6 cycles 1 0 0: 7 cycles 1 0 1: 8 cycles 1 1 0: 9 cycles 1 1 1: 10 cycles	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Writing to this register is possible only once after release from the reset state. Operation is not guaranteed after write access more than once is attempted.

ARFI[3:0] Bits (Initialization Auto-Refresh Interval)

These bits specify the interval at which auto-refresh commands are issued in the SDRAM initialization sequence.

ARFC[3:0] Bits (Initialization Auto-Refresh Count)

These bits specify the number of times auto-refresh is to be performed in the SDRAM initialization sequence.

PRC[2:0] Bits (Initialization Precharge Cycle Count)

These bits specify the number of precharge cycles in the SDRAM initialization sequence.

Note: Make settings that satisfy the specifications of the connected SDRAM before starting the initialization sequence.

16.3.15 SDRAM Address Register (SDADR)

Address(es): 0008 3C40h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	MXC[1:0]	Address Multiplex Select	b1 b0 0 0: 8-bit shift 0 1: 9-bit shift 1 0: 10-bit shift 1 1: 11-bit shift	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Set SDADR while the conditions listed in Table 16.16, 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 16.21, Address Multiplexing.

16.3.16 SDRAM Timing Register (SDTR)

Address(es): 0008 3C44h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	RAS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RCD[1:0]		RP[2:0]			WR	—	—	—	—	—	CL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CL[2:0]	SDRAMC Column Latency	b2 b0 0 0 0: Setting prohibited 0 0 1: 1 cycle 0 1 0: 2 cycles 0 1 1: 3 cycles 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: Setting prohibited 1 1 1: Setting prohibited	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	WR	Write Recovery Interval	0: 1 cycle 1: 2 cycles	R/W
b11 to b9	RP[2:0]	Row Precharge Interval	b11 b9 0 0 0: 1 cycle 0 0 1: 2 cycles 0 1 0: 3 cycles 0 1 1: 4 cycles 1 0 0: 5 cycles 1 0 1: 6 cycles 1 1 0: 7 cycles 1 1 1: 8 cycles	R/W
b13, b12	RCD[1:0]	Row Column Latency	b13 b12 0 0: 1 cycle 0 1: 2 cycles 1 0: 3 cycles 1 1: 4 cycles	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	RAS[2:0]	Row Active Interval	b18 b16 0 0 0: 1 cycle 0 0 1: 2 cycles 0 1 0: 3 cycles 0 1 1: 4 cycles 1 0 0: 5 cycles 1 0 1: 6 cycles 1 1 0: 7 cycles 1 1 1: Setting prohibited	R/W
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SDTR specifies the timing for read and write accesses to SDRAM. For details, refer to section 16.6.12.3, Timing Register Settings and Access Timing.

Set SDTR while the conditions listed in Table 16.16, Conditions for Register Modification, are satisfied. The operation is not guaranteed if this bit is set while these conditions are not satisfied.

Writing to this register is possible only once after release from the reset state. Operation is not guaranteed after write access more than once is attempted.

CL[2:0] Bits (SDRAMC Column Latency)

These bits specify the column latency of the SDRAM controller. This setting only affects the latency setting on the SDRAM controller side. To specify the column latency for externally connected SDRAM, use the SDRAM mode register (SDMOD), which is described below.

WR Bit (Write Recovery Interval)

This bit specifies the interval that must elapse between the SDRAM write command (WRIT) and deactivation (PALL).

RP[2:0] Bits (Row Precharge Interval)

These bits specify the minimum number of cycles that must elapse between the SDRAM deactivation command (PALL) and the next valid command.

RAS[2:0] Bits (Row Active Interval)

These bits specify the minimum interval that must elapse between the SDRAM row activation command (ACTV) and deactivation (PALL). The value specified by these bits should be less than or equal to the sum of the row-column latency (RCD[1:0]) and column latency (CL[2:0]) settings.

16.3.17 SDRAM Mode Register (SDMOD)

Address(es): 0008 3C48h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	MR[14:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b14 to b0	MR[14:0]	Mode Register Setting	Writing to these bits: Mode register set command is issued	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

SDMOD specifies the value to be written to the SDRAM mode register.
Writing to SDMOD causes a mode register set command to be issued automatically to SDRAM.
Set SDMOD while the conditions listed in Table 16.16, Conditions for Register Modification, are satisfied. The operation is not guaranteed if this bit is set while these conditions are not satisfied.

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.
- Before writing these bits, check each time that the status bits (SRFST, INIST, and MRSST) in the SDRAM status register (SDSR) are all cleared to 0.

16.3.18 SDRAM Status Register (SDSR)

Address(es): 0008 3C50h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SRFST	INIST	—	—	MRSST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MRSST	Mode Register Setting Status	0: Mode register setting not in progress 1: Mode register setting in progress	R
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	INIST	Initialization Status	0: Initialization sequence not in progress 1: Initialization sequence in progress	R
b4	SRFST	Self-Refresh Transition/Recovery Status	0: Transition/recovery not in progress 1: Transition/recovery in progress	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

MRSST Bit (Mode Register Setting Status)

When set to 1, this bit indicates that SDRAM mode register setting is in progress. If SDSR is accessed during the mode register setting operation, the CPU processing can be suspended until the setting operation ends.

INIST Bit (Initialization Status)

When set to 1, this bit indicates that the SDRAM initialization sequence is in progress. If SDSR is accessed during initialization sequence, the CPU processing can be suspended until the initialization sequence ends.

SRFST Bit (Self-Refresh Transition/Recovery Status)

When set to 1, this bit indicates that a transition to or recovery from self-refresh operation is in progress for SDRAM. “Transition to or recovery from self-refresh operation in progress” refers to the interval from the point at which the bits listed in Table 16.10 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 16.10 when any of the status bits (SRFST, INIST, MRSST) is set to 1.

Table 16.10 List of Registers and Bits Requiring Checking Status Bits

Function	Register	Bits
Self-refresh	SDSELF	SFEN
Initialization sequence	SDICR	INIRQ
Mode register setting	SDMOD	MR[14:0]

16.3.19 Bus Error Status Clear Register (BERCLR)

Address(es): 0008 1300h

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

Bit	Symbol	Bit Name	Description	R/W
b0	STSCLR	Status Clear	0: Invalid 1: Bus error status register cleared	(W)*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only writing 1 is effective; i.e. writing 0 has no effect.

STSCLR Bit (Status Clear)

Writing 1 to this bit clears the bus error status registers 1 and 2 (BERSR1 and BERSR2).
Writing 0 has no effect. It is read as 0.

16.3.20 Bus Error Monitoring Enable Register (BEREN)

Address(es): 0008 1304h

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

Bit	Symbol	Bit Name	Description	R/W
b0	IGAEN	Illegal Address Access Detection Enable	0: Illegal address access detection is disabled. 1: Illegal address access detection is enabled.	R/W
b1	TOEN	Timeout Detection Enable*1, *2	0: Bus timeout detection is disabled. 1: Bus timeout detection is enabled.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When detection is disabled (the TOEN bit is set to 0), bus access can cause the bus to freeze.
Note 2. Do not set the TOEN bit to 0 (bus timeout detection disabled) while timeout errors are being detected.

16.3.21 Bus Error Status Register 1 (BERSR1)

Address(es): 0008 1308h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	MST[2:0]			—	—	TO	IA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IA	Illegal Address Access	0: Illegal address access not made 1: Illegal address access made	R
b1	TO	Timeout	0: Timeout not generated 1: Timeout generated	R
b3, b2	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b6 to b4	MST[2:0]	Bus Master Code	b6 b4 0 0 0: CPU 0 0 1: Reserved 0 1 0: Reserved 0 1 1: DTC/DMAC 1 0 0: Reserved 1 0 1: Reserved 1 1 0: Reserved 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
	BPXB[1:0]		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 (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 (Code Flash Memory) Priority Control	b3 b2 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b5, b4	BPIB[1:0]	Internal Peripheral Bus 1 Priority Control	b5 b4 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b7, b6	BPGB[1:0]	Internal Peripheral Bus 2 and 3 Priority Control	b7 b6 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b9, b8	BPHB[1:0]	Internal Peripheral Bus 4 and 5 Priority Control	b9 b8 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b11, b10	BPFB[1:0]	Internal Peripheral Bus 6 Priority Control	b11 b10 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b13, b12	BPEB[1:0]	External Bus Priority Control	b13 b12 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R(/W) *1
b15, b14	BPXB[1:0]	Internal Expansion Bus Priority Control	b15 b14 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

Note 1. Writing to these bits is only possible once, and this must be done while the DTC, DMAC, and EXDMAC are all stopped. If the value of these bits is changed more than once, further operation is not guaranteed.

BPRA[1:0] Bits (Memory Bus 1 (RAM) Priority Control)

These bits specify the priority order for memory bus 1 (RAM).

When the priority order is fixed, internal main bus 2 has priority over CPU buses (instruction and operand buses).

When the priority order is toggled, the bus from which a request has currently been accepted has the lower priority.

BPRO[1:0] Bits (Memory Bus 2 (Code Flash Memory) Priority Control)

These bits specify the priority order for memory bus 2 (code flash memory).

When the priority order is fixed, internal main bus 2 has priority over CPU buses (instruction and operand buses).
When the priority order is toggled, the bus from which a request has currently been accepted has the lower priority.

BPIB[1:0] Bits (Internal Peripheral Bus 1 Priority Control)

These bits specify the priority order for internal peripheral bus 1.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

BPGB[1:0] Bits (Internal Peripheral Bus 2 and 3 Priority Control)

These bits specify the priority order for internal peripheral buses 2 and 3.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

BPHB[1:0] Bits (Internal Peripheral Bus 4 and 5 Priority Control)

These bits specify the priority order for internal peripheral buses 4 and 5.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

BPFB[1:0] Bits (Internal Peripheral Bus 6 Priority Control)

These bits specify the priority order for internal peripheral bus 6.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

BPEB[1:0] Bits (External Bus Priority Control)

These bits specify the priority order for the external bus.
When the priority order is fixed, the order is EXDMAC, internal main bus 2, and then internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted, between internal main bus 1 and other buses (internal main bus 2 and EXDMAC). However, the order of priority is EXDMAC and then internal main bus 2, regardless of the BPEB[1:0] bits settings.

BPXB[1:0] Bits (Internal Expansion Bus Priority Control)

These bits specify the priority order for internal expansion bus.
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.

16.4 Endian and Data Alignment

The external bus has a data-alignment function to control which byte of the data bus (D15 to D8, or D7 to D0) is used according to the bus specifications of the area to be accessed (8-bit or 16-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) 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. The BC0# and BC1# 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# pin is invalid (always output the high level). The valid byte position is indicated by the BC0# and BC1# pins.
In 16-bit bus space, page access can occur in access to data in 32- or 64-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# 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		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
64 bits	4n	Four	First	16 bits	4n		15 8 7 0
			Second	16 bits	4n+2	(p)	31 24 23 16
			Third	16 bits	4n+4		47 40 39 32
			Fourth	16 bits	4n+6	(p)	63 56 55 48

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 16.8 Data Alignment (Little Endian) in 16-Bit Bus Space

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		WR1#/BC1# RD# Data Bus D15 D8	WR0#/BC0# Data Bus D7 D0
8 bits	4n	One	First	8 bits	4n		7 0	
	4n+1	One	First	8 bits	4n			7 0
	4n+2	One	First	8 bits	4n+2		7 0	
	4n+3	One	First	8 bits	4n+2			7 0
16 bits	4n	One	First	16 bits	4n		15 8	7 0
	4n+1	Two	First	8 bits	4n			15 8
			Second	8 bits	4n+2		7 0	
	4n+2	One	First	16 bits	4n+2		15 8	7 0
	4n+3	Two	First	8 bits	4n+2			15 8
			Second	8 bits	4n+4		7 0	
32 bits	4n	Two	First	16 bits	4n		31 24	23 16
			Second	16 bits	4n+2	(p)	15 8	7 0
	4n+1	Three	First	8 bits	4n			31 24
			Second	16 bits	4n+2		23 16	15 8
			Third	8 bits	4n+4		7 0	
	4n+2	Two	First	16 bits	4n+2		31 24	23 16
			Second	16 bits	4n+4		15 8	7 0
	4n+3	Three	First	8 bits	4n+2			31 24
			Second	16 bits	4n+4		23 16	15 8
			Third	8 bits	4n+6		7 0	
64 bits	4n	Four	First	16 bits	4n		63 56	55 48
			Second	16 bits	4n+2	(p)	47 40	39 32
			Third	16 bits	4n+4		31 24	23 16
			Fourth	16 bits	4n+6	(p)	15 8	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

(2) 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# pin and the BC0# and BC1# pins are not used.

Page access can occur in access to data in 16-, 32-, or 64-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# RD# D15 D8	WR0#/BC0# D7 D0
8 bits	4n	One	First	8 bits	4n			7 0
	4n+1	One	First	8 bits	4n+1			7 0
	4n+2	One	First	8 bits	4n+2			7 0
	4n+3	One	First	8 bits	4n+3			7 0
16 bits	4n	Two	First	8 bits	4n			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
64 bits	4n	Eight	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
			Fifth	8 bits	4n+4			39 32
			Sixth	8 bits	4n+5	(p)		47 40
			Seventh	8 bits	4n+6	(p)		55 48
			Eighth	8 bits	4n+7	(p)		63 56

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 16.10 Data Alignment (Little Endian) in 8-Bit Bus Space

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address		WR1#/BC1# RD# D15 D8	WR0#/BC0# RD# D7 D0
8 bits	4n	One	First	8 bits	4n			7 0
	4n+1	One	First	8 bits	4n+1			7 0
	4n+2	One	First	8 bits	4n+2			7 0
	4n+3	One	First	8 bits	4n+3			7 0
16 bits	4n	Two	First	8 bits	4n			15 8
			Second	8 bits	4n+1	(p)		7 0
	4n+1	Two	First	8 bits	4n+1			15 8
			Second	8 bits	4n+2	(p)		7 0
	4n+2	Two	First	8 bits	4n+2			15 8
			Second	8 bits	4n+3	(p)		7 0
	4n+3	Two	First	8 bits	4n+3			15 8
			Second	8 bits	4n+4			7 0
32 bits	4n	Four	First	8 bits	4n			31 24
			Second	8 bits	4n+1	(p)		23 16
			Third	8 bits	4n+2	(p)		15 8
			Fourth	8 bits	4n+3	(p)		7 0
	4n+1	Four	First	8 bits	4n+1			31 24
			Second	8 bits	4n+2	(p)		23 16
			Third	8 bits	4n+3	(p)		15 8
			Fourth	8 bits	4n+4			7 0
	4n+2	Four	First	8 bits	4n+2			31 24
			Second	8 bits	4n+3	(p)		23 16
			Third	8 bits	4n+4			15 8
			Fourth	8 bits	4n+5	(p)		7 0
	4n+3	Four	First	8 bits	4n+3			31 24
			Second	8 bits	4n+4			23 16
			Third	8 bits	4n+5	(p)		15 8
			Fourth	8 bits	4n+6	(p)		7 0
64 bits	4n	Eight	First	8 bits	4n			63 56
			Second	8 bits	4n+1	(p)		55 48
			Third	8 bits	4n+2	(p)		47 40
			Fourth	8 bits	4n+3	(p)		39 32
			Fifth	8 bits	4n+4			31 24
			Sixth	8 bits	4n+5	(p)		23 16
			Seventh	8 bits	4n+6	(p)		15 8
			Eighth	8 bits	4n+7	(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) 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 A26 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.

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. 32-bit access is executed as two cycles of 16-bit access or three cycles in order of 8-, 16-, and 8-bit access (depending on the alignment of the address), and 64-bit access is executed as four cycles of 16-bit access.

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 16.12 and Figure 16.13 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 16.12 and Figure 16.13. Figure 16.16 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
64 bits	4n	Four	First	16 bits	4n		15 8 7 0
			Second	16 bits	4n+2		31 24 23 16
			Third	16 bits	4n+4		47 40 39 32
			Fourth	16 bits	4n+6		63 56 55 48

(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 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
64 bits	4n	Four	First	16 bits	4n		63 56 | 55 48
			Second	16 bits	4n+2		47 40 | 39 32
			Third	16 bits	4n+4		31 24 | 23 16
			Fourth	16 bits	4n+6		15 8 | 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 16-Bit Bus Space

(2) 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 A26 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 16.14 and Figure 16.15 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 16.14 and Figure 16.15. Figure 16.16 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+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
64 bits	4n	Eight	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
			Fifth	8 bits	4n+4		39 32
			Sixth	8 bits	4n+5		47 40
			Seventh	8 bits	4n+6		55 48
			Eighth	8 bits	4n+7		63 56

(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 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
64 bits	4n	Eight	First	8 bits	4n		63 56
			Second	8 bits	4n+1		55 48
			Third	8 bits	4n+2		47 40
			Fourth	8 bits	4n+3		39 32
			Fifth	8 bits	4n+4		31 24
			Sixth	8 bits	4n+5		23 16
			Seventh	8 bits	4n+6		15 8
			Eighth	8 bits	4n+7		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 8-Bit Bus Space

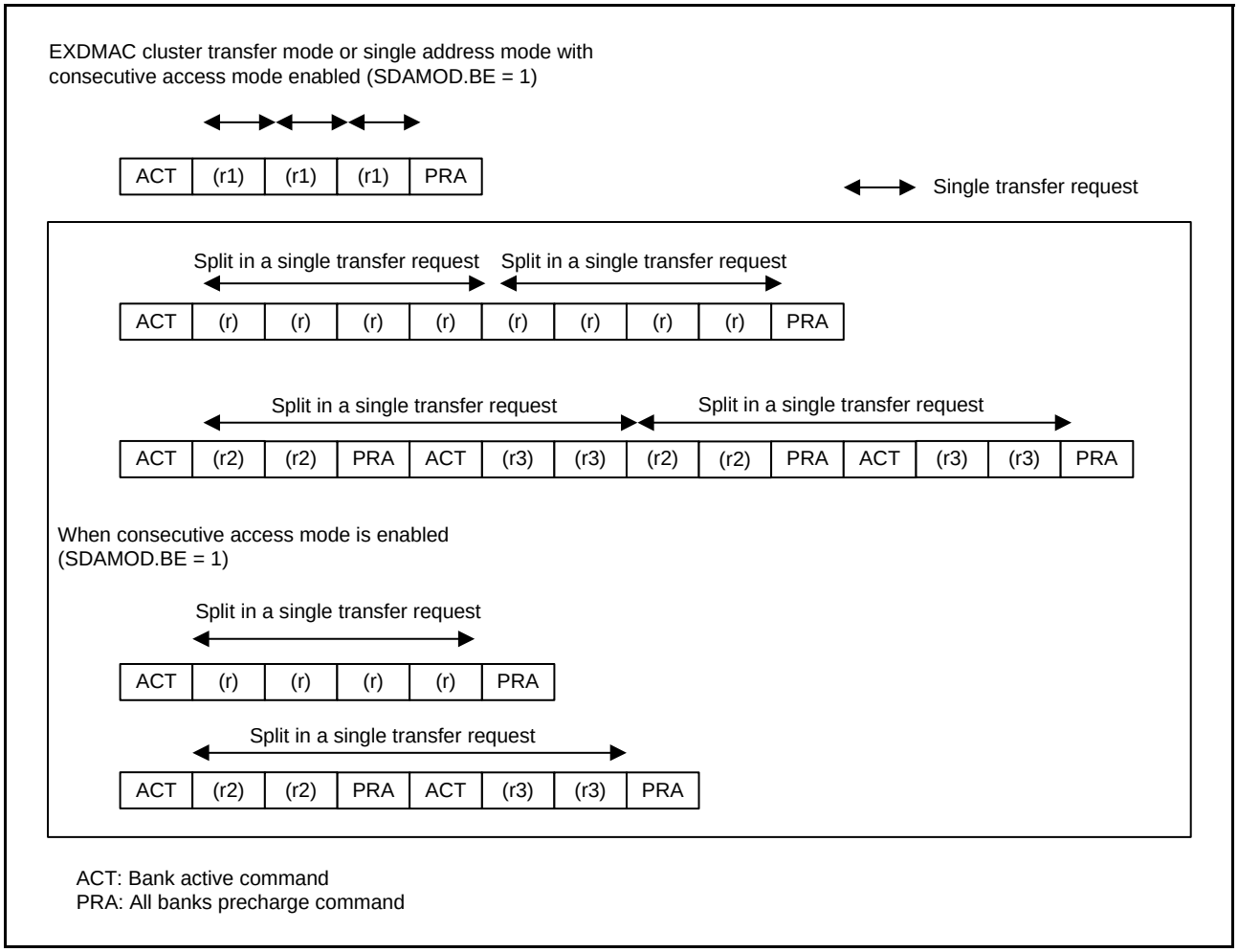


Figure 16.16 Consecutive Access Example

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 (refer to Figure 16.22 to Figure 16.26). 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 (refer to Figure 16.42).

(a) Tw1 to Twn (Clock Cycles of Waiting for a Normal Read Cycle or Normal Write Cycle)

The period Tw1 to Twn is made up of the number of clock cycles between the start of access via the external bus clock and one cycle before the strobe signal is valid. The number of cycles are selectable within the range from zero to 31. Within this period, the timing of CSn#, RD#, and WRn# assertion (placing the signals at the low level) is determined by the respective wait settings. Specifically, the periods of waiting are controlled by the CS assert wait select bits (CSON[2:0]), the RD assert wait select bits (RDON[2:0]), the WR assert wait select bits (WRON[2:0]), and the write data output wait select bits (WDON[2:0]) in CSn wait control register 2 (CSnWCR2). The number of clock cycles for each of these periods of waiting is selectable as a value from zero to seven counted from the start of external bus access. Selectable numbers of cycles are also within the overall number of clock cycles of waiting for reading or writing.

(b) Tend (Clock Cycle where the Strobe Signal is Valid)

Tend is the next clock cycle after completion of the period of waiting for a normal cycle of reading or writing or for a cycle of page reading or page writing. If each wait select bit for a normal cycle of reading or writing or for a cycle of page reading or page writing is zero, the clock cycle where bus access starts is the clock cycle where the strobe signal is valid. The RD# and WRn# signals are negated in the next clock cycle after the cycle where the strobe signal is valid. In the case of read access, the clock cycle where the strobe signal is valid becomes the clock cycle where the data to be read are sampled.
If an external wait is enabled, the wait signal is sampled at the time of the cycle where the strobe signal is valid. The bus cycle is extended if the wait signal is at the low level. The bus cycle is completed in the next clock cycle if the wait signal is at the high level. Tend indicates the cycle where sampling of the wait signal starts.
After the first cycle where the strobe signal is valid during page access, second and subsequent page access operations (point (e) below) start in the next cycle except in cases of write access where a setting (other than zero) for write-data output extension clock cycles (point (d) below) has been made. If the setting for the RD or WR assertion wait is a value other than zero, the RD# and WRn# signals are negated in the next clock cycle. If the setting is zero, assertion continues. Furthermore, the CSn# signal continues to be asserted rather than being negated.

(c) Tn1 to Tnm (Clock Cycles of CS Extension)

In the case of normal access, Tn1 to Tnm represent the clock cycles of the period following the cycle where the strobe signal is valid (Tend) up to negation of the CSn# signal. For read or write access, the timing of negation can be controlled by the read-access CS extension cycle select bits (CSROFF[2:0]) and the write-access CS extension cycle select bits (CSWOFF[2:0]) in the CSn wait control register 2 (CSnWCR2), respectively.
The number of cycles are counted from the cycle following the cycle where the strobe signal is valid.
In the case of page access, Tn1 to Tnm represent the clock cycles of the period for the cycle following the last cycle where the strobe signal is valid up to negation of the CSn# signal.

For write access, setting the write data output extension cycle select bits (WDOFF) controls extension of the period where the address and output data are valid.

(d) Tdw1 to Tdwn (Write-Data Output Extension Clock Cycles)

For write access, if the setting for write-data output extension wait is a value other than zero, clock cycles of write-data output extension are inserted from the cycle that follows the cycle where the strobe signal is valid (Tend).
In the case of normal access, this is inserted within the period of clock cycles of CS extension (point (c) above).
In the case of page access, this is inserted within the period of the cycle where the strobe signal is valid and subsequent page access or within the period of clock cycles of CS extension (point (c) above). Valid address and data output are extended over this period, and the WRn# signal is negated.

(e) Tpw1 to Tpwn (Page-Read Cycle Wait or Page-Write Cycle Wait)

For the second and subsequent bus cycles during page access, the values for a page-read cycle wait or page-write cycle wait are used instead of the settings for a normal read or write cycle wait. Setting the WR assert wait select bits becomes enabled in the same way as for the first round of access. How the setting for RD assertion controls operation depends on the setting for page-read access mode (the PRMOD bit in CSnMOD) as described below.

CSnMOD.PRMOD = 0: A wait until RD assertion is inserted in the same way as for the first round of access, and the RD# signal is negated.
CSnMOD.PRMOD = 1: Although a wait until RD assertion is inserted in the same way as for normal-access compatibility mode, the RD# signal continues to be asserted over this period.

(f) Tr1 to Trn (Recovery Cycles)

Recovery cycles can be inserted from the point where a bus cycle is completed (CSn# signal negation). The number of recovery cycles can be controlled by the setting of the read recovery (RRCV) or write recovery (WRCV) bits in the CSn recovery cycle register (CSnREC). Both numbers of recovery cycles are counted from the end of a bus cycle (CSn# negation) and can be selected from 0 to 15 cycles. For details on recovery cycles, refer to 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 16.17 to Figure 16.19 show the normal access operations.

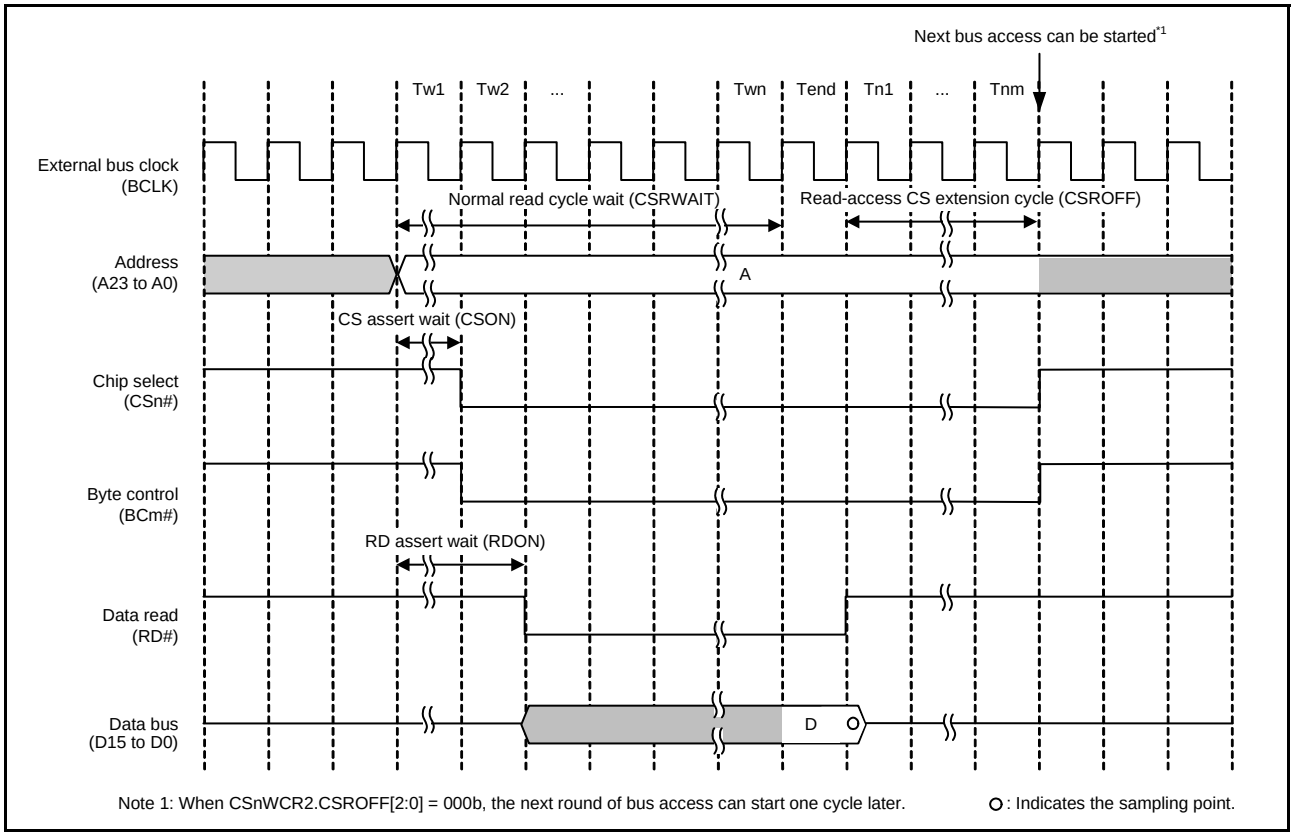


Figure 16.17 Bus Timing (Normal-Read Operation) (n = 0 to 7, m = 0, 1)

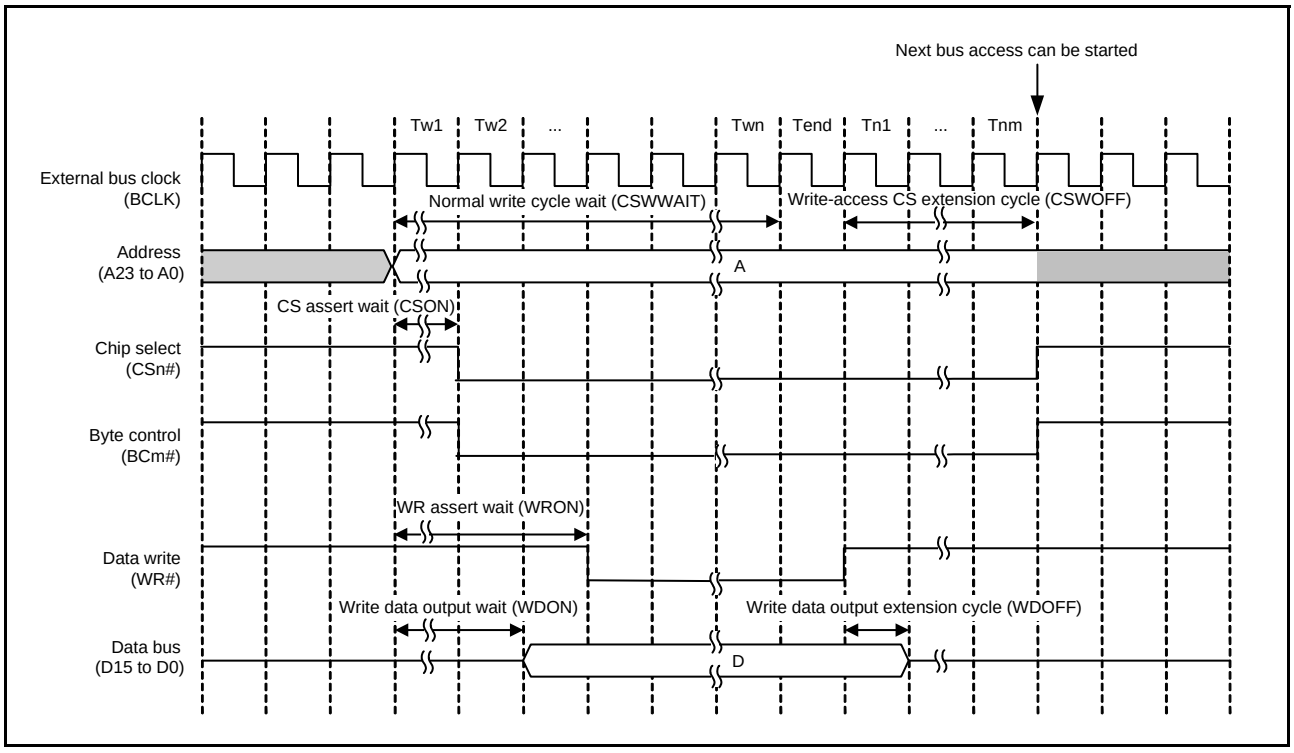


Figure 16.18 Bus Timing (Normal-Write Operation, Single Write Strobe Mode) (n = 0 to 7, m = 0, 1)

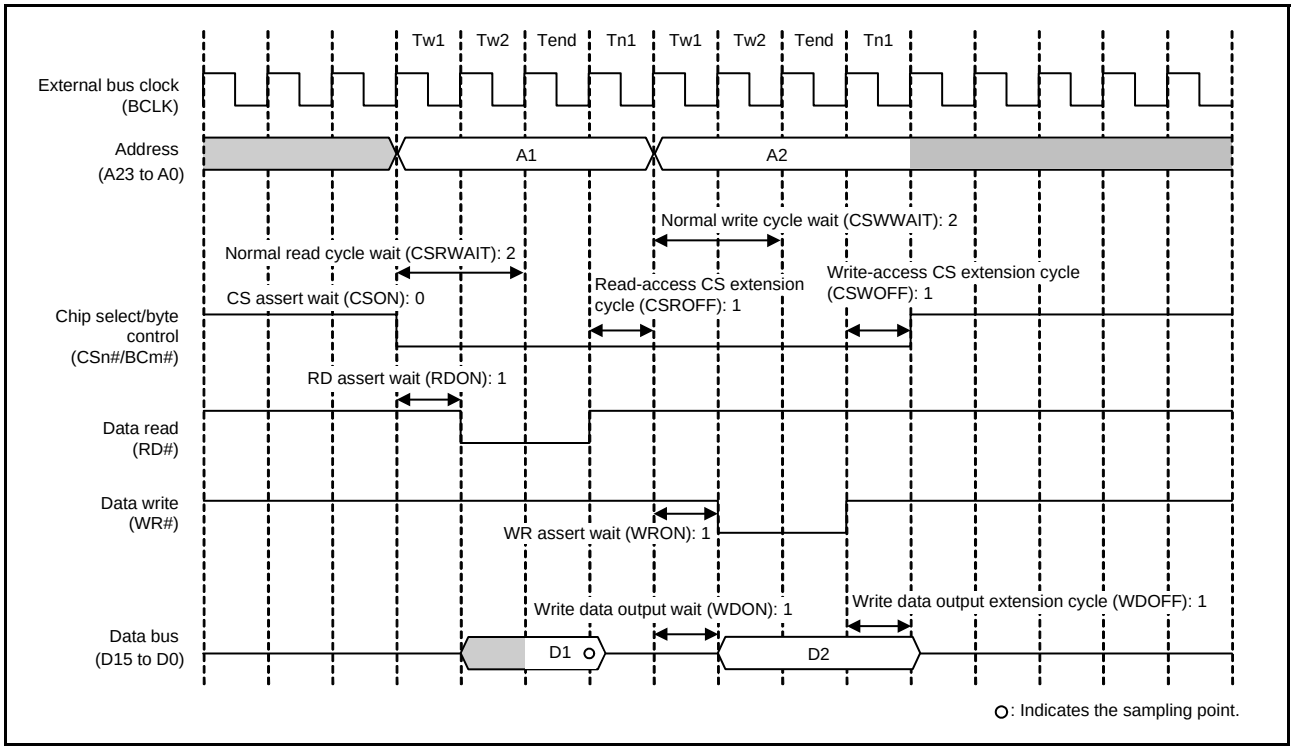


Figure 16.19 Example of Normal Access Operation (Read/Write) (n = 0 to 7, m = 0, 1)

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 16.20 and Figure 16.21 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 (refer to Figure 16.40).

The values of the wait control registers are example settings. In practice, set the register bits according to the specifications of connected devices.

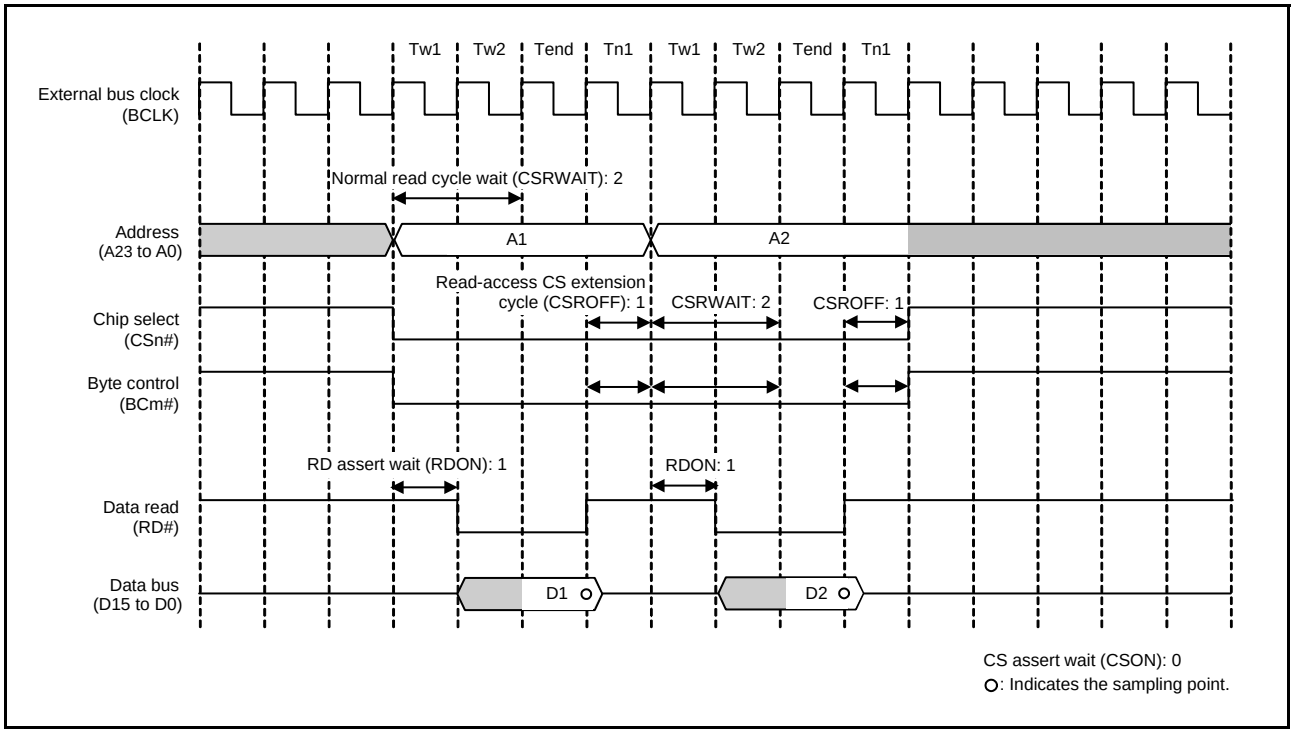


Figure 16.20 Example of Normal-Read Operation (when Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 7, m = 0, 1)

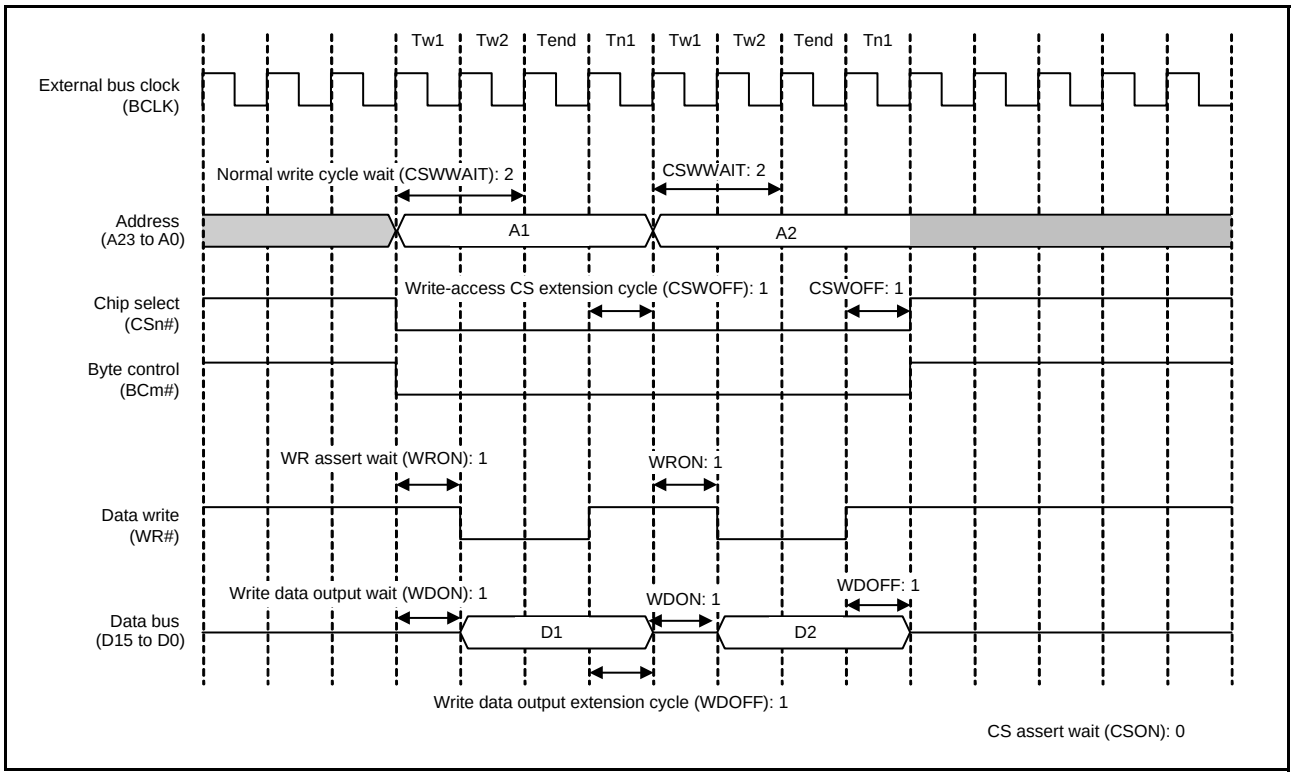


Figure 16.21 Example of Normal-Write Operation (when Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer, in Single Write Strobe Mode) (n = 0 to 7, m = 0, 1)

Figure 16.22 to Figure 16.26 show examples of normal accesses made with the 1/2 BCLK selected with the BCLK pin output select bit.

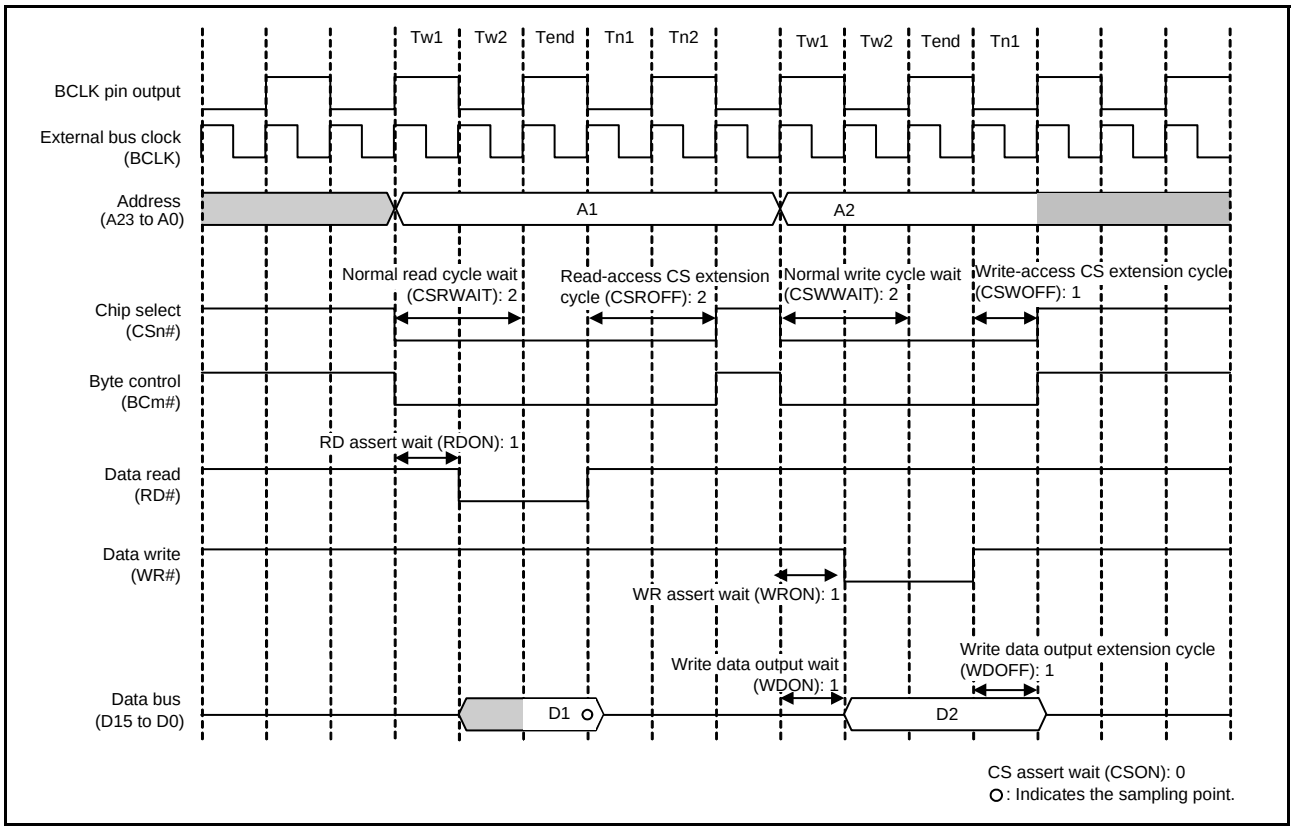


Figure 16.22 Example of Normal Access (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit) (n = 0 to 7, m = 0, 1)

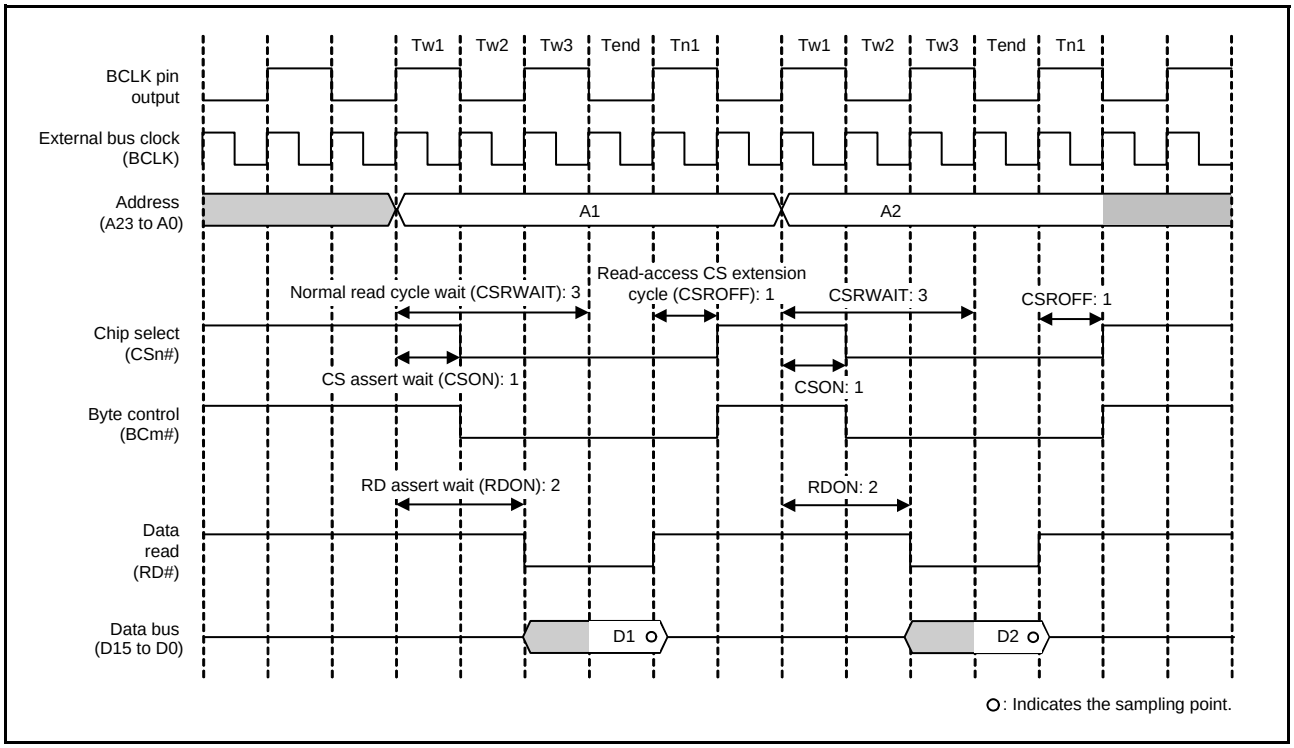


Figure 16.23 Example of Normal-Read Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit) (n = 0 to 7, m = 0, 1)

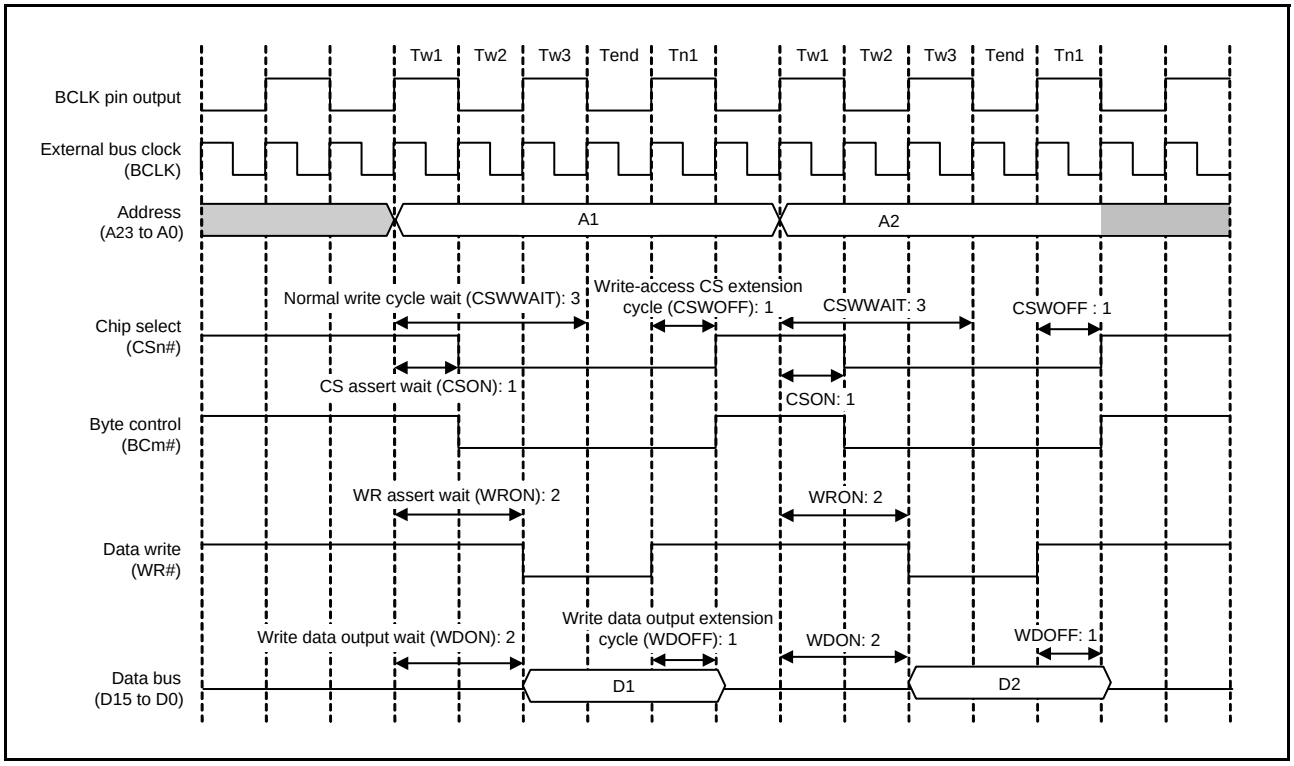


Figure 16.24 Example of Normal-Write Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit) (n = 0 to 7, m = 0, 1)

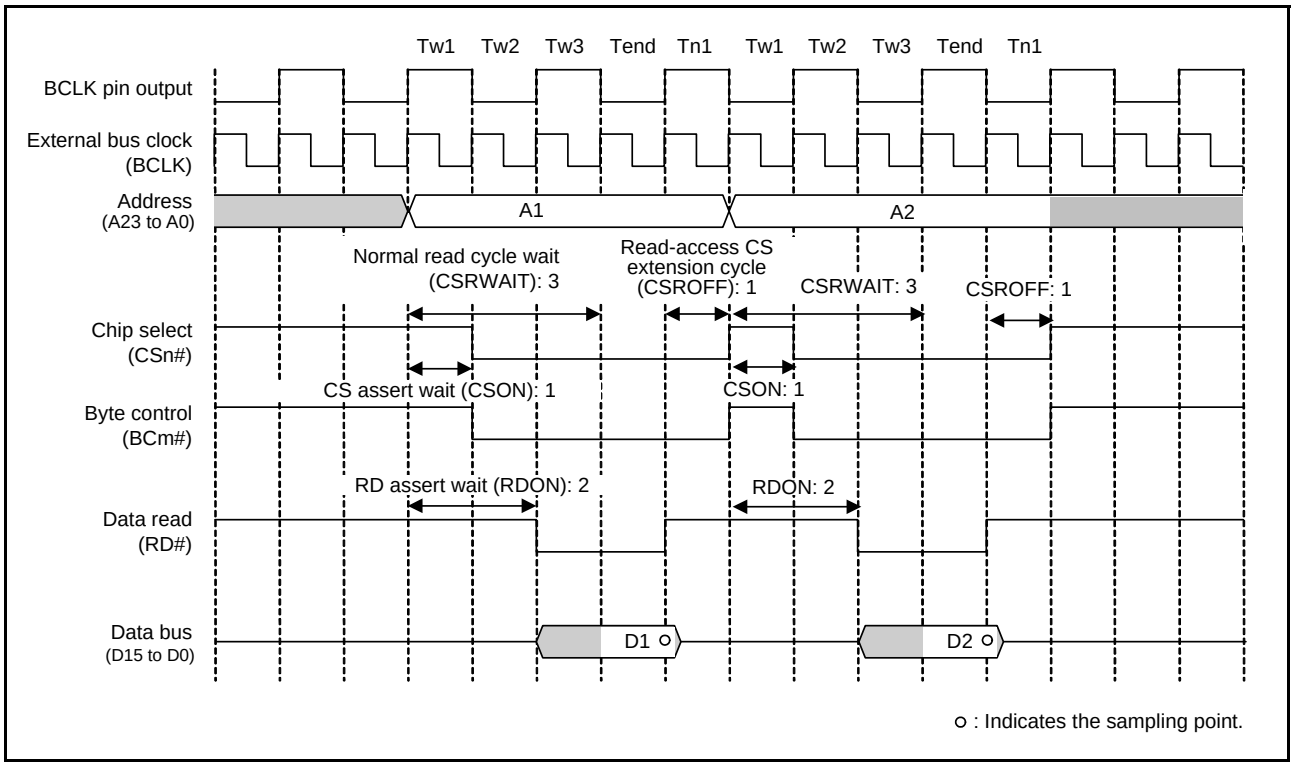


Figure 16.25 Example of Normal-Read Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 7, m = 0, 1)

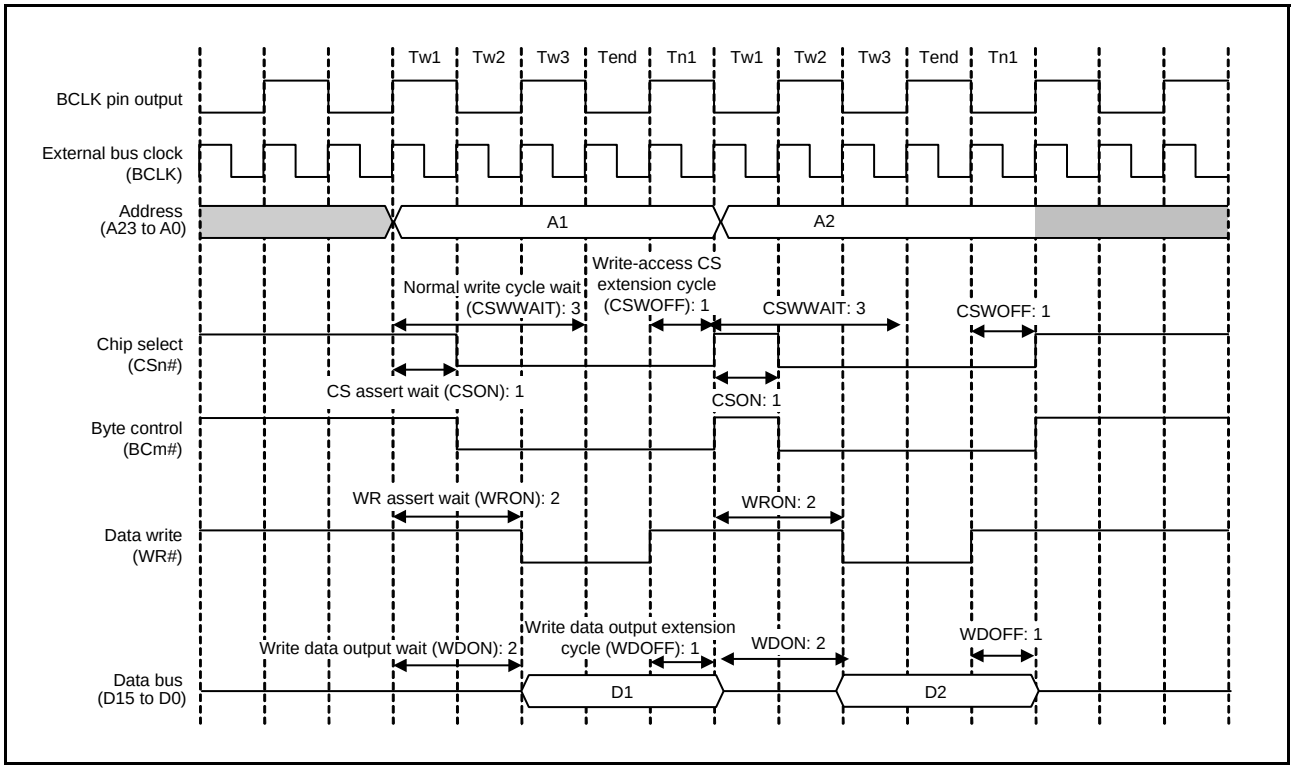


Figure 16.26 Example of Normal-Write Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 7, m = 0, 1)

(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. Refer to Figure 16.8 to Figure 16.11 for the conditions under which page access occurs.
Figure 16.27 and Figure 16.28 show examples of page access operations.

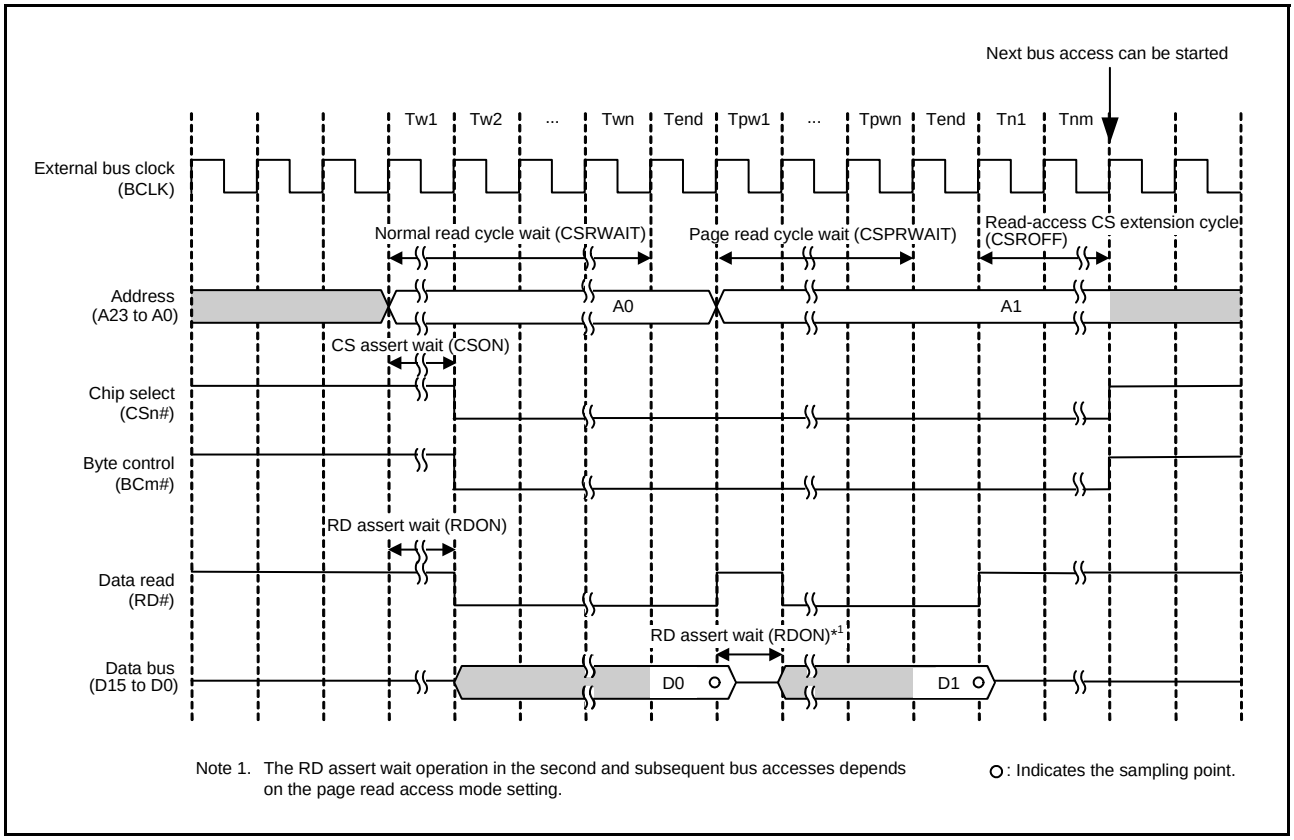


Figure 16.27 Page-Read Access Timing (n = 0 to 7, m = 0, 1)

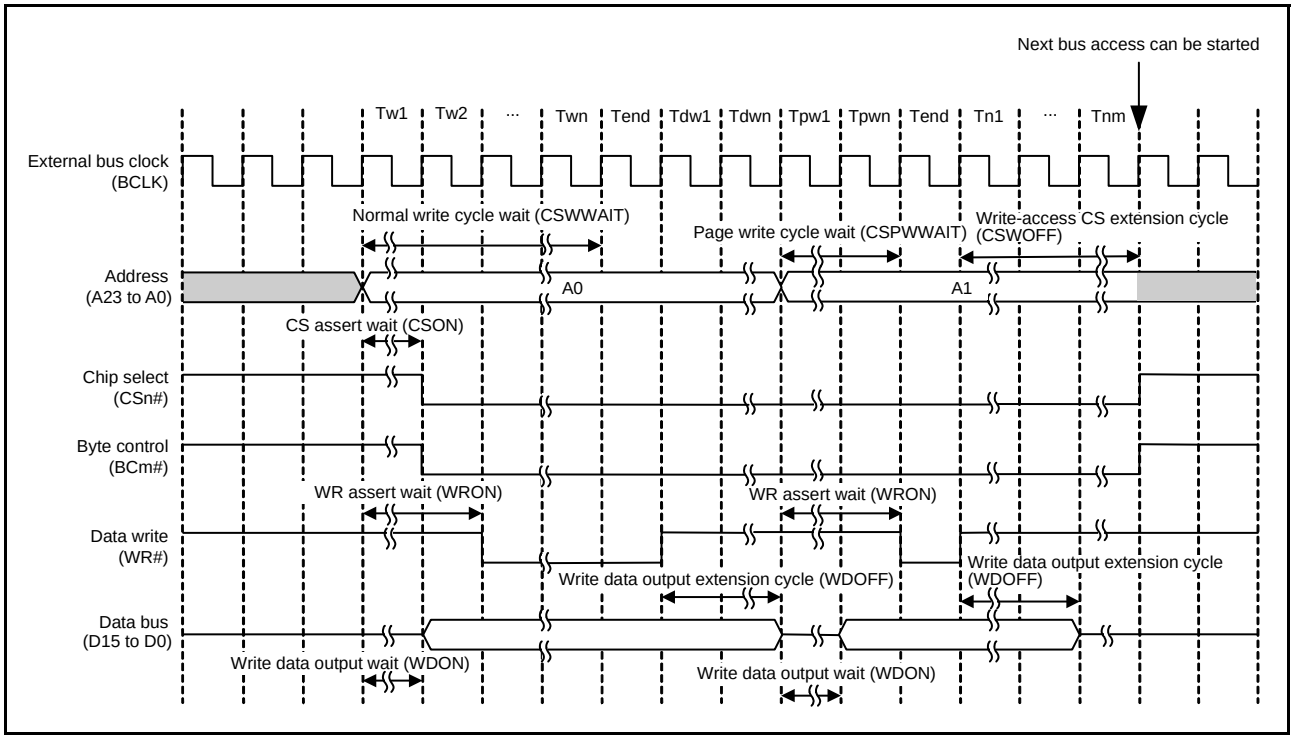


Figure 16.28 Page-Write Access Timing (n = 0 to 7, m = 0, 1)

Figure 16.29 and Figure 16.30 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.

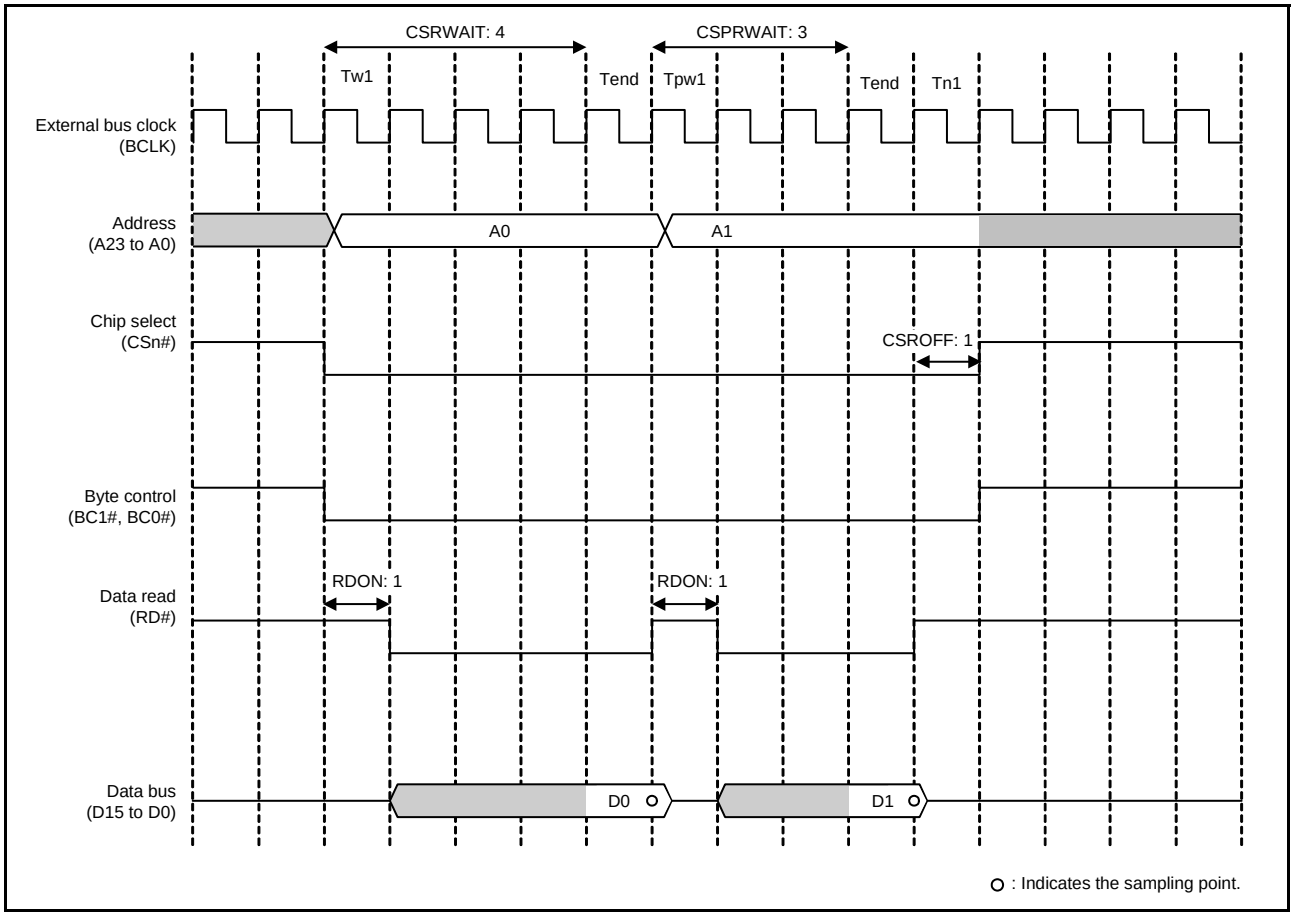


Figure 16.29 Example of Page-Read Access Operation (when 16-Bit Bus Space is Accessed in 32 Bits) (n = 0 to 7)

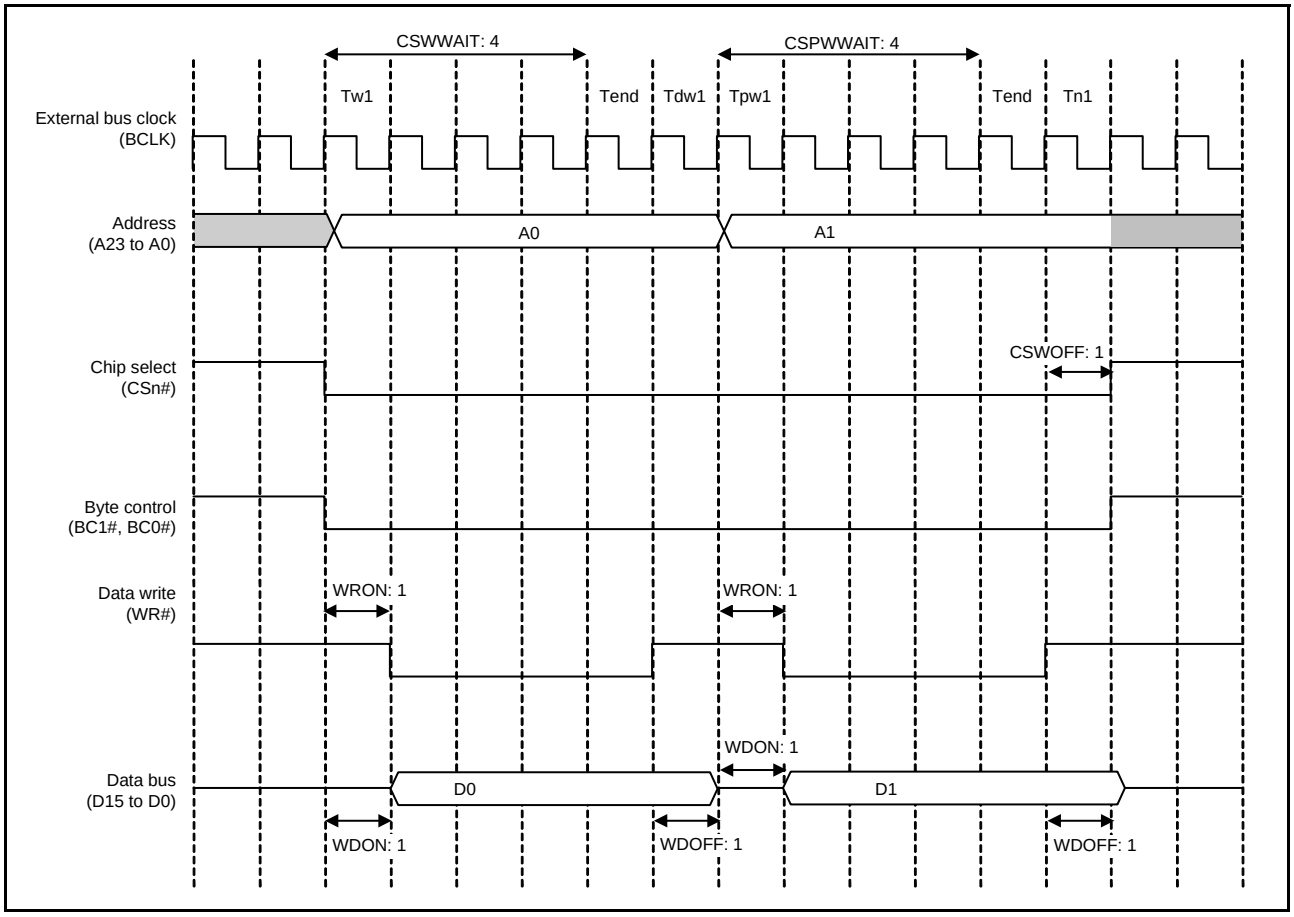


Figure 16.30 Example of Page-Write Access Operation
(when 16-Bit Bus Space is Accessed in 32 Bits, in Single Write Strobe Mode) (n = 0 to 7)

Figure 16.31 and Figure 16.32 show examples of page access operations performed with the 1/2 BCLK selected with the BCLK pin output select bit.

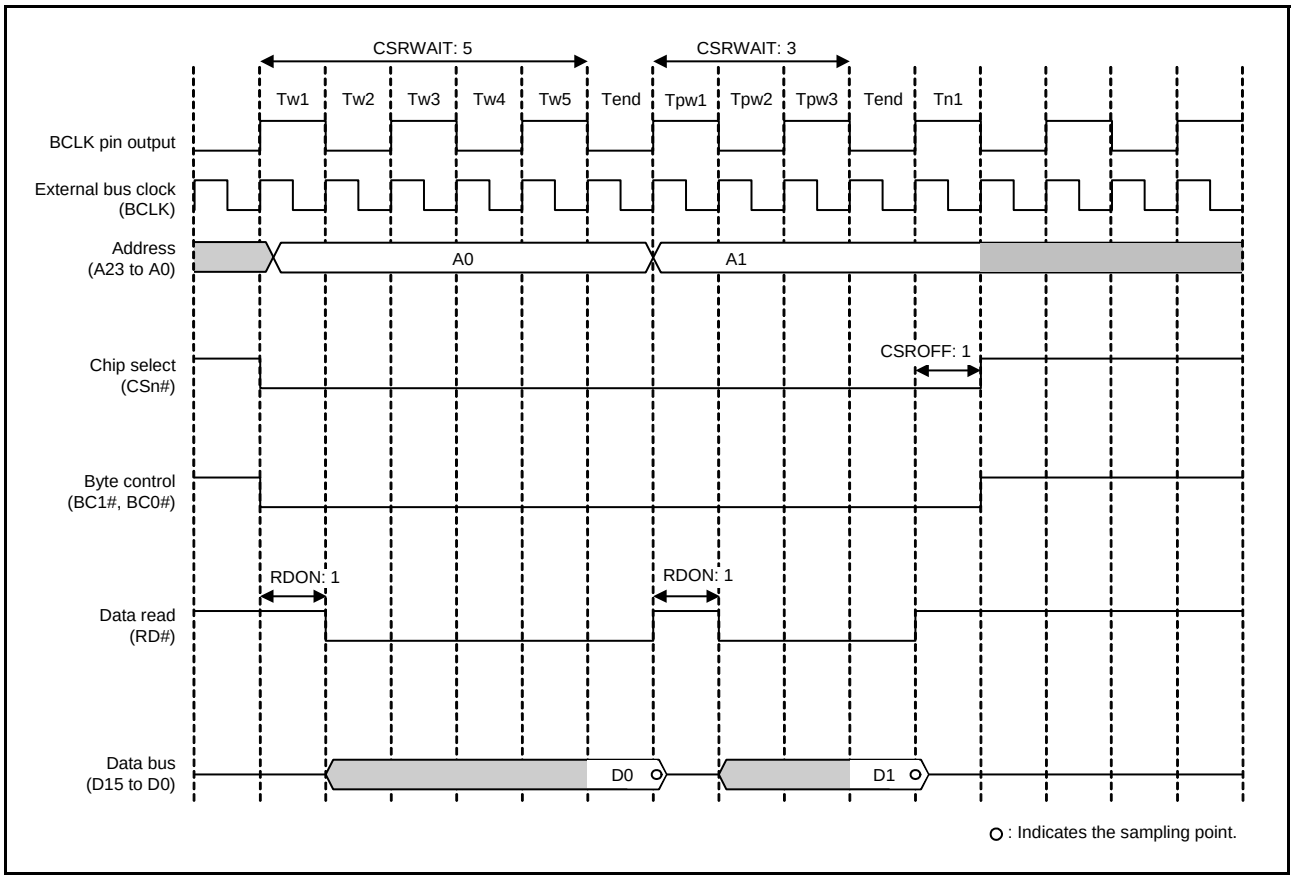


Figure 16.31 Example of Page Read Access Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 7)

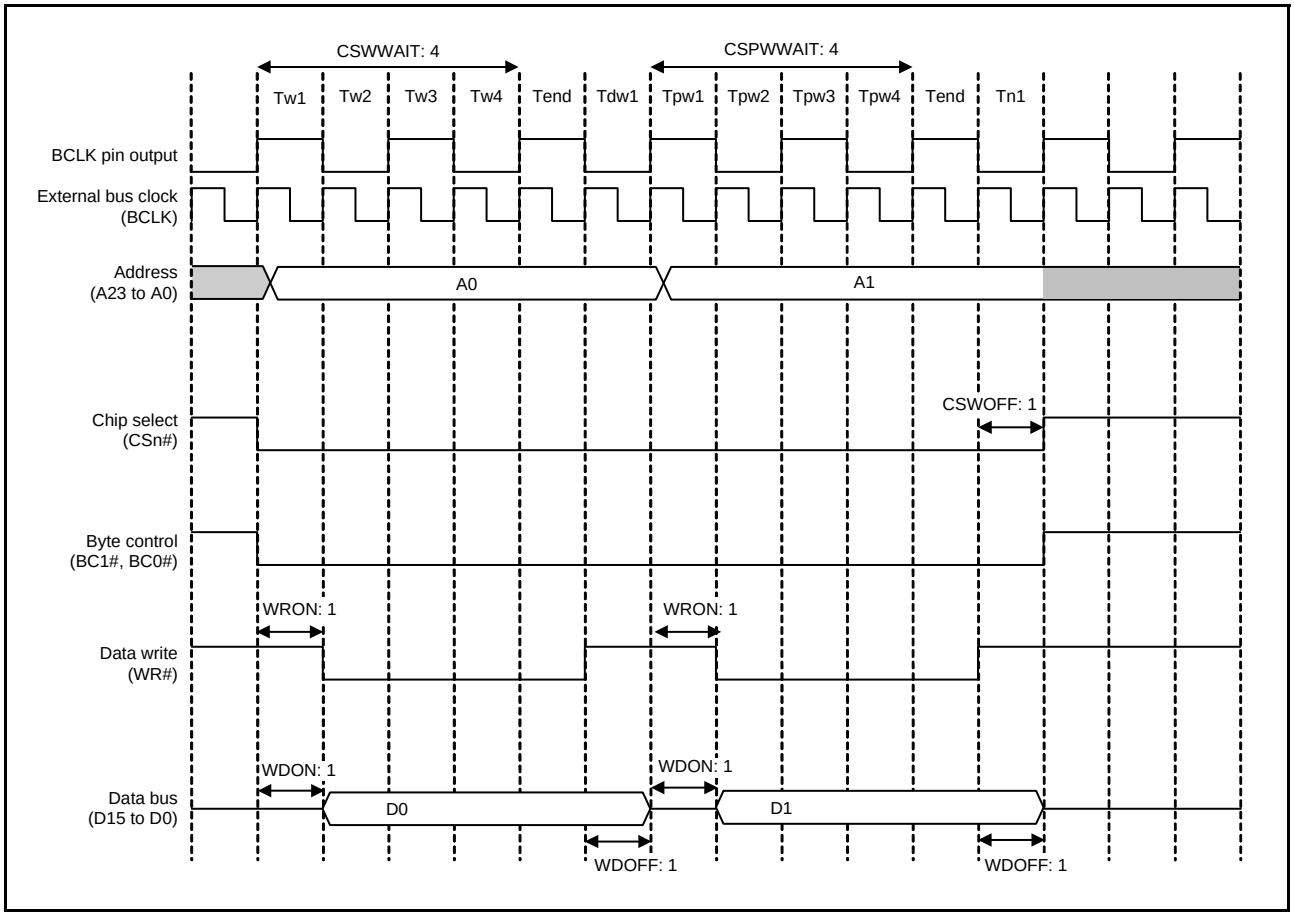


Figure 16.32 Example of Page Write Access Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer, in Single Write Strobe Mode) (n = 0 to 7)

16.5.2 Address/Data Multiplexed Bus

When the address/data multiplexed I/O interface select bit (MPXEN) in CSnCR is set to 1, addresses and data can be multiplexing input/output to/from the D15 to D0 pins in the corresponding area. Using this function enables direct connection of this MCU to peripheral LSIs requiring address/data multiplexing. When 8-bit width is selected with the BSIZE[1:0] bits in CSnCR, D7 to D0 are multiplexed with A7 to A0. When 16-bit width is selected, D15 to D0 are multiplexed with A15 to A0. In the address/data multiplexed I/O space, accesses are controlled with the ALE, RD#, WRn#, and BCn# signals.
Byte strobe mode or single-write strobe mode is selectable in the same way as for a separate bus. However, with regard to the BCn# signals within the address cycle, the byte-control signal is output for the data being read or written.
During the address/data multiplexed I/O space access, after the number of wait cycles specified by the address cycle wait select bits (AWAIT[1:0]) in CSnWCR2 is inserted in the address output cycle, data access is performed.

- Ta1 to Tan (Address Cycle Wait)

The period Ta1 to Tan is valid only when the address/data multiplexed I/O space is specified. This period is made up of the number of clock cycles between the start of external bus access and one cycle before the address latch (ALE) signal is negated. The number of cycles are selectable within the range from zero to three. Addresses are output until the next cycle of ALE signal negation (address cycle). The timing of ALE signal is the same as that of CS# assertion. After the address cycle, a data cycle is started. CSnWCR1 and CSnWCR2 should be set so that an address cycle and a data cycle do not overlap.

Page access to the address/data multiplexed I/O space is invalid. When the PRENB or PWENB bit in CSnMOD is set to 1 to enable page-read or page-write access, these settings are ignored and normal read or write operation is performed.

Figure 16.33 to Figure 16.35 show examples of operations with the address/data multiplexed I/O interface.

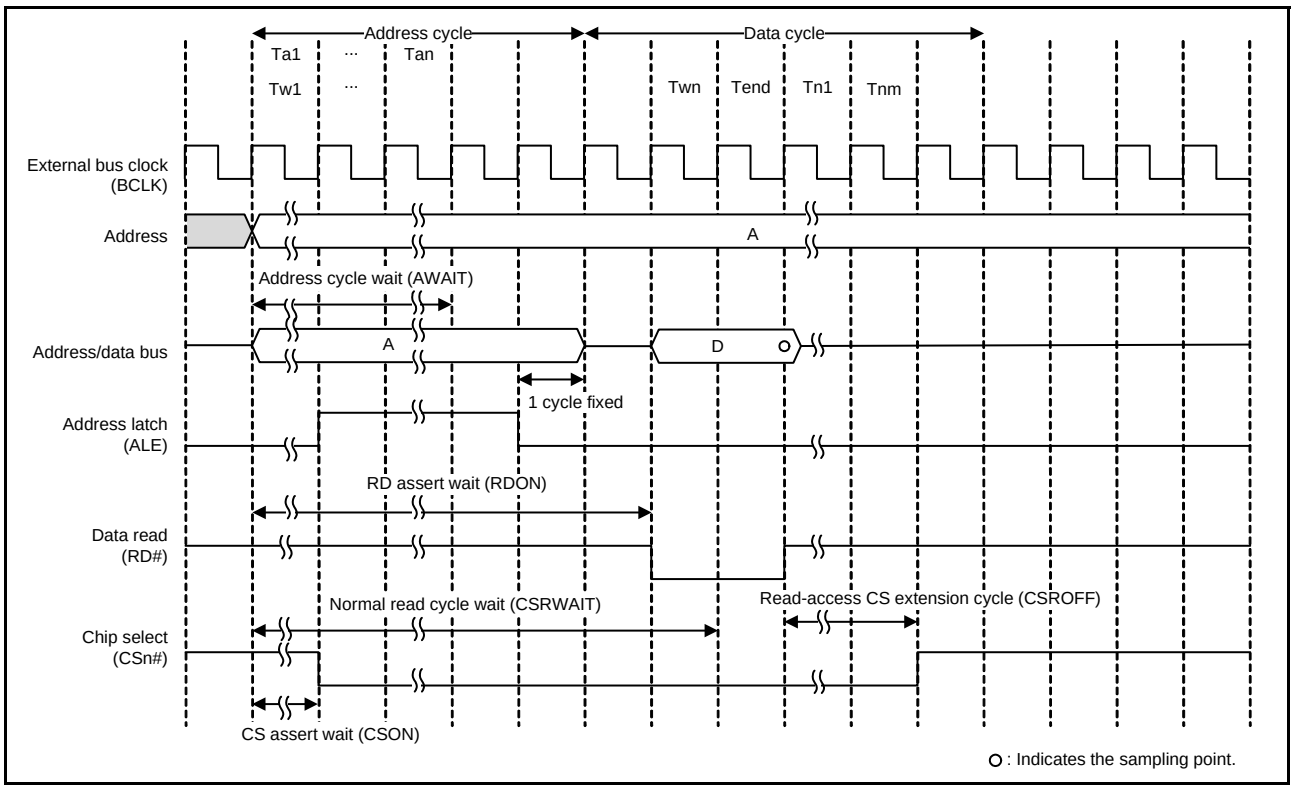


Figure 16.33 Example of Read Access Operation with Address/Data Multiplexed I/O Interface (n = 0 to 7)

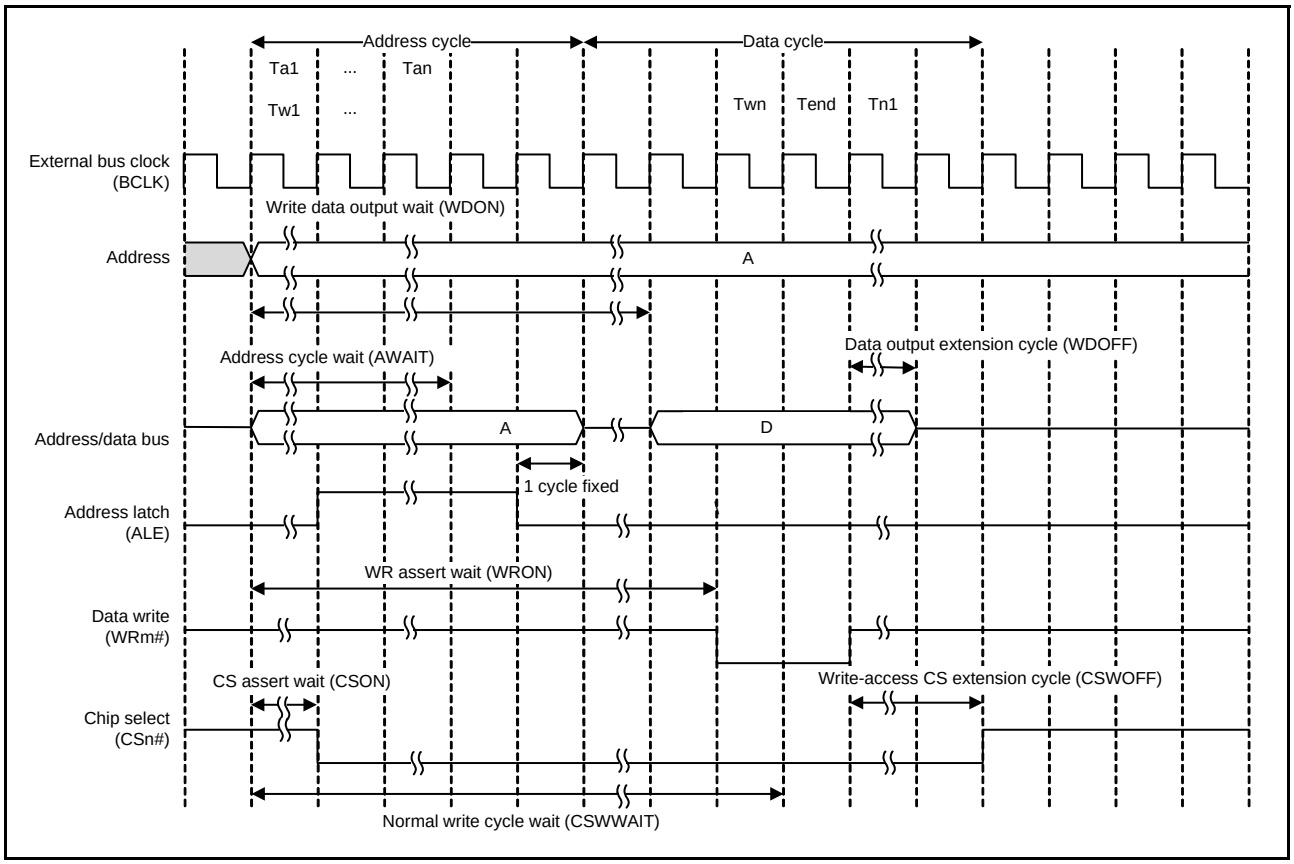


Figure 16.34 Example of Write Access Operation with Address/Data Multiplexed I/O Interface (n = 0 to 7, m = 0, 1)

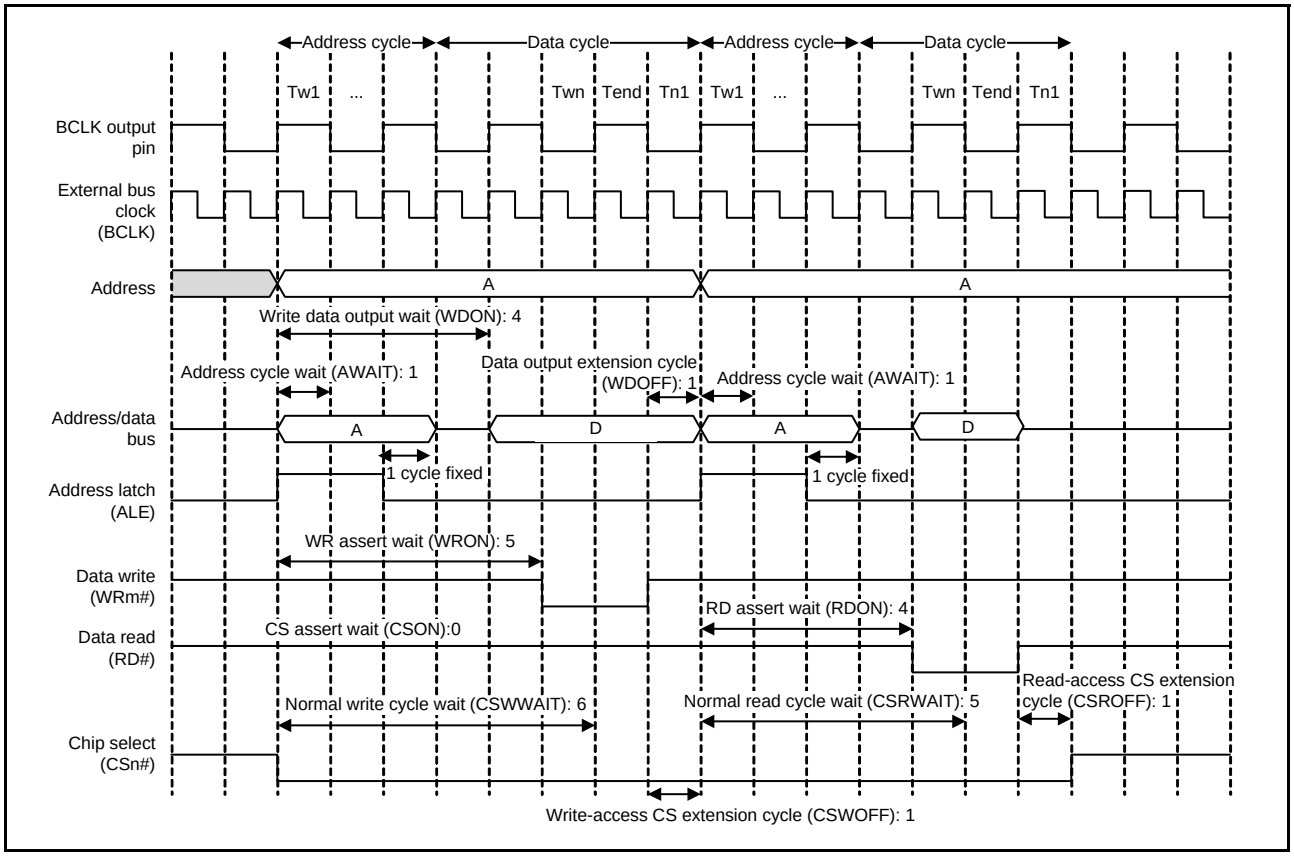


Figure 16.35 Example of Bus Timing with Address/Data Multiplexed I/O Interface (n = 0 to 7, m = 0, 1)

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 16.36 and Figure 16.37 show examples of external wait insertion timing with the separate bus interface.

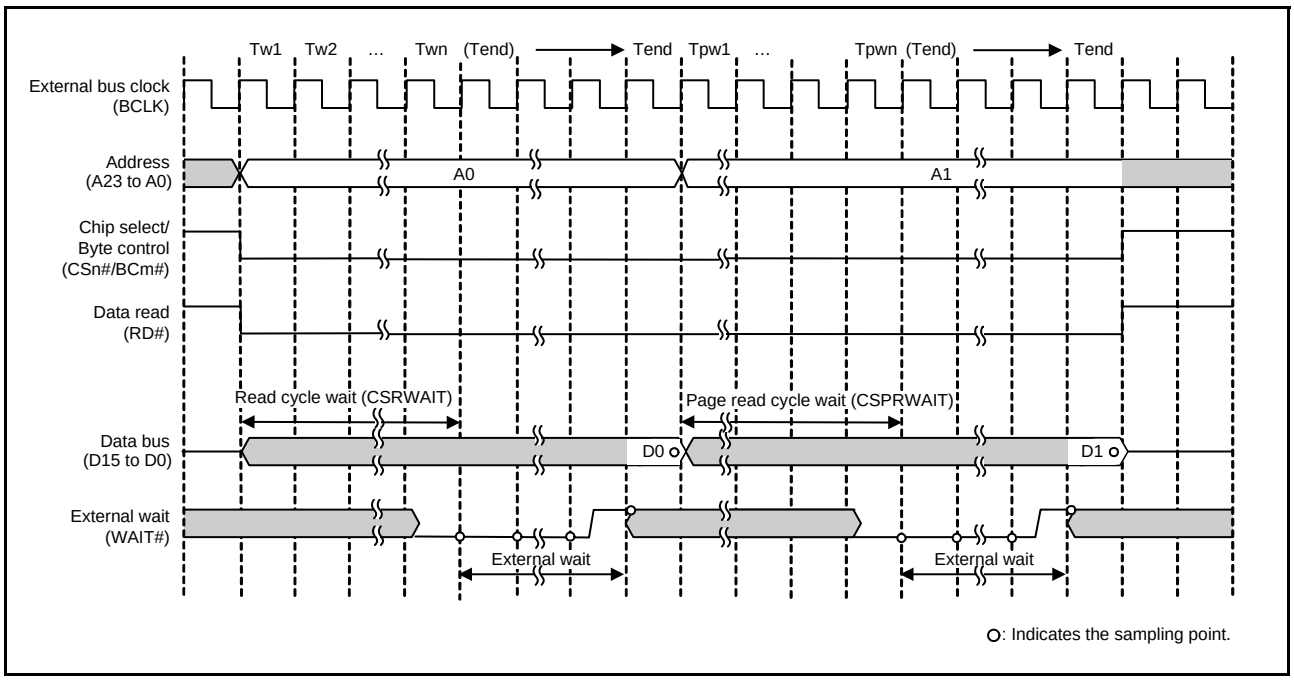


Figure 16.36 Example of External Wait Timing (Page-Read Access to 16-Bit Bus Space) (n = 0 to 7, m = 0, 1)

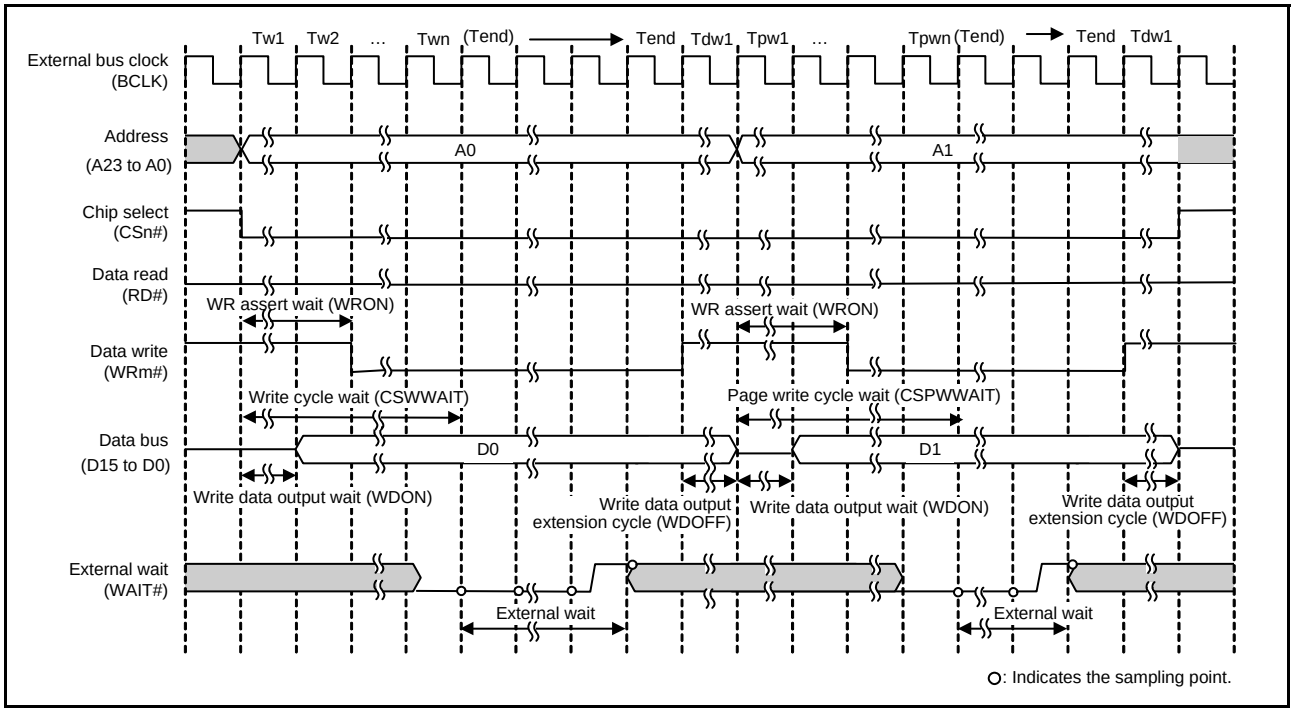


Figure 16.37 Example of External Wait Timing (Page-Write Access to 16-Bit Bus Space, in Byte Strobe Mode) (n = 0 to 7, m = 0, 1)

(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 16.38 shows an example of external wait insertion timing with the address/data multiplexed I/O interface.

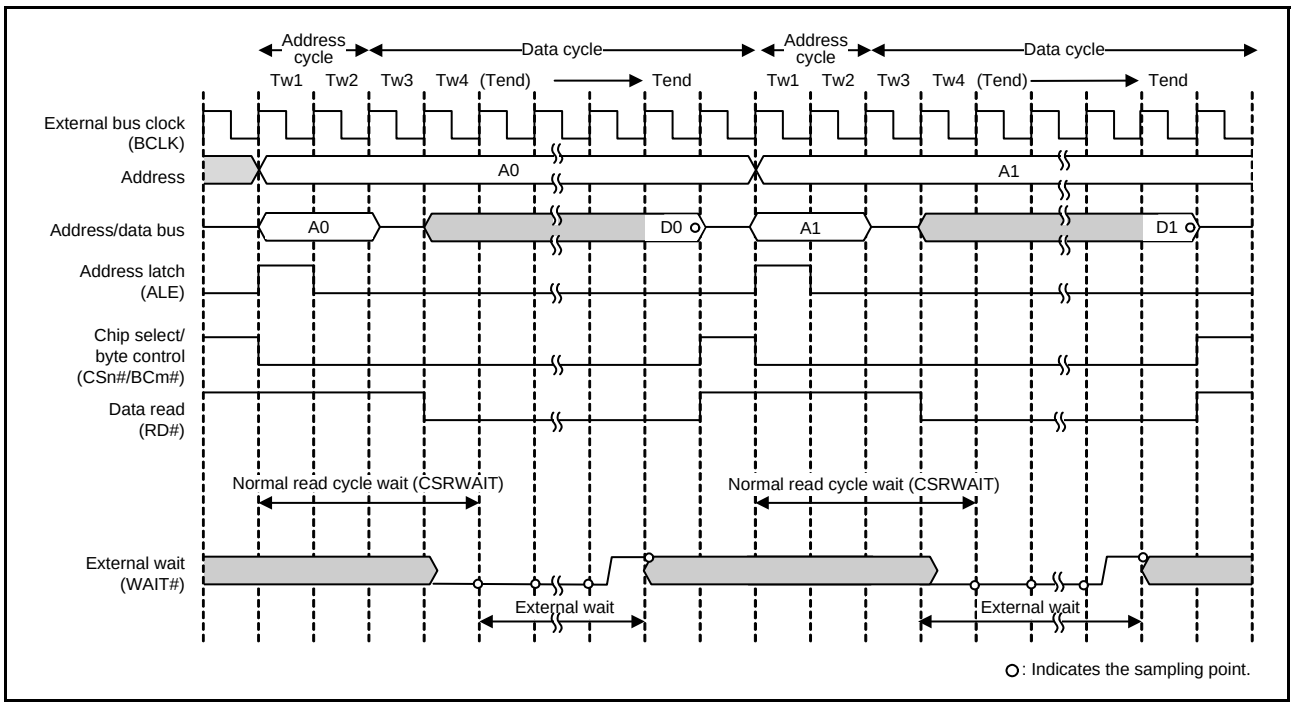


Figure 16.38 Example of External Wait Insertion Timing with Address/Data Multiplexed I/O Interface (n = 0 to 7, m = 0, 1)

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 16.41).
Similarly, during normal accesses with page access enabled, recovery cycles are not inserted between bus access cycles but inserted only after the last bus access cycle of the transfer.
With the address/data multiplexed I/O interface, when the recovery cycle insertion condition is satisfied, recovery cycles are inserted between bus access cycles regardless of the page access enable setting.

Figure 16.39 to Figure 16.41 show examples of recovery cycle insertion with the separate bus interface.

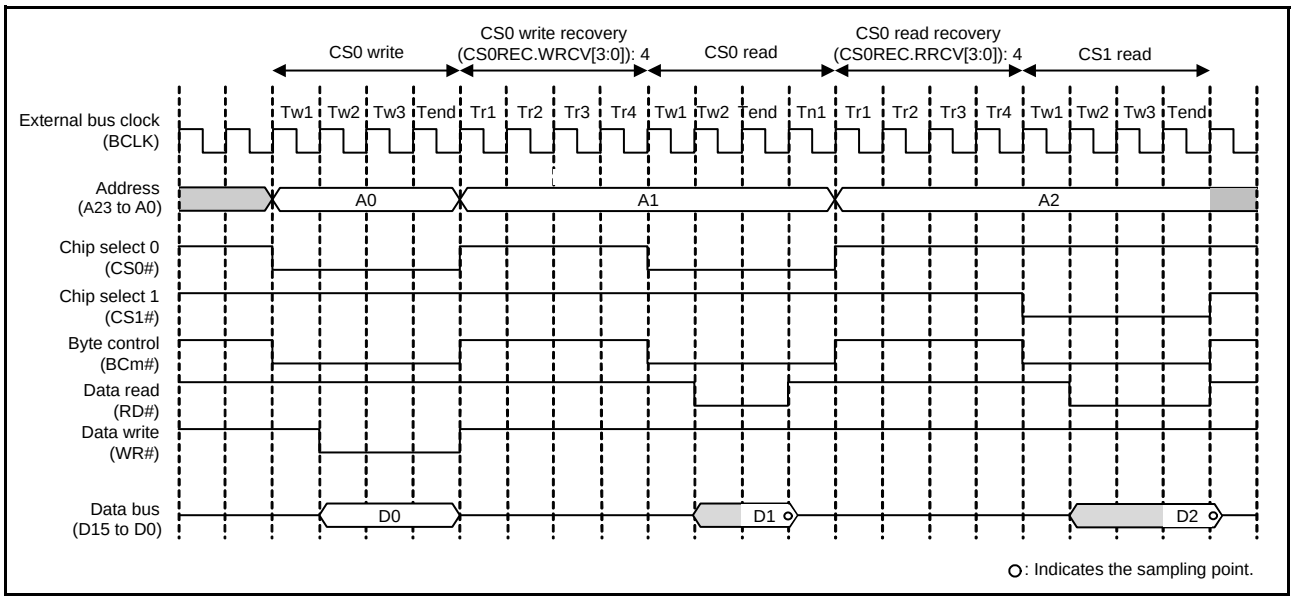


Figure 16.39 Example of Recovery Cycle Insertion with Separate Bus Interface (m = 0, 1)

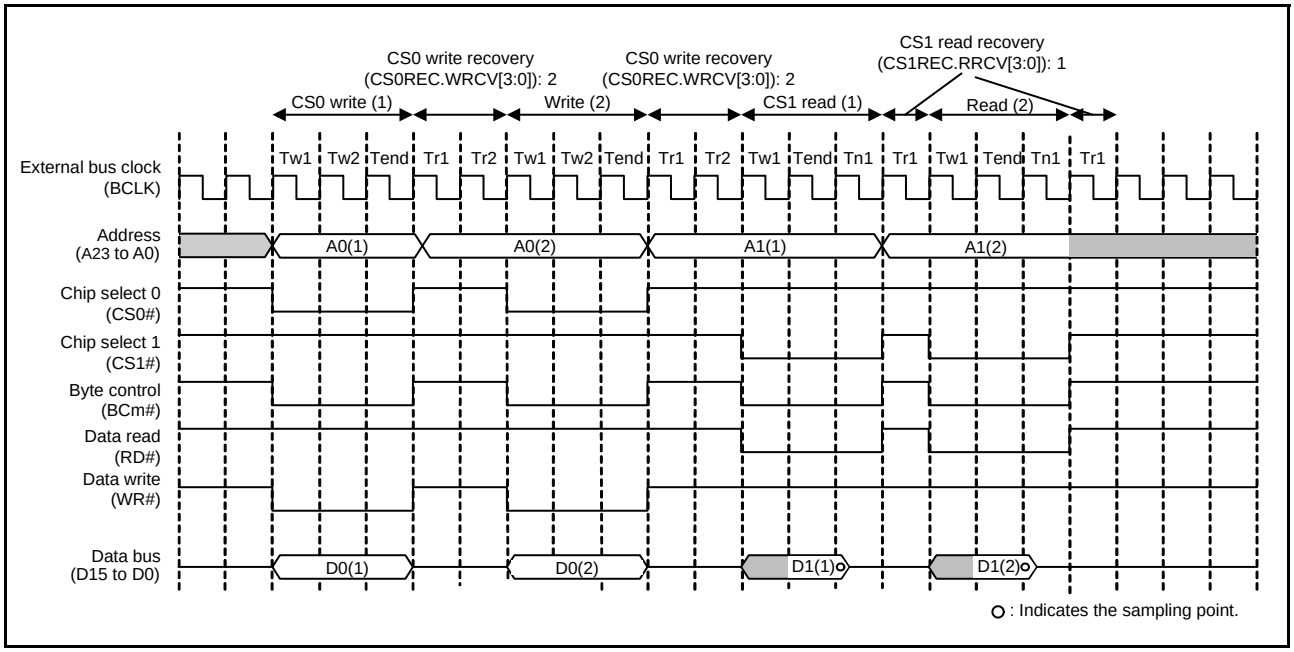


Figure 16.40 Example of Recovery Cycle Insertion When a Bus Access is Split (with Separate Bus Interface, Normal Access) (m = 0, 1)

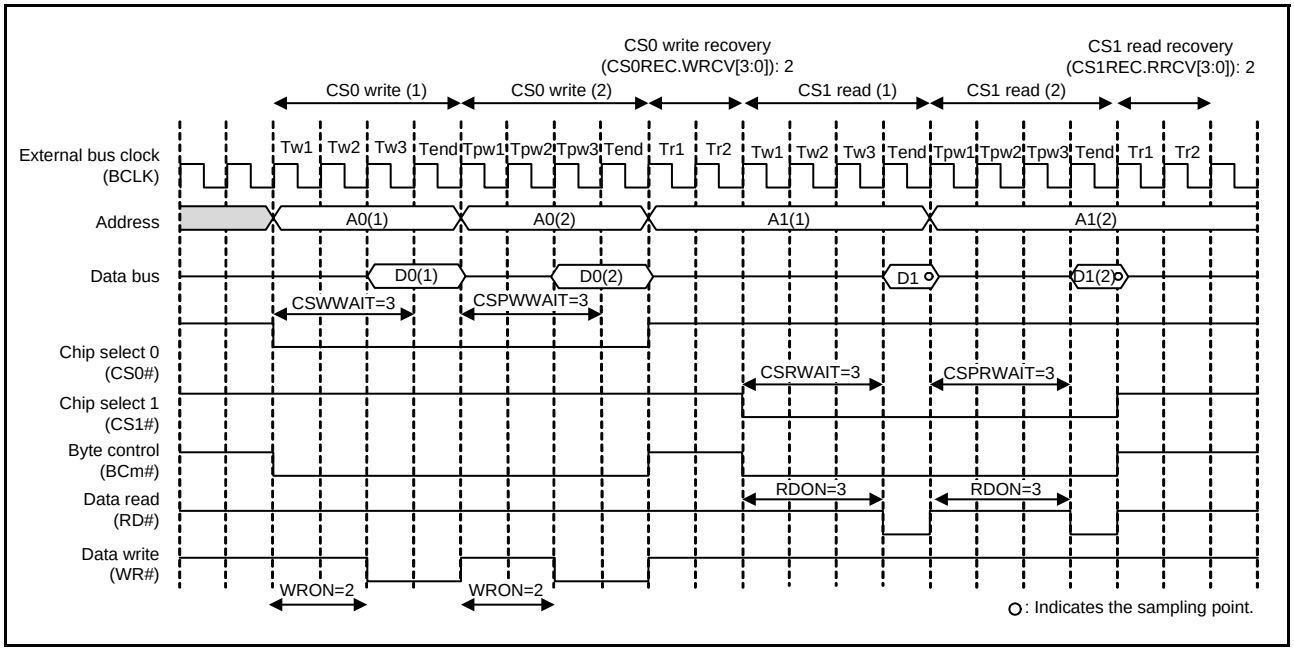


Figure 16.41 Example of Recovery Cycle Insertion When a Bus Access is Split (with Separate Bus Interface, Page Access) (m = 0, 1)

Figure 16.42 shows examples of operations when the BCLK pin output selection bits are set for frequency-division of BCLK by 2.

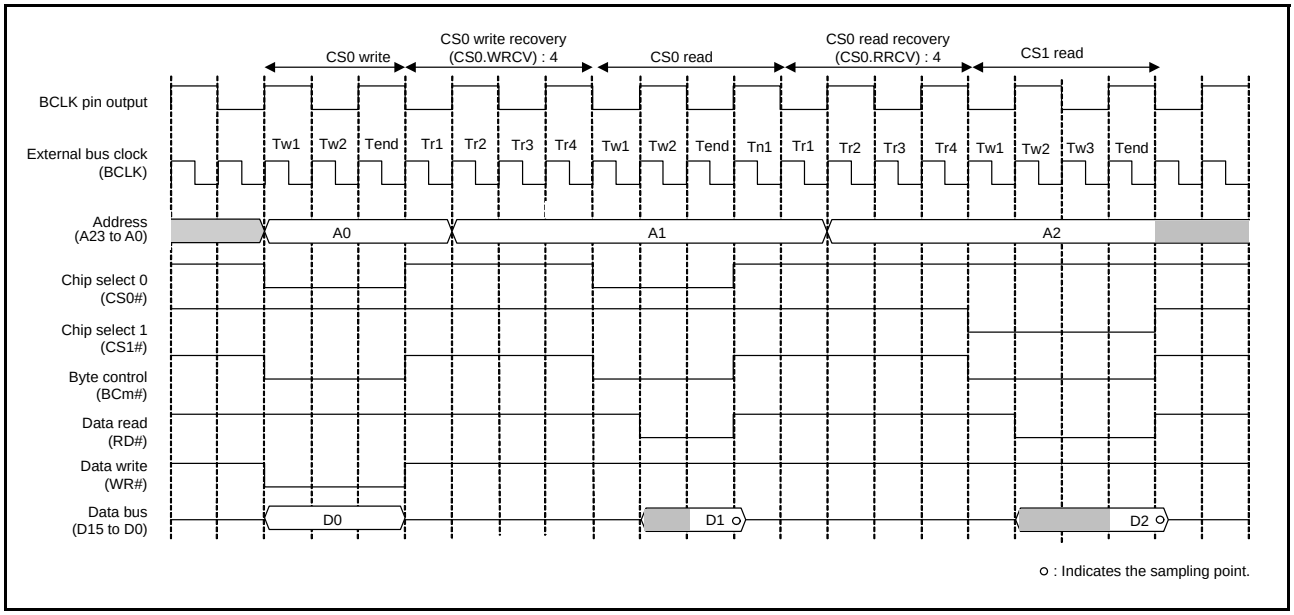


Figure 16.42 Example of Operation for Recovery Cycles when the BCLK Pin Output Selection Bits are Set for Frequency-Division of BCLK by 2 (For the Case of Normal Access through a Separate Bus Interface) (m = 0, 1)

With the address/data multiplexed I/O interface, recovery cycles are inserted in the same way as that with the separate bus interface. Figure 16.43 and Figure 16.44 show examples of recovery cycle insertion with the address/data multiplexed I/O interface.

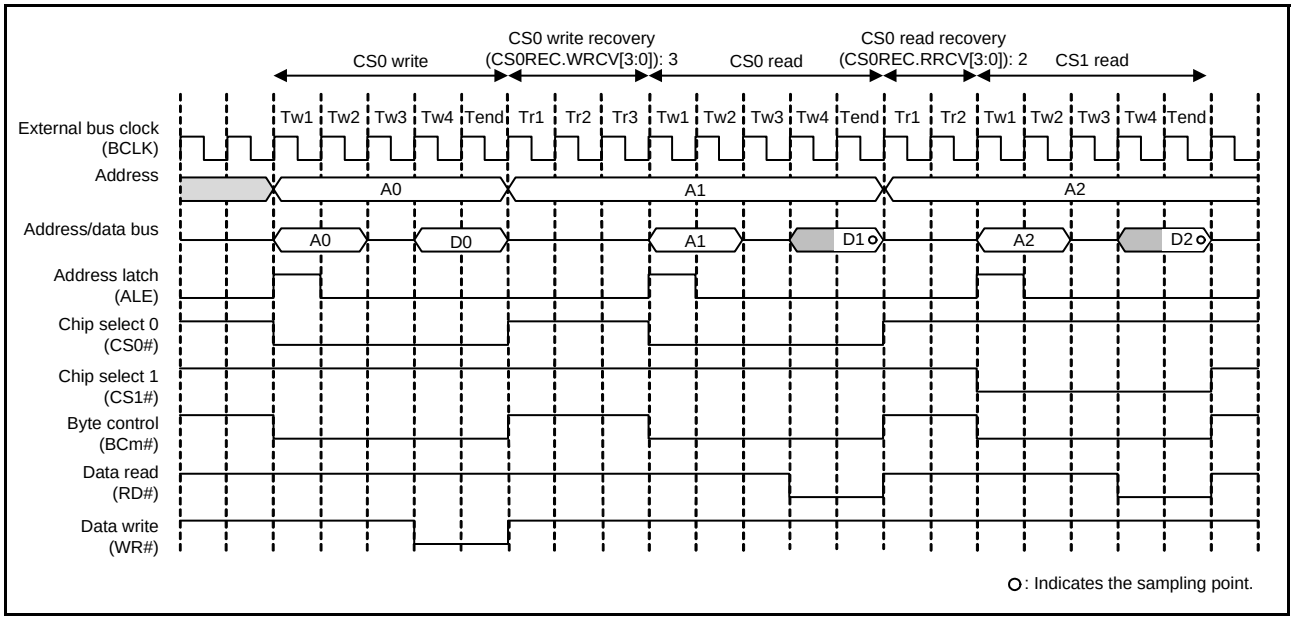


Figure 16.43 Example of Recovery Cycle Insertion with Address/Data Multiplexed I/O Interface (m = 0, 1)

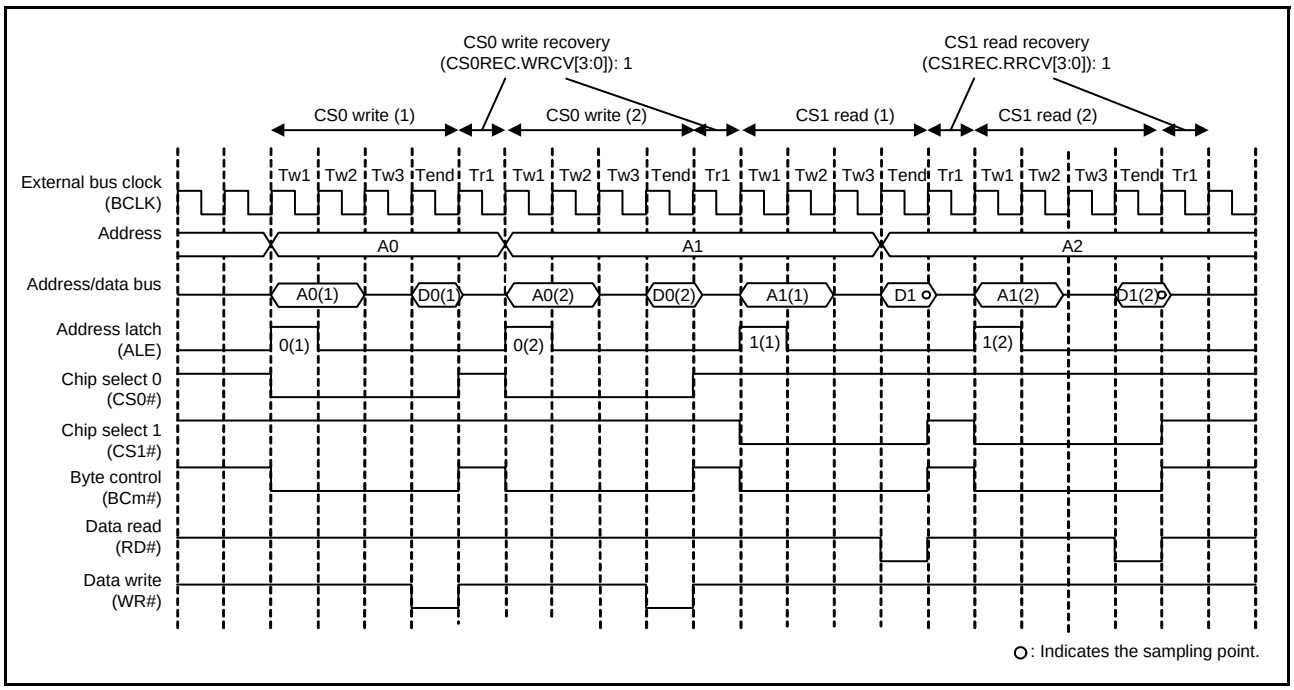


Figure 16.44 Example of Recovery Cycle Insertion When a Bus Access is Split with Address/Data Multiplexed I/O Interface (m = 0, 1)

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 D15 to D0 are in the high-impedance state.

16.5.6 Write Buffer Function (External Bus)

The internal main bus is released by writing data to the write buffer before the write access is completed, which allows the next round of bus access to start. However, if the following round of bus access is to an external address space or to a register of the external bus controller, it is suspended until the external bus operations already in progress are completed. Figure 16.45 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.

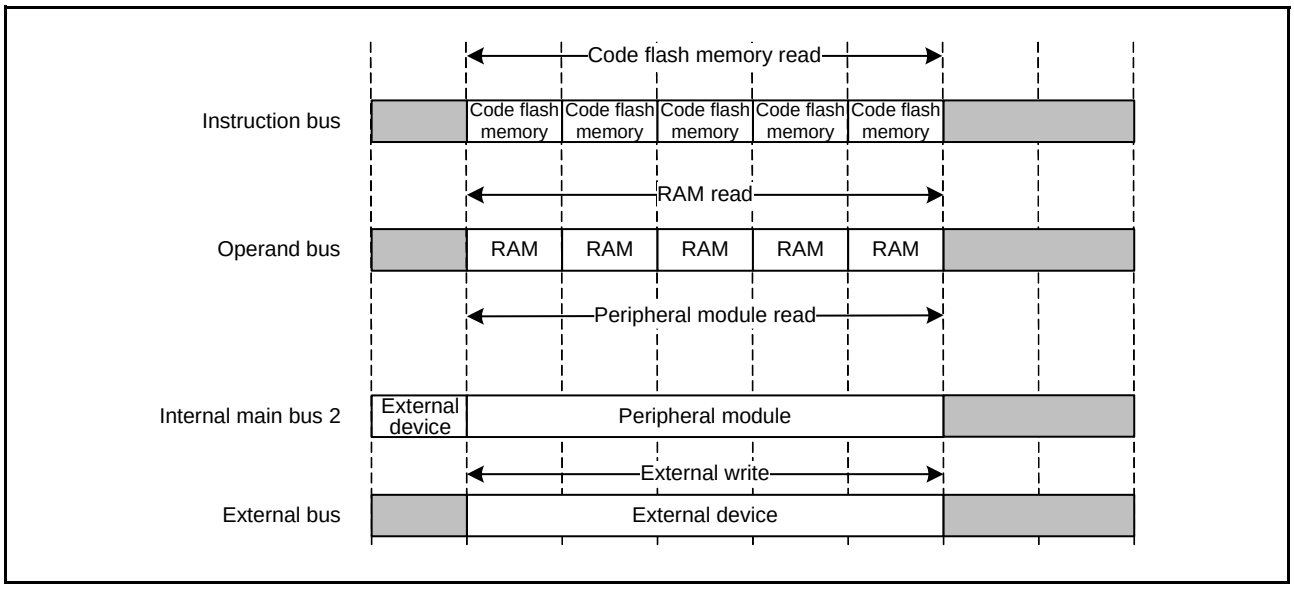


Figure 16.45 Example of Operation when the Write-Buffer Function is in Use

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 16.11.
  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 16.11 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.

Table 16.12 Limitations at the Time of Normal 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 16.13 and Table 16.14 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 16.13 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 16.14 Limitations on EXDMAC Single Address Transfer Mode (EDMOMD.DACKW = 1)

Limitations at the Time of Normal Access		Limitations at the Time of Page Access	
Reading	**Writing**	**Reading**	**Writing**
CSON[2:0] ≤ CSRWAIT RDON[2:0] < CSRWAIT CSON[2:0] ≤ RDON 1 ≤ RDON	CSON[2:0] ≤ CSWWAIT WRON[2:0] ≤ CSWWAIT WDON[2:0] ≤ CSWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON 1 ≤ WRON 1 ≤ WDOFF	CSON[2:0] ≤ CSPRWAIT RDON[2:0] < CSPRWAIT CSON[2:0] ≤ RDON 1 ≤ RDON	CSON[2:0] ≤ CSPWWAIT WRON[2:0] ≤ CSPWWAIT WDON[2:0] ≤ CSPWWAIT WDOFF[2:0] ≤ CSWOFF WDON[2:0] ≤ WRON CSON[2:0] ≤ WRON 1 ≤ WRON 1 ≤ WDOFF

(6) Prohibition of Access that Spans Areas of Address Space

Single access that spans several areas of the address space is prohibited, and operation of such an access is not guaranteed. Setting must be made so that two areas are not accessed at the same time by a single word or longword access.

Single access that spans two areas of the address space is also prohibited in the EXDMAC block transfer in single address mode or cluster transfer, and operation of such an access is not guaranteed. Setting must be made so that two areas are not accessed at the same time by a single access in the EXDMAC block transfer in single address mode or cluster transfer.

(7) Restrictions on RMPA and String-Manipulation Instructions

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

(8) Restriction on Instruction Code

When the endian setting for each area is different from that for the chip, no instruction code can be allocated in the area. The instruction code should be allocated to the external space whose endian setting is the same as that for the chip.

16.6 SDRAM Area Controller Operation

The following sections describe how the SDRAM area controller (SDRAMC) is enabled and the SDRAM bus width is set, which is followed by the description of SDRAMC operations including read, write, auto-refresh, self-refresh, initialization sequence, and mode register setting.

16.6.1 Enabling/Disabling SDRAM Access and Setting SDRAM Bus Width

SDRAM access can be enabled or disabled using the SDC control register (SDCCR). The SDRAM bus width can also be set using SDCCR.
Even when the operation of the SDRAM address space is disabled, refresh operation is available as long as self-refresh or auto-refresh operation is enabled.

16.6.2 No Access State

When no external address space is accessed, SDCS#, WEn#, RAS#, and CAS# signals are high.

16.6.3 Insertion of Recovery Cycles

When access to the SDRAM area follows access to the CS area, data recovery cycles are inserted for the CS area controller (CSC). If the number of recovery cycles for the CSC is 0, the ACT command for the next SDRAM access is issued immediately after negation of CSn# signal at the earliest. If the number of recovery cycles are not 0, the ACT command is issued two cycles after the specified recovery cycle period elapsed after negation of CSn# signal at the earliest. Since no data conflicts can occur during access to the SDRAM area, there is no need to set data recovery cycles for the SDRAM (fixed to zero cycle).

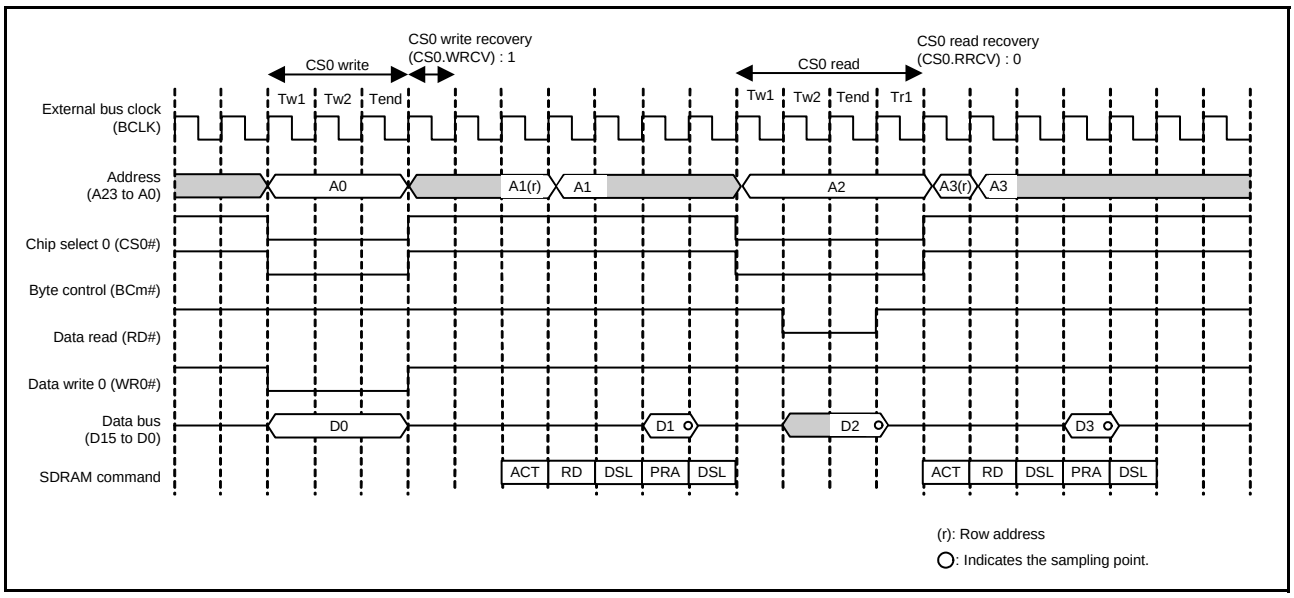


Figure 16.46 Example of Recovery Timing (for SDRAM Access) (m = 0, 1)

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 16.15 lists the commands issued by the SDRAMC.

Table 16.15 List of SDRAMC Commands

							CKE	
Name	**Abbreviation**	**Command**	**SDCS#**	**RAS#**	**CAS#**	**WE#**	**n-1**	**n**
DESL	DSL	Device deselect	H	x	x	x	H	x
ACTV	ACT	Bank active	L	L	H	H	H	x
READ	RD	Read	L	H	L	H	H	x
WRIT	WRI	Write	L	H	L	L	H	x
PALL	PRA	All bank precharge	L	L	H	L	H	x
REF	RFA	Auto-refresh	L	L	L	H	H	H
MRS	MRS	Mode register set	L	L	L	L	H	x
SELF	RFS	Self-refresh entry	L	L	L	H	H	L
SELFX	RFX	Self-refresh end	H	x	x	x	L	H

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 16.16.

Table 16.16 Conditions for Register Modification

Function/Operation	**Registers**	**Conditions**
Self-refresh	SDSELF*1	• SDRAM access is disabled. (SDCCR.EXENB = 0*2) • Auto-refresh operation is enabled. (SDRFEN.RFEN = 1)
Auto-refresh	SDRFCR	Self-refresh operation is disabled. (SDSELF.SFEN = 0)
	SDRFEN	Self-refresh operation is disabled. (SDSELF.SFEN = 0)
Initialization sequence	SDIR*1	The SDICR has not been set yet, and the same conditions as SDICR modification should be satisfied.
	SDICR*1	• SDRAM access is disabled. (SDCCR.EXENB = 0*2) • Auto-refresh operation is disabled. (SDRFEN.RFEN = 0) • Self-refresh operation is disabled. (SDSELF.SFEN = 0)
Address register	SDADR	• SDRAM access is disabled. (SDCCR.EXENB = 0*2) • Auto-refresh operation is disabled. (SDRFEN.RFEN = 0) • Self-refresh operation is disabled. (SDSELF.SFEN = 0)
Timing register	SDTR	• During self-refresh operation (SDSELF.SFEN = 1) or • SDRAM access is disabled. (SDCCR.EXENB = 0*2) • Auto-refresh operation is disabled. (SDRFEN.RFEN = 0) • Self-refresh operation is disabled. (SDSELF.SFEN = 0)
Mode register	SDMOD*1	• SDRAM access is disabled. (SDCCR.EXENB = 0*2) • Self-refresh operation is disabled. (SDSELF.SFEN = 0)
Access mode register	SDAMOD	• SDRAM access is disabled. (SDCCR.EXENB = 0*2) • Auto-refresh operation is disabled. (SDRFEN.RFEN = 0) • Self-refresh operation is disabled. (SDSELF.SFEN = 0)

Note 1. Before modification, confirm that all the status bits in SDSR are 0.
Note 2. After writing 0 to the EXENB bit, confirm that the EXENB bit is set to 0.

16.6.7 Self-Refresh

Transition to or recovery from self-refresh mode can be controlled using the SDRAM self-refresh control register (SDSELF).

Immediately before transition to self-refresh mode, auto-refresh operation is performed. In self-refresh mode, the CKE signal is low. Immediately after recovery from self-refresh mode, the auto-refresh cycle is started.

Figure 16.47 and Figure 16.48 show timing examples of transition to self-refresh mode and recovery from self-refresh mode, respectively.

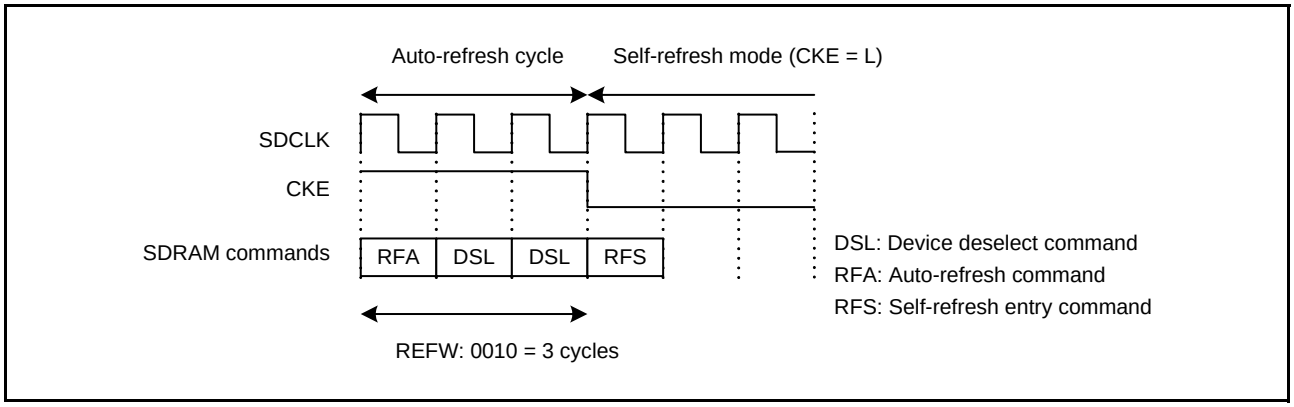


Figure 16.47 Timing Example of Transition to Self-Refresh Mode (when SDRFCR.REFW[3:0] = 0010b: 3 Cycles)

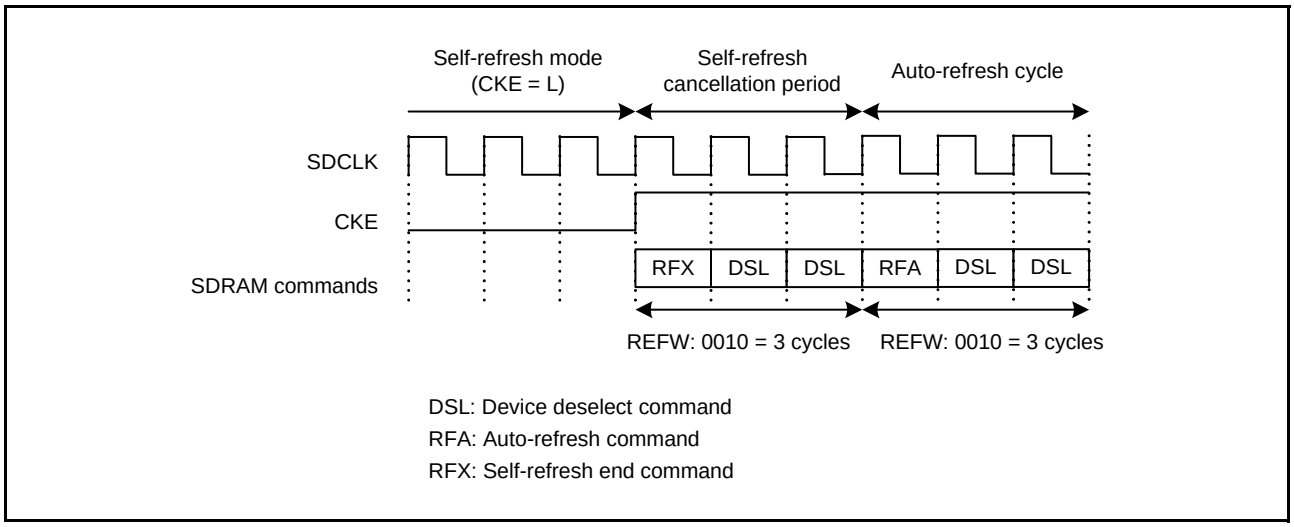


Figure 16.48 Timing Example of Recovery from Self-Refresh Mode

(1) Self-Refresh in All-Module-Clock Stop Mode

When causing transition to self-refresh mode in all-module-clock stop mode, first cause transition to self-refresh mode according to the procedure shown in section 16.6.12.2, Procedure for Transition to and Recovery from Self-Refresh Mode, and then make settings to cause transition to all-module-clock stop mode.

After canceling all-module-clock stop mode, follow the procedure shown in section 16.6.12.2, Procedure for Transition to and Recovery from Self-Refresh Mode.

For details of transition and cancellation of all-module-clock stop mode, refer to section 11, Low Power Consumption.

(2) Self-Refresh in Software Standby Mode

When causing transition to self-refresh mode in software standby mode, first cause transition to self-refresh mode

according to the procedure shown in section 16.6.12.2, Procedure for Transition to and Recovery from Self-Refresh Mode, and then make settings to cause transition to software standby mode. In software standby mode, set the output port enable bit (OPE) in the standby control register (SBYCR) to 1 to hold the output state of the address bus and bus control signals.
After canceling software standby mode, follow the procedure shown in section 16.6.12.2, Procedure for Transition to and Recovery from Self-Refresh Mode.
For details of transition and cancellation of software standby mode, refer to section 11, Low Power Consumption.

(3) Self-Refresh in Deep Software Standby Mode

Transition to deep software standby mode is performed via software standby mode. On transition to deep software standby mode from software standby mode, the state of pins remains unchanged. Therefore, transition to self-refresh mode in deep software standby mode can be made according to the same procedure as that in software standby mode. In deep software standby mode, however, additional setting is necessary to cause transition to self-refresh mode; it is necessary to set the I/O port keep bit (IOKEEP) in the deep software standby control register (DPSBYCR) to 1.
Since the SDRAMC is internally reset by an internal reset signal when deep software standby mode is canceled, the SDRAM control registers need to be set again. After canceling software standby mode, follow the procedure shown below to cancel self-refresh mode. Figure 16.49 shows self-refresh timing in deep software standby mode.
For details of transition and cancellation of deep software standby mode, refer to section 11, Low Power Consumption.

1. In deep software standby mode, the CKE signal output remains low according to the IOKEEP setting in DPSBYCR.
2. Start clock supply to SDRAMC.
3. Set the SDRAM control registers (SDCMOD, SDAMOD, SDADR, and SDTR) again, which have been initialized by an internal reset upon transition to deep software standby mode, and then enable auto-refresh operation (RFEN bit in SDRFEN = 1).
4. Check that all the status bits in SDSR are set to 0 and set the SFEN bit in SDSELF to 1 to set self-refresh mode again.
5. Modify port settings for the SDRAM interface according to the procedure below.
   (1) Set the enable bits for the SDRAM pins (PFBCR1.MDSDE and PFBCR1.DQM1E in PFBCR1) to 1 to set the ports for SDRAM again.
   (2) Set the enable bit for the SDCLK pin (PFBCR1.SDCLKE in PFBCR1) to 1 to enable SDCLK pin output again.
   (3) Set the IOKEEP bit in DPSBYCR to 0 to release the I/O ports from the held state.
6. Set the PSTOP0 bit in SCKCR to 0 to start clock supply to the SDRAM via the SDCLK pin.
7. Check that all the status bits in SDSR are set to 0 and set the SFEN bit in SDSELF to 0 to cancel self-refresh mode.

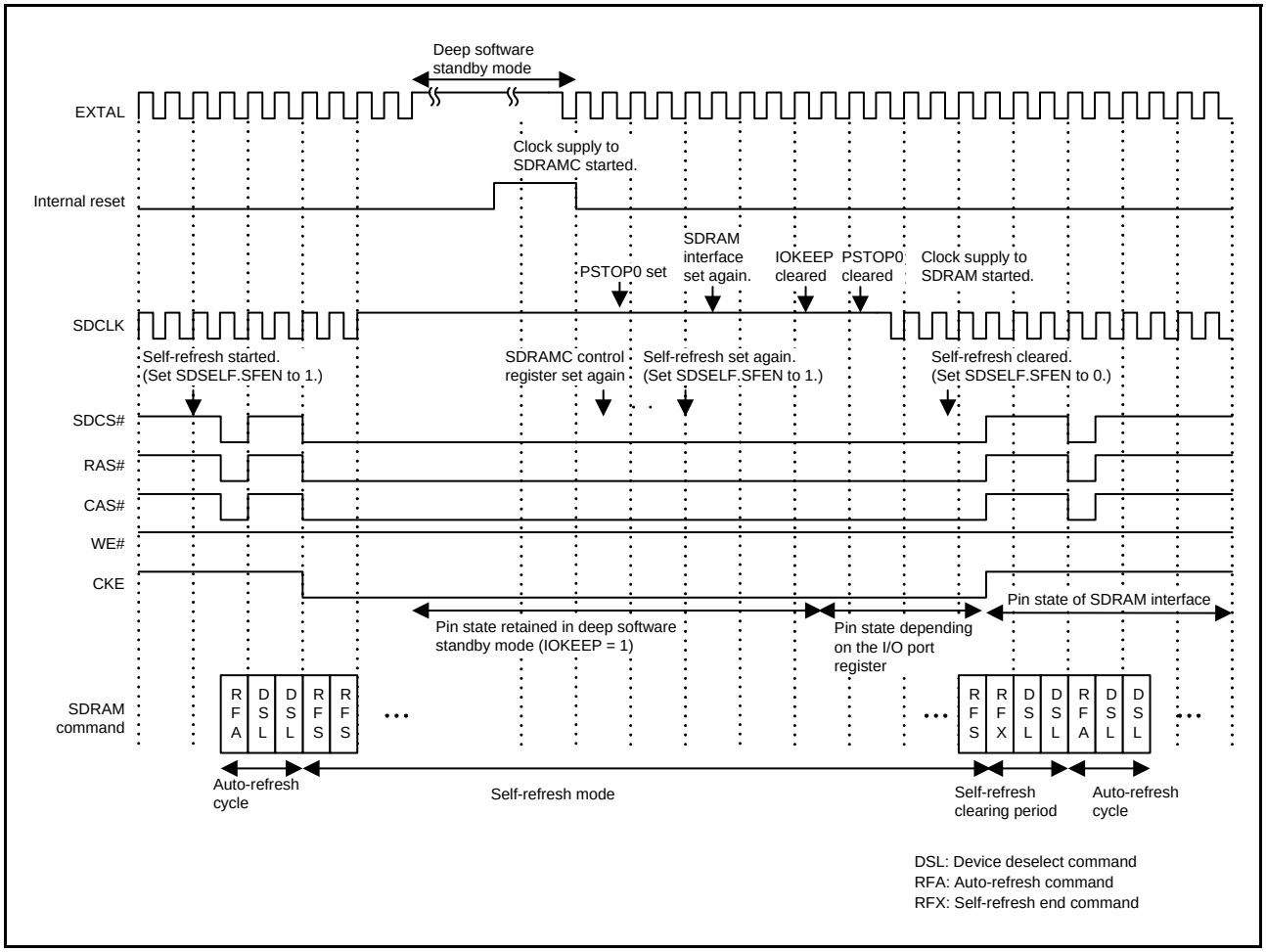


Figure 16.49 Timing Example of Self-Refresh Cycle (in Deep Software Standby Mode)

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 16.50 shows an example of the timing of an auto-refresh cycle.

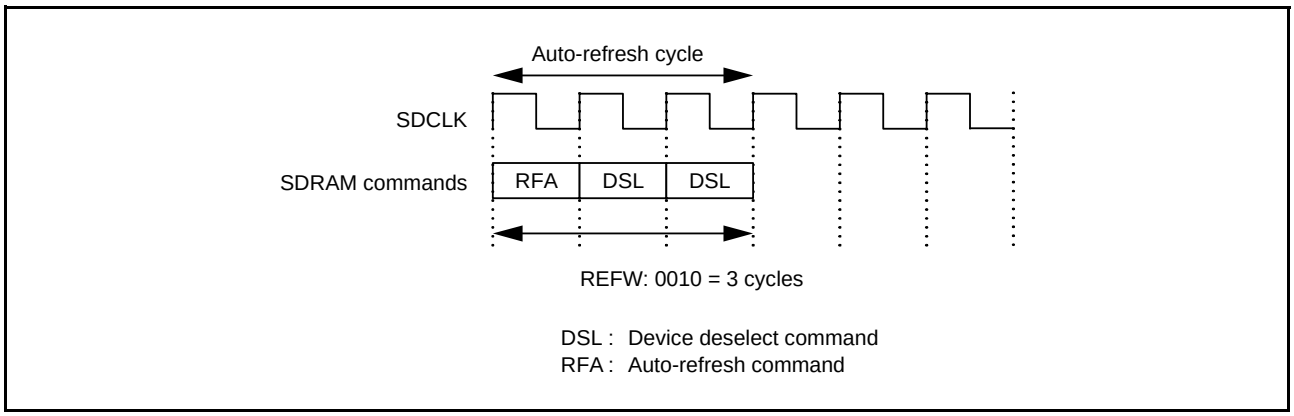


Figure 16.50 Timing Example of Auto-Refresh Cycle (1)

Figure 16.51 and Figure 16.52 show examples of operation when an auto-refresh request is generated during single access and continuous access, respectively.

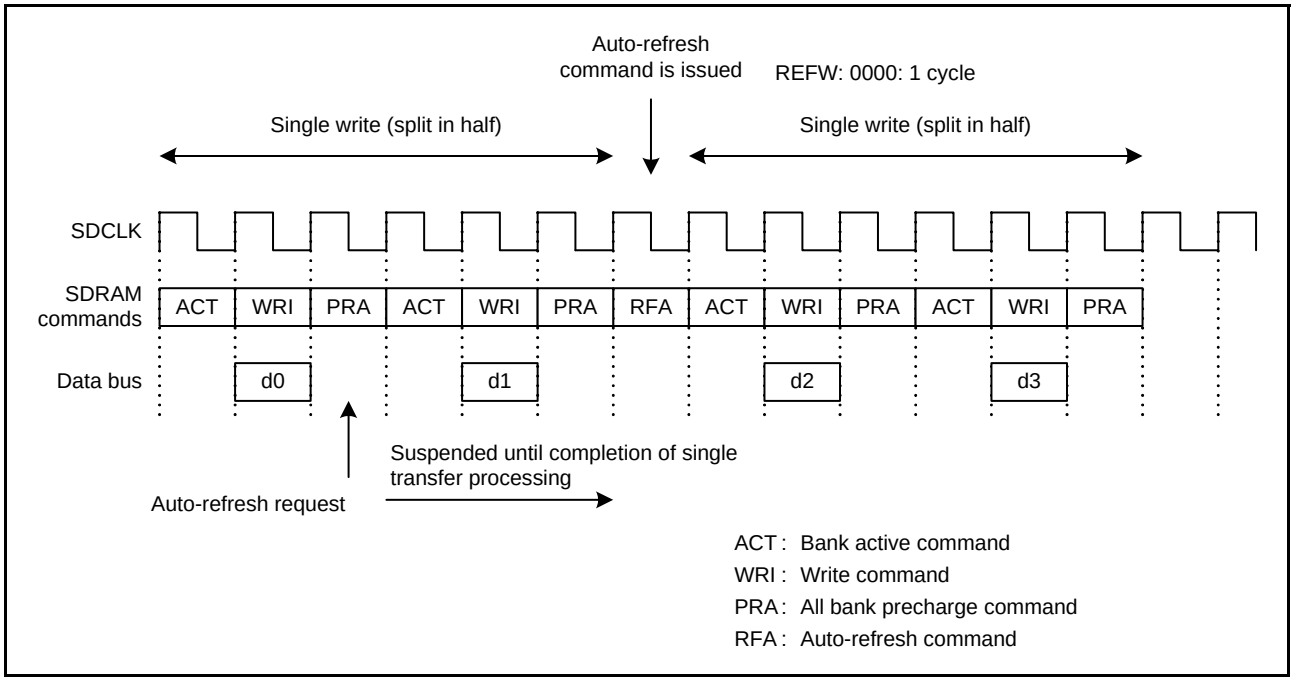


Figure 16.51 Timing Example of Auto-Refresh Cycle (2)
(Auto-Refresh Request is Made during Single Access)

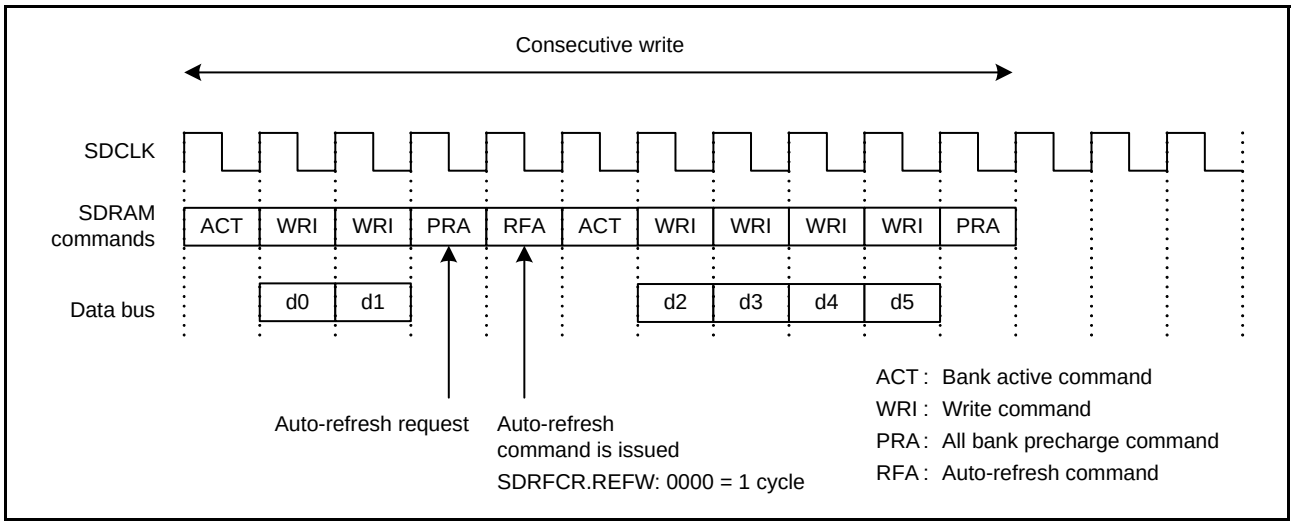


Figure 16.52 Timing Example of Auto-Refresh Cycle (3)
(Auto-Refresh Request is Made during Continuous Access)

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 16.16, Conditions for Register Modification.
Figure 16.53 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.

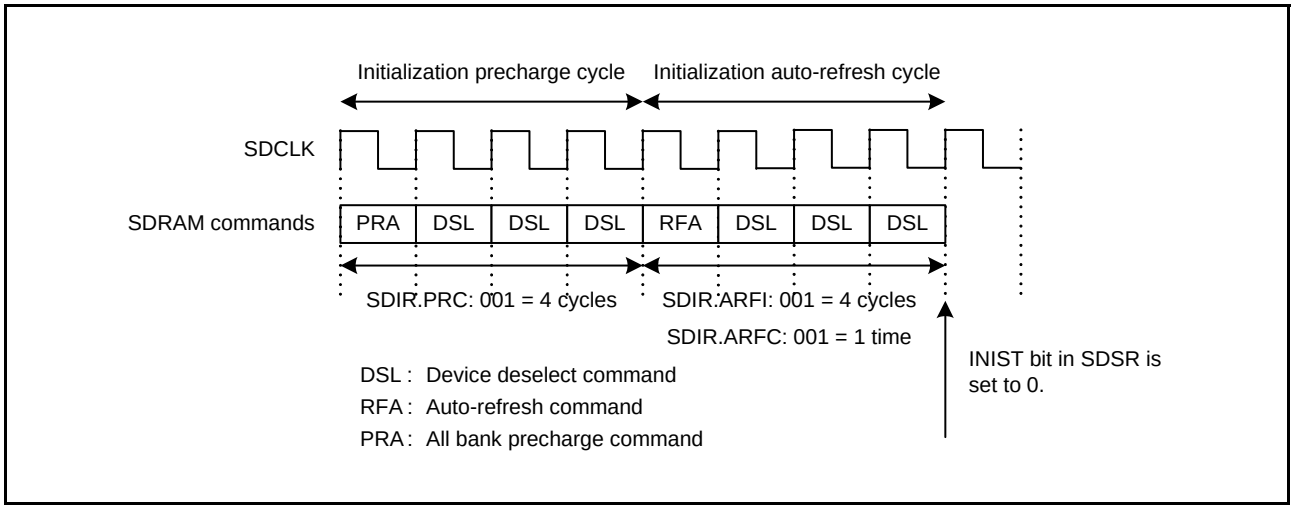


Figure 16.53 Timing Example of SDRAM Initialization Sequence

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 16.54 and Figure 16.55 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.

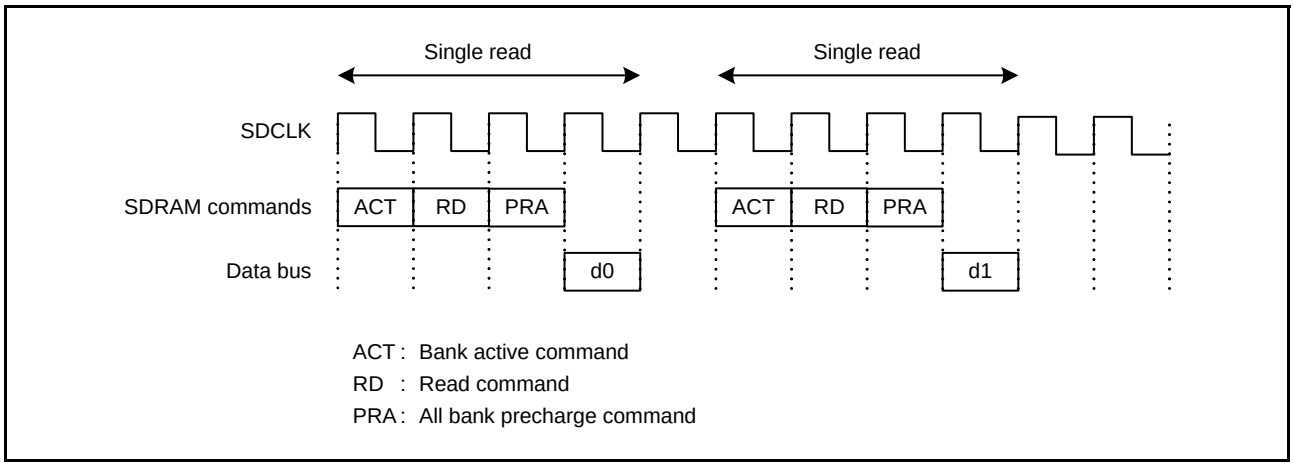


Figure 16.54 Timing Example of Single Read (SDTR.CL[2:0] = 010b: 2 Cycles)

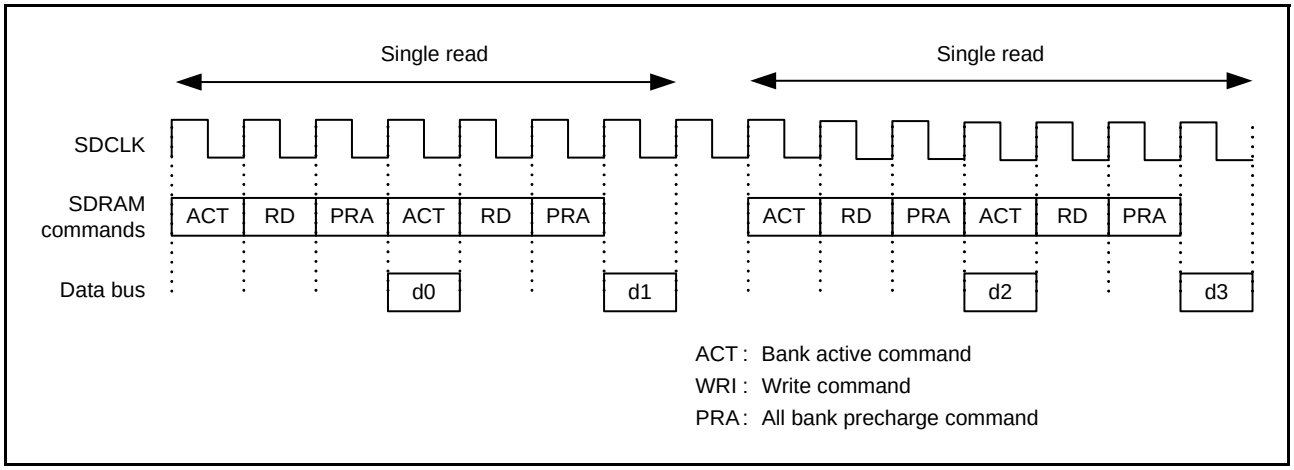


Figure 16.55 Timing Example of Single Read (Cluster Transfer by EXDMAC or Block Transfer in Single Address Mode with SDAMOD.BE = 0 and SDTR.CL[2:0] = 010b: 2 Cycles)

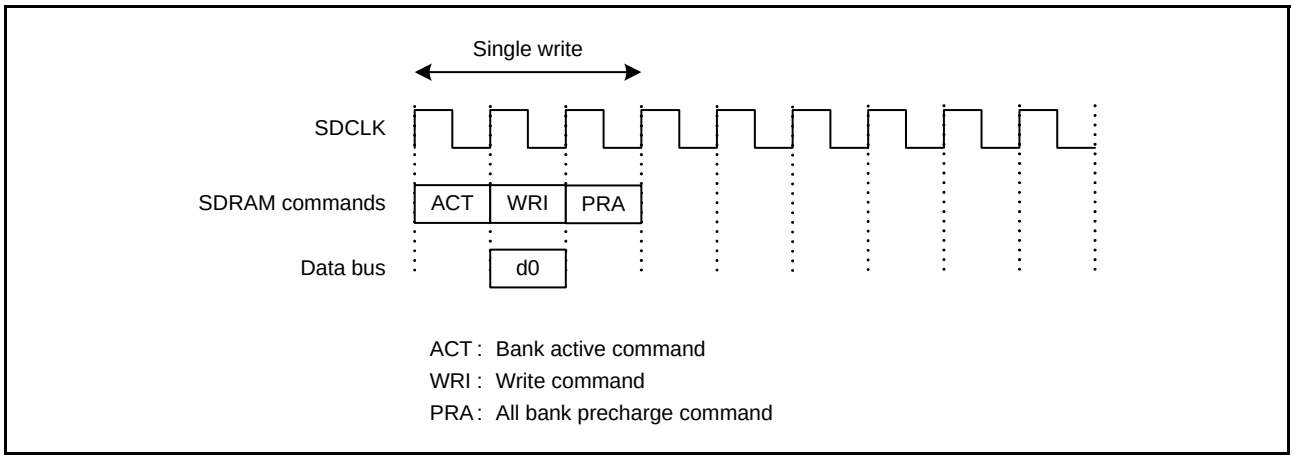


Figure 16.56 Timing Example of Single Write (when the Shortest Timing is Set)

(2) Consecutive Access

Figure 16.57 and Figure 16.58 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 16.59 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.

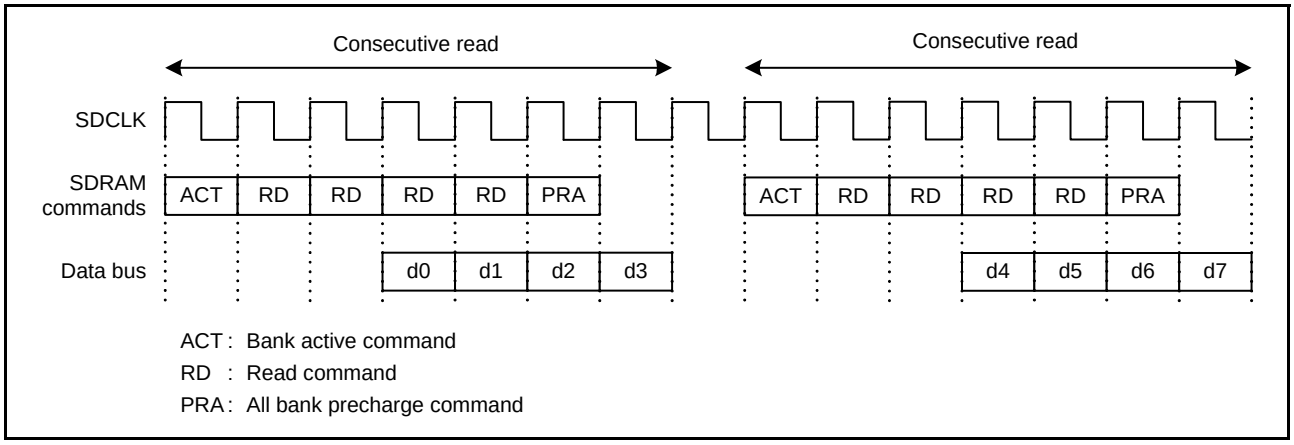


Figure 16.57 Timing Example of Consecutive Read (SDAMOD.BE = 1 and SDTR.CL[2:0] = 010b: 2 Cycles)

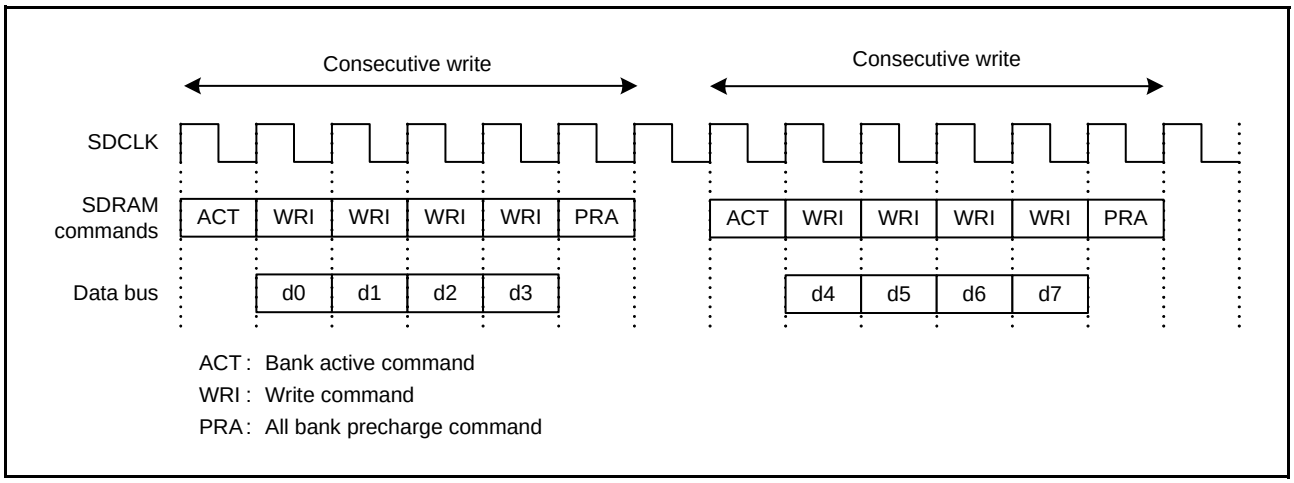


Figure 16.58 Timing Example of Consecutive Write (SDAMOD.BE = 1, when the Earliest Timing is Set)

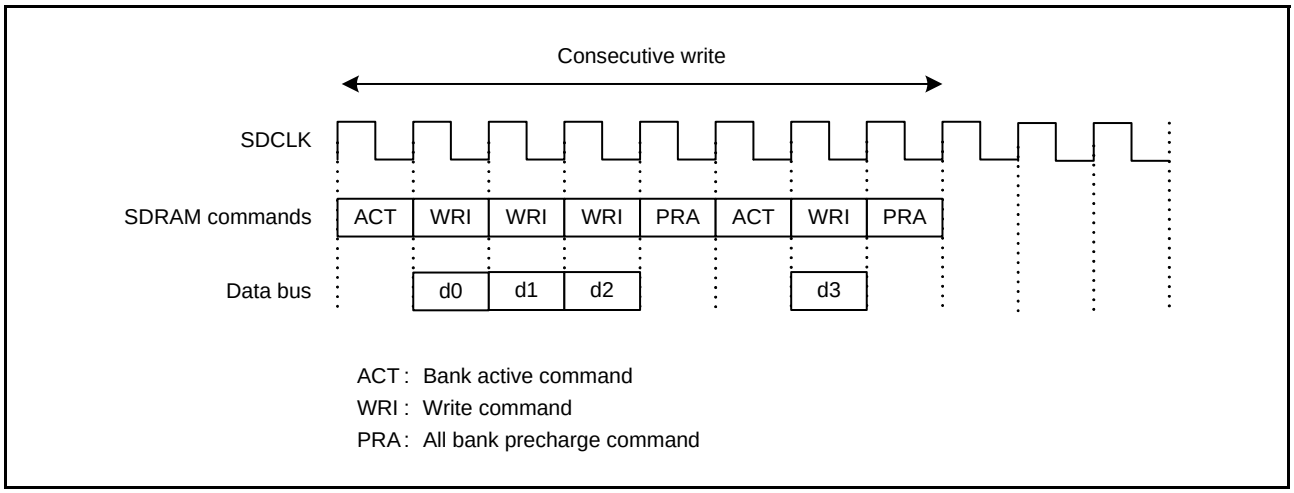


Figure 16.59 Timing Example of Consecutive Write (SDAMOD.BE = 1, when the Earliest Timing is Set) in which the Row Address Changes

16.6.11 Setting Mode Register

Setting the SDRAM mode register (SDMOD) allows the mode register set command to be issued to SDRAM and the value set in the MR[14:0] bits in SDMOD to be output to the lower bits of the address; specifically, to the A14 to A0 for 8-bit bus width, A15 to A1 for 16-bit bus width, and A16 to A2 for 32-bit bus width. Therefore, set the SDCCR.BSIZE[1:0] bits before setting the mode register, to determine the data bus width of the SDRAM.
Figure 16.60 shows the mode register setting timing.

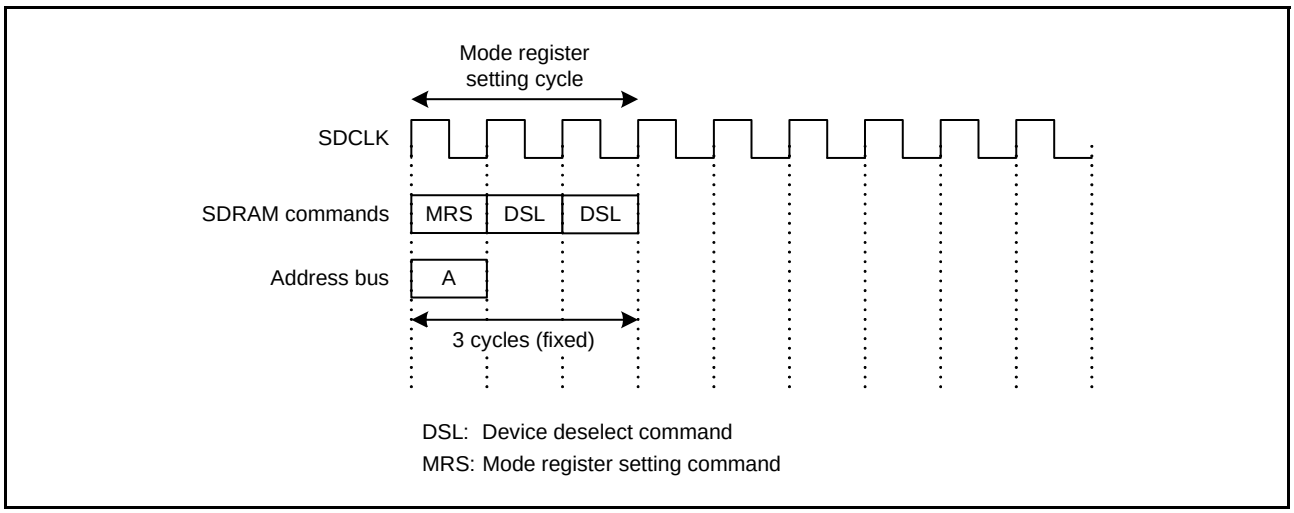


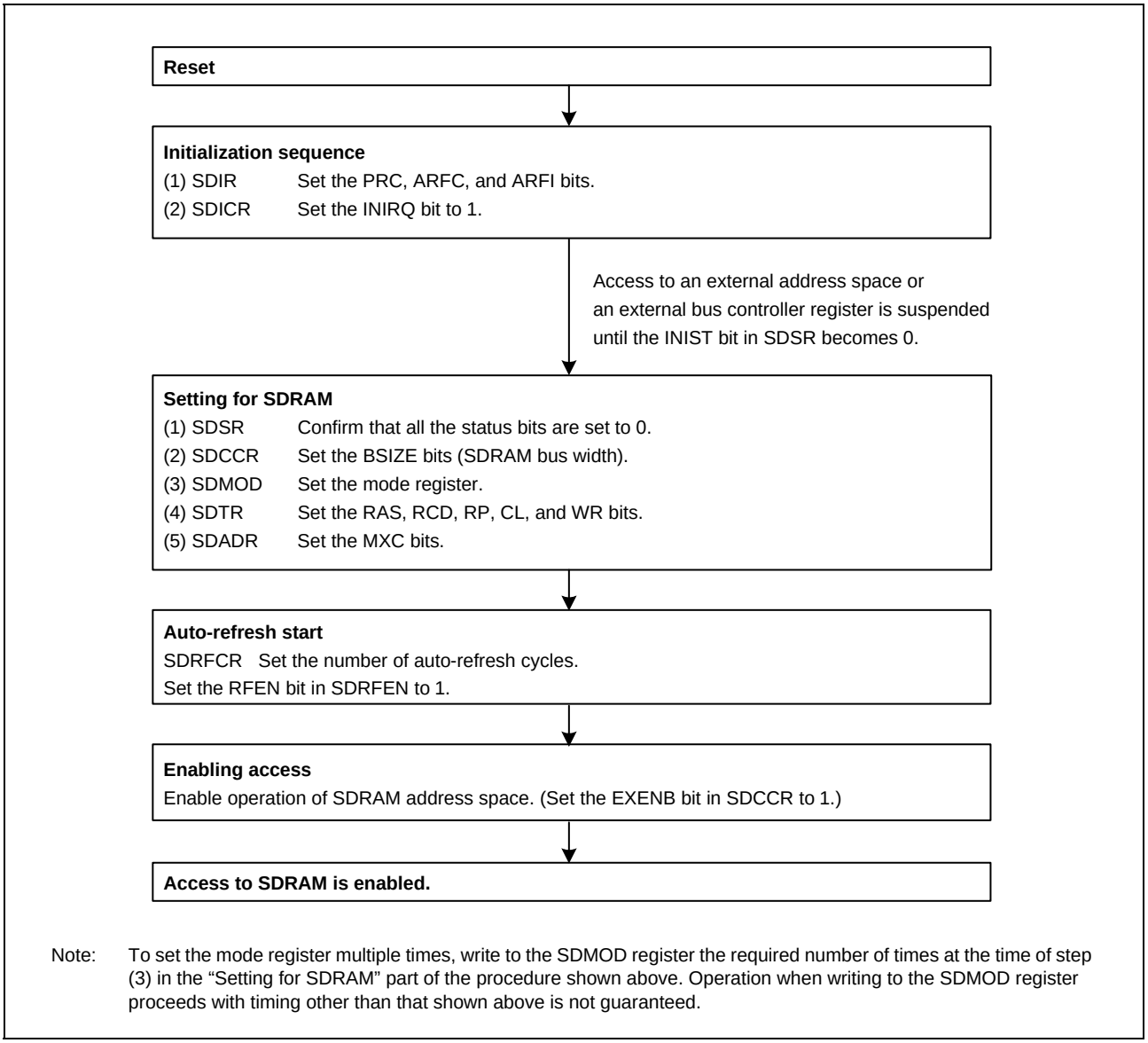
Figure 16.60 Mode Register Setting Timing

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 16.61 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.



Note: To set the mode register multiple times, write to the SDMOD register the required number of times at the time of step (3) in the “Setting for SDRAM” part of the procedure shown above. Operation when writing to the SDMOD register proceeds with timing other than that shown above is not guaranteed.

Figure 16.61 SDRAMC Setting Procedure

16.6.12.2 Procedure for Transition to and Recovery from Self-Refresh Mode

Figure 16.62 shows the procedure for transition to and recovery from self-refresh mode.

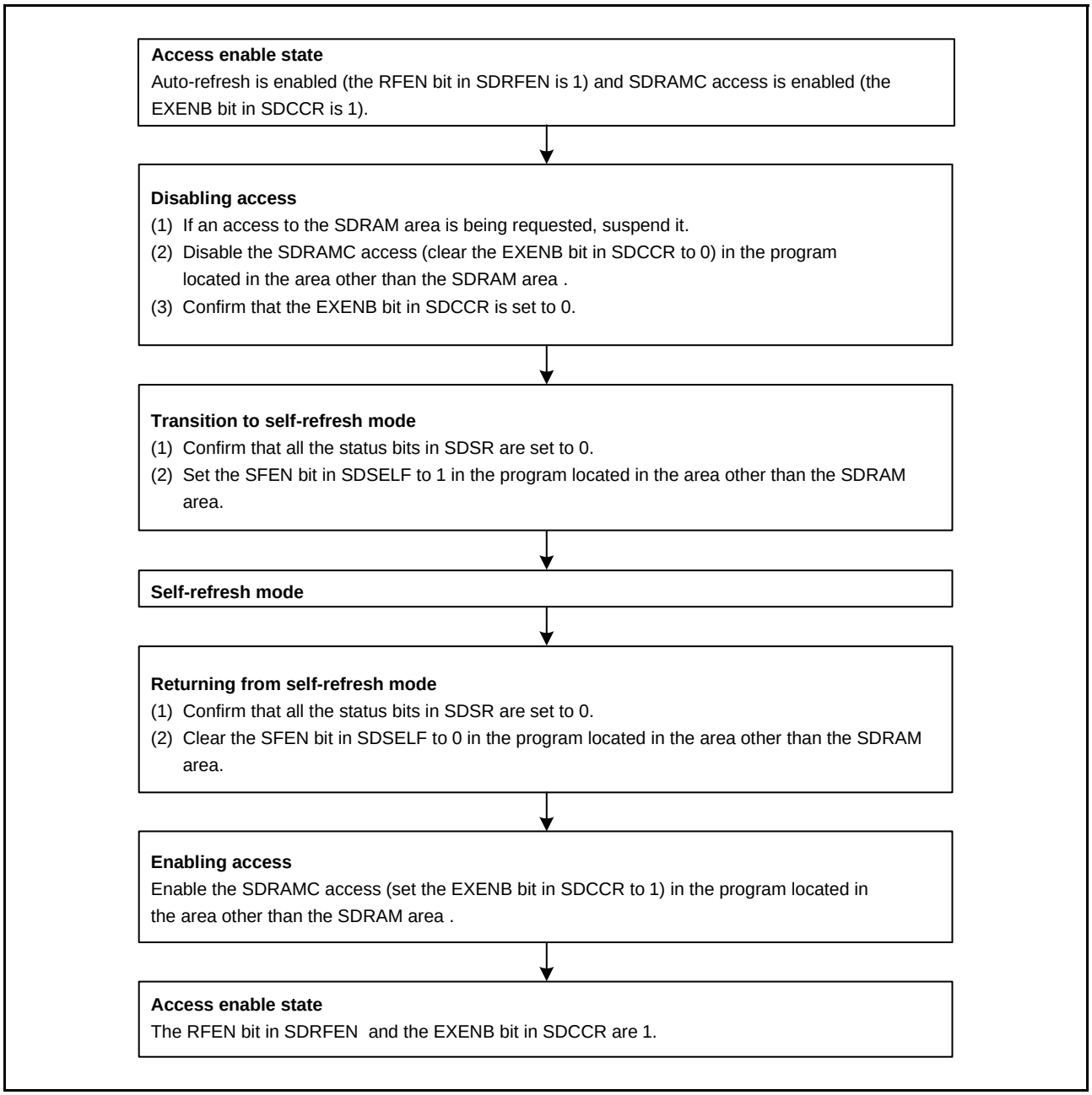


Figure 16.62 Procedure for Transition to and Recovery from Self-Refresh Mode

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 16.63 shows the procedure for transition to and recovery from self-refresh mode in deep software standby mode.

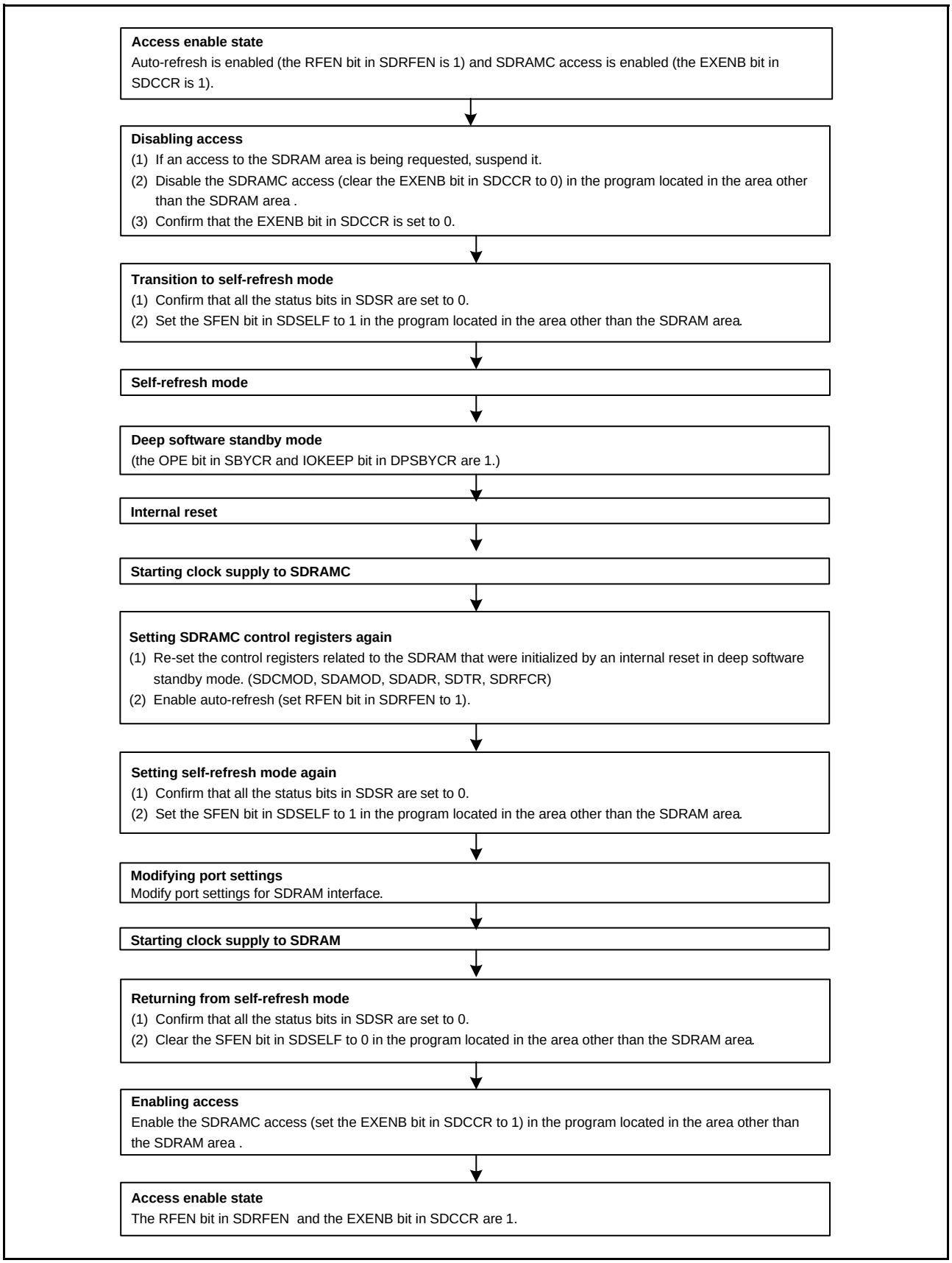


Figure 16.63 Procedure for Transition to and Recovery from Self-Refresh Mode in Deep Software Standby Mode

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 16.64 to Figure 16.68 show the relationship between the single read timing and the SDTR register settings. Table 16.17 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 16.68.

Table 16.17 Correspondence between Target Figures and SDTR Register Settings (Single Read Timing)

Figure No.	RAS[2:0] Settings	Number of Cycles	RCD[1:0] Settings	Number of Cycles	RP[2:0] Settings	Number of Cycles	CL[2:0] Settings	Number of Cycles
Figure 16.64	010	3	00	1	001	2	010	2
Figure 16.65	000	1	01	2	001	2	010	2
Figure 16.66	000	1	01	2	001	2	011	3
Figure 16.67, Figure 16.68	010	3	00	1	000	1	010	2

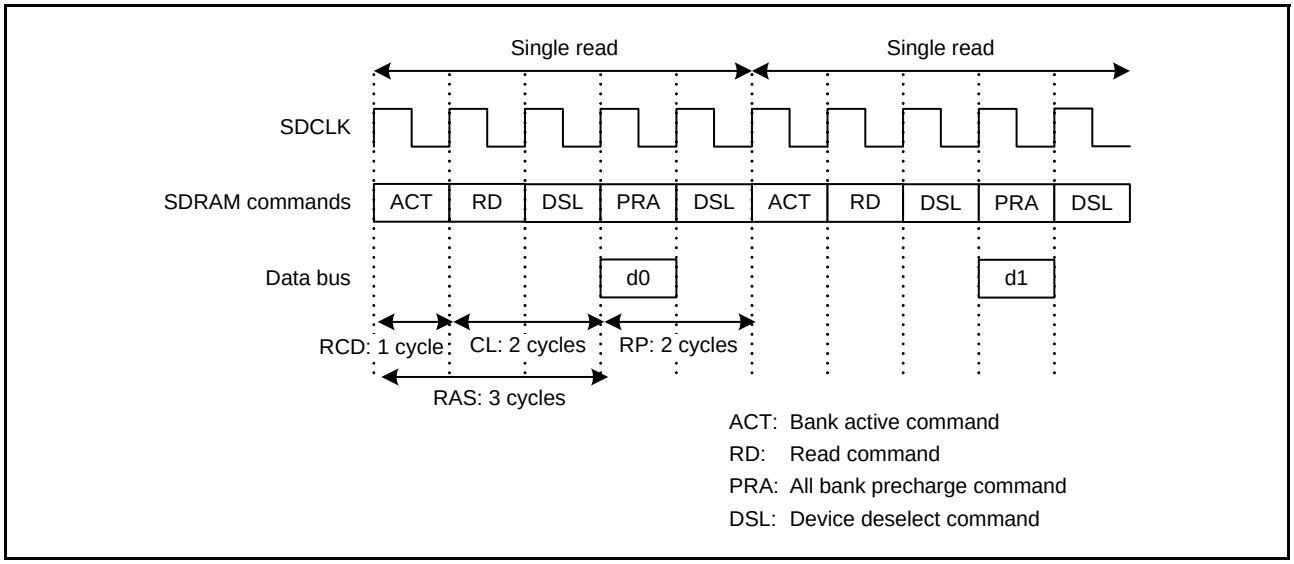


Figure 16.64 Timing Example of Single Read (1)

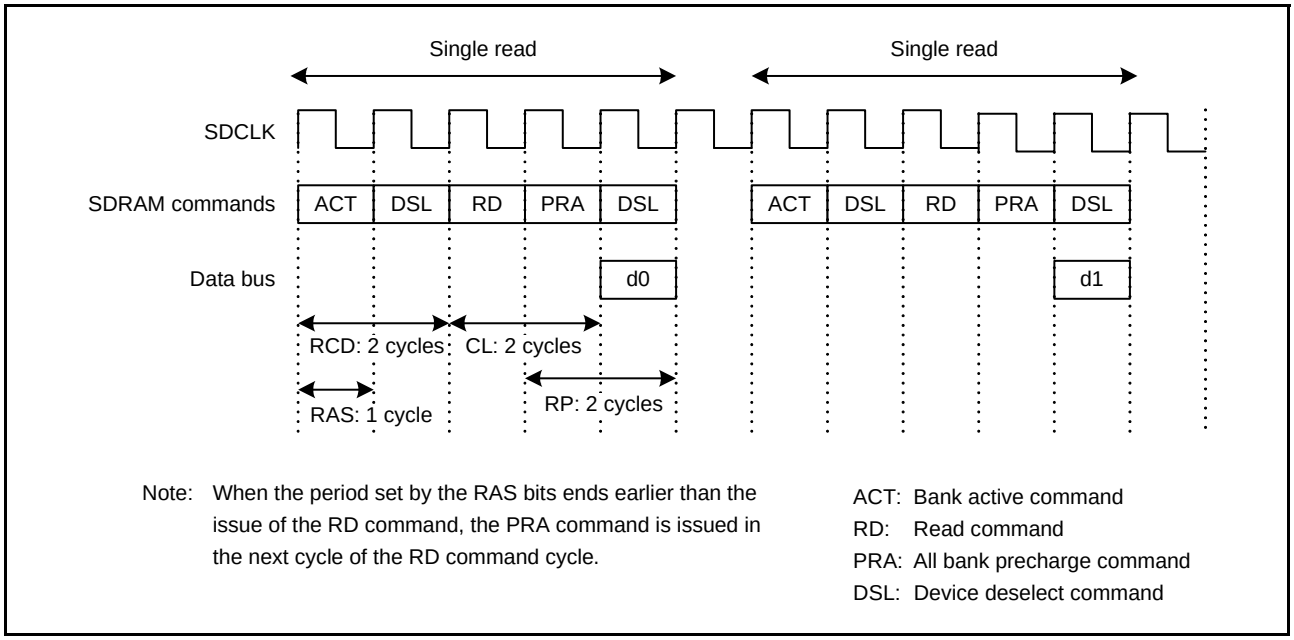


Figure 16.65 Timing Example of Single Read (2)

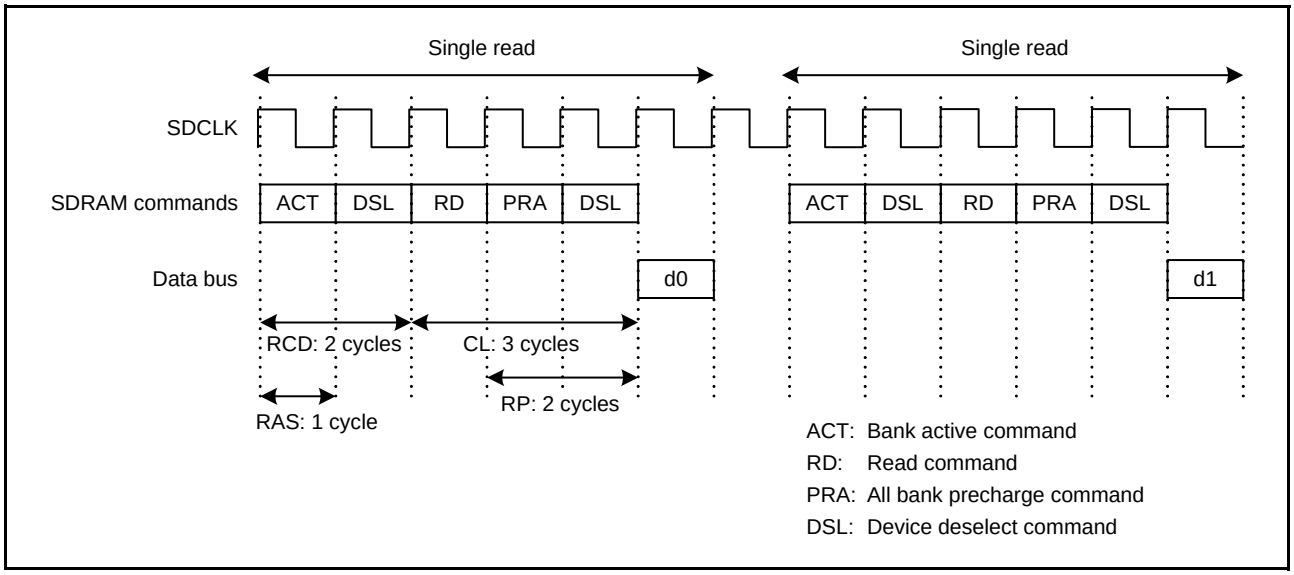


Figure 16.66 Timing Example of Single Read (3)

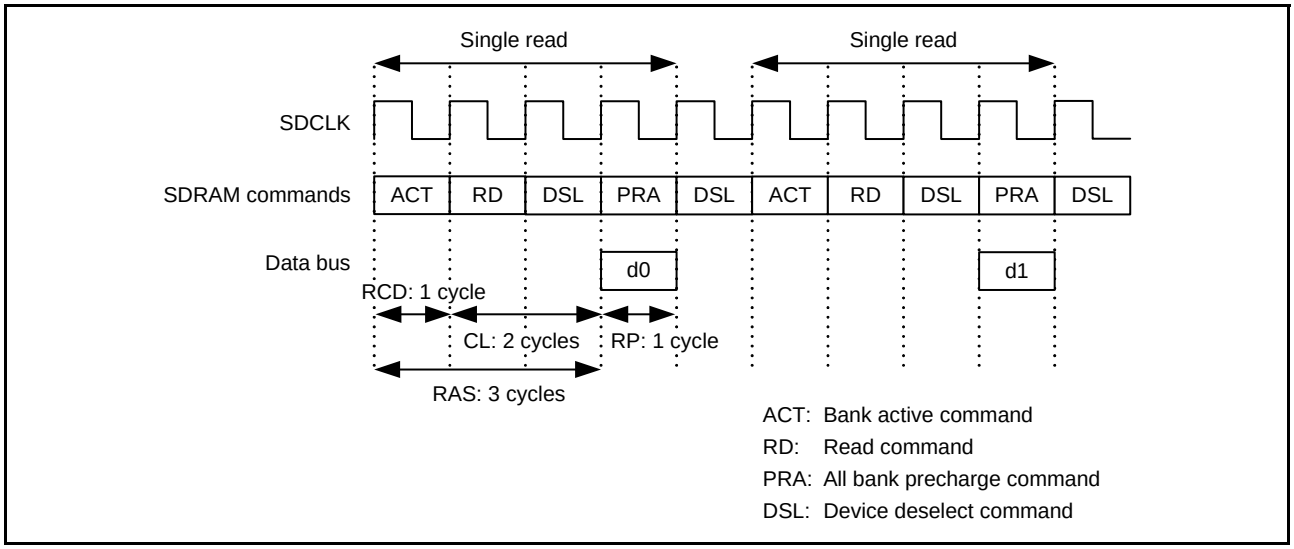


Figure 16.67 Timing Example of Single Read (4)

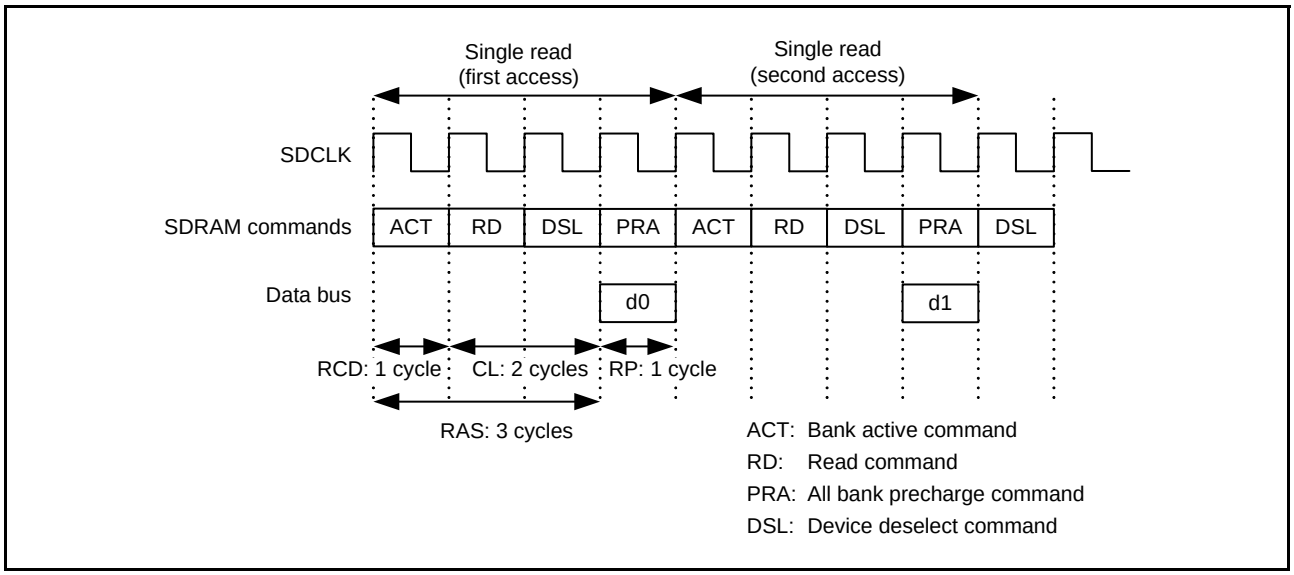


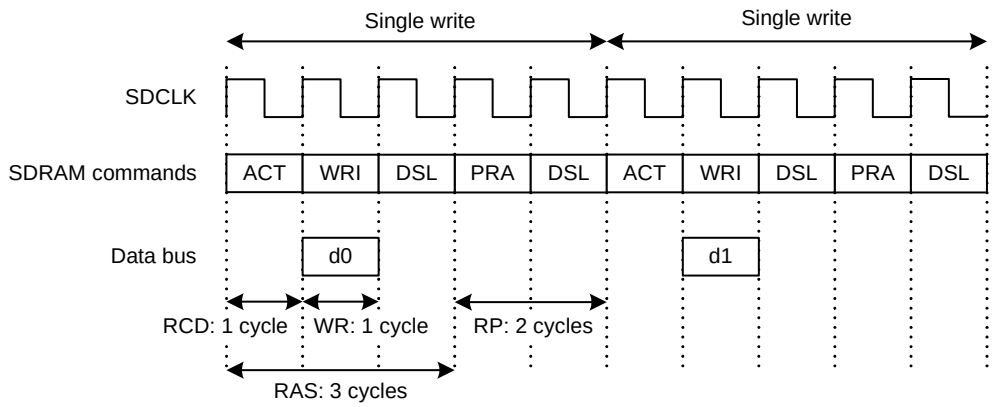
Figure 16.68 Timing Example of Single Read (5) (Two Bus Accesses Occur for One Transfer Request)

(2) Single Write Timing Examples

Figure 16.69 to Figure 16.73 show the relationship between the single write timing and the SDTR register settings. Table 16.18 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 16.73.

Table 16.18 Correspondence between Target Figures and SDTR Register Settings (Single Write Timing)

Figure No.	RAS[2:0] Settings	Number of Cycles	RCD[1:0] Settings	Number of Cycles	RP[2:0] Settings	Number of Cycles	WR Settings	Number of Cycles
Figure 16.69	010	3	00	1	001	2	0	1
Figure 16.70	000	1	01	2	001	2	0	1
Figure 16.71	000	1	01	2	001	2	1	2
Figure 16.72, Figure 16.73	010	3	00	1	000	1	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 16.69 Timing Example of Single Write (1)

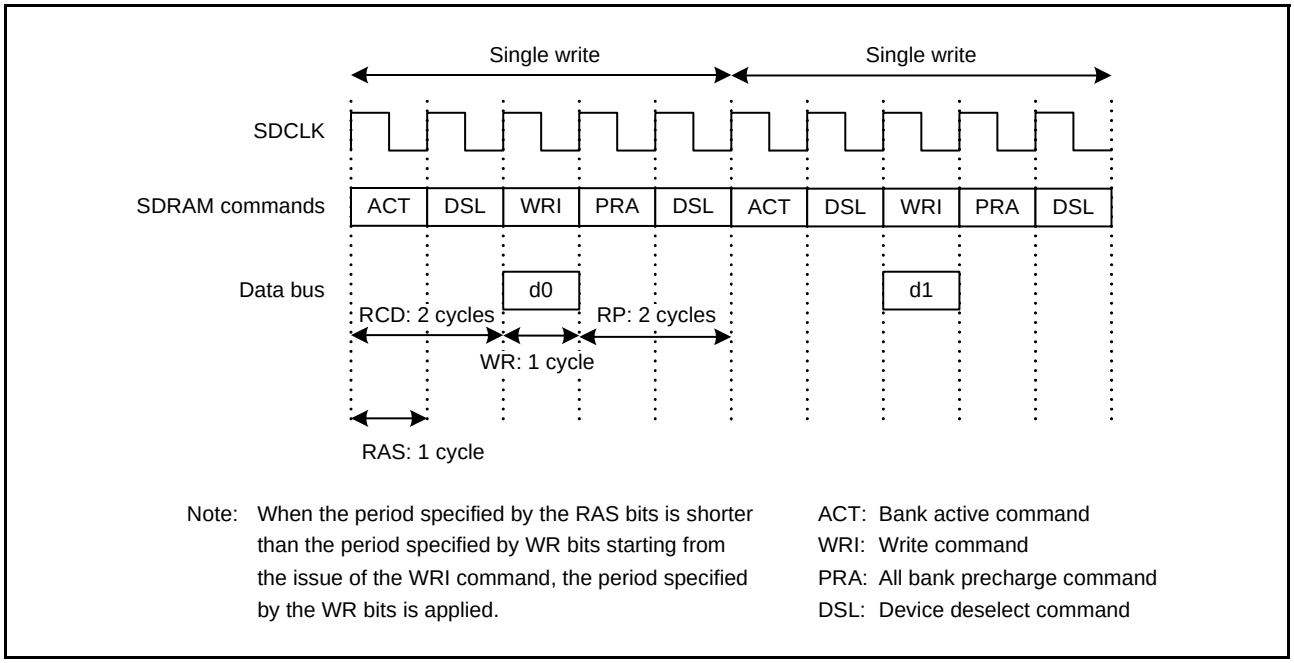


Figure 16.70 Timing Example of Single Write (2)

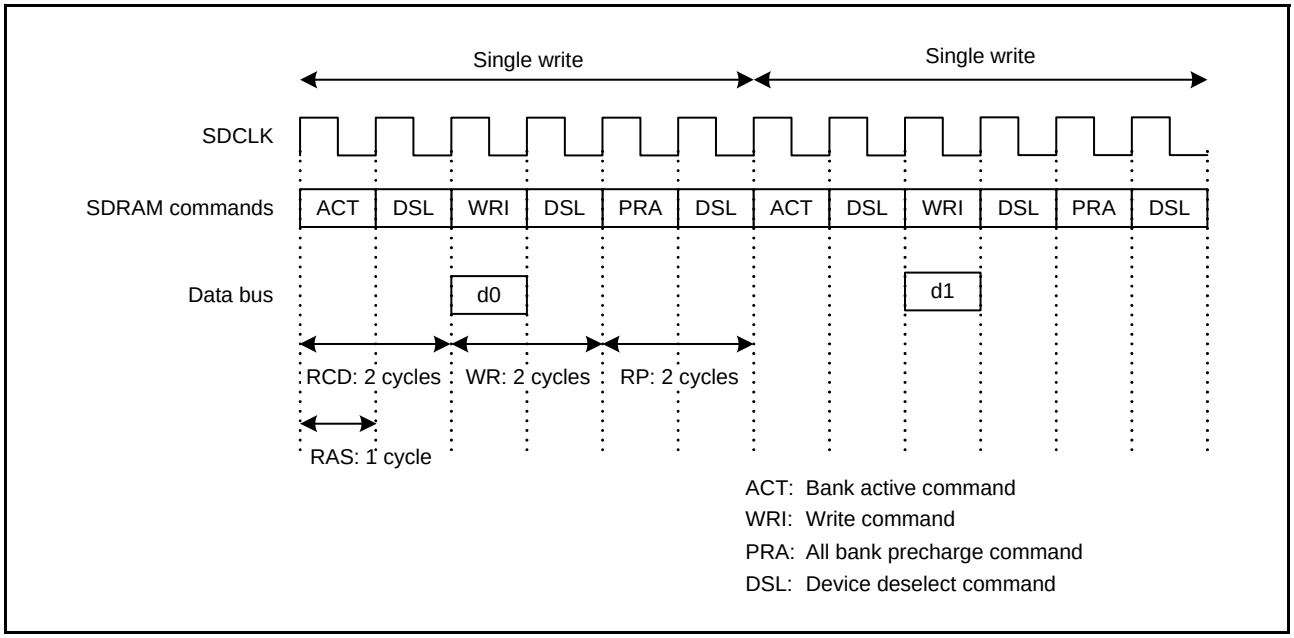
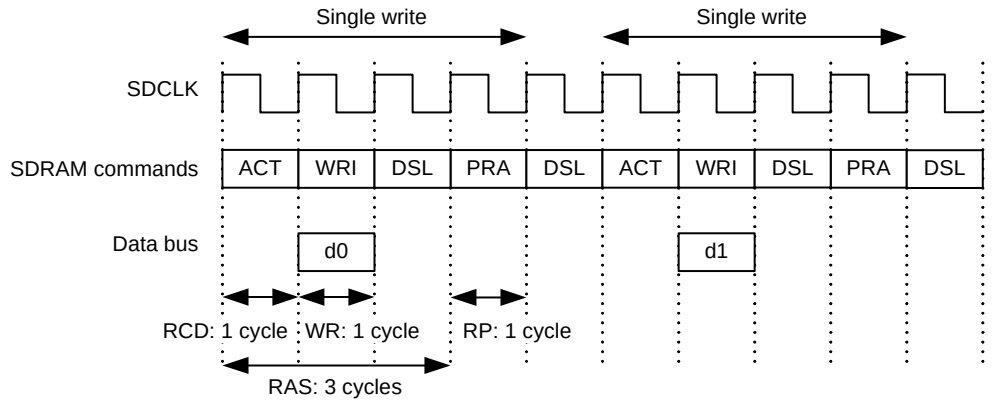


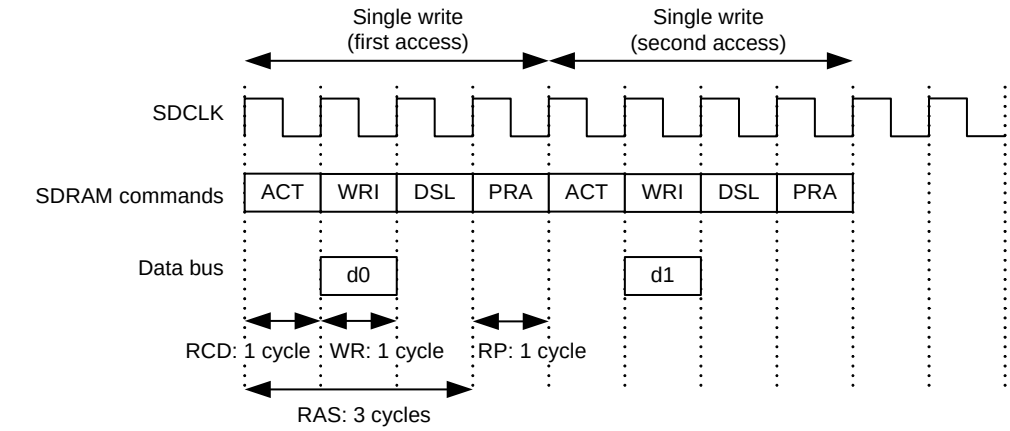
Figure 16.71 Timing Example of Single Write (3)



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 16.72 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 16.73 Timing Example of Single Write (5) (Two Bus Accesses Occur for One Transfer Request)

(3) Consecutive Read Timing Examples

Figure 16.74 to Figure 16.76 show the relationship between the consecutive read timing for four data and the SDTR register settings. Table 16.19 shows the correspondence between the figures and the SDTR register settings.

Table 16.19 Correspondence between Target Figures and SDTR Register Settings (Consecutive 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 16.74	010	3	00	1	001	2	010	2
Figure 16.75	000	1	01	2	001	2	010	2
Figure 16.76	000	1	01	2	001	2	011	3

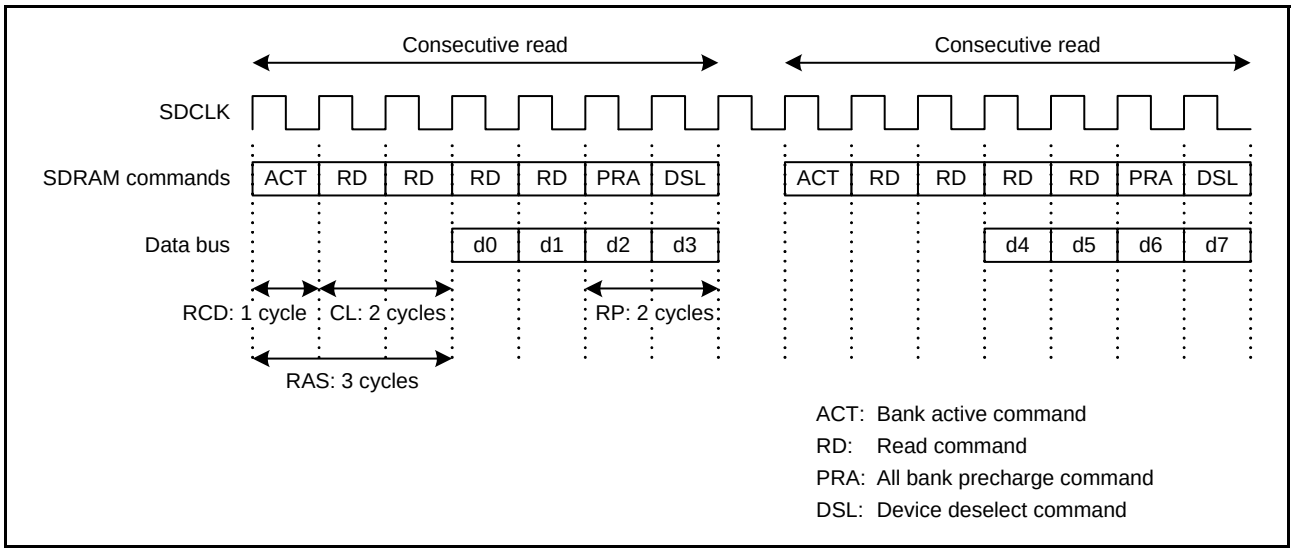


Figure 16.74 Timing Example of Consecutive Read (1)

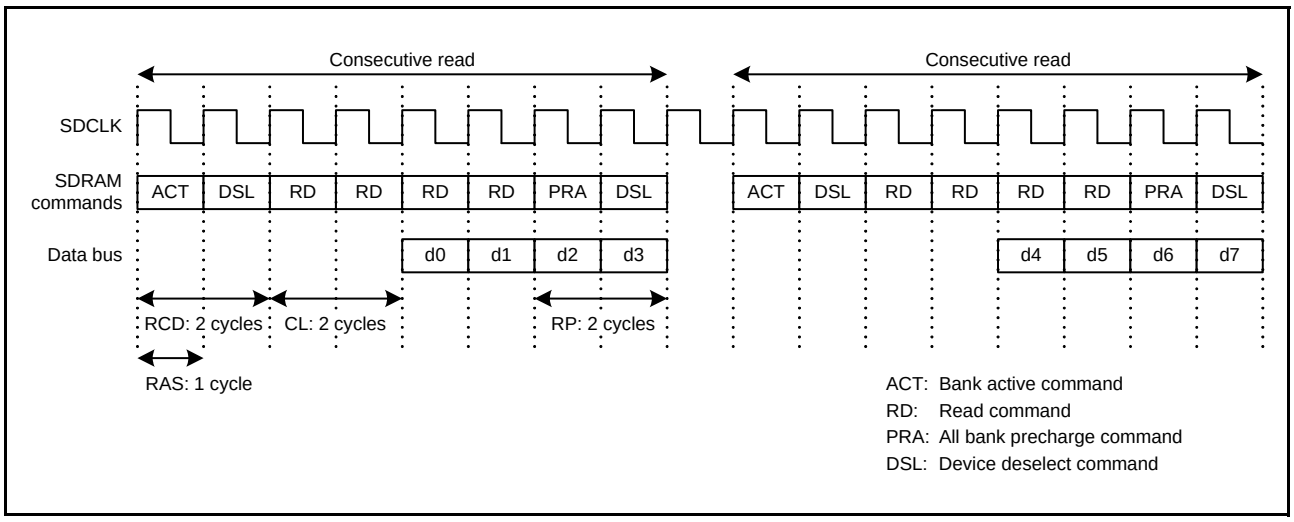


Figure 16.75 Timing Example of Consecutive Read (2)

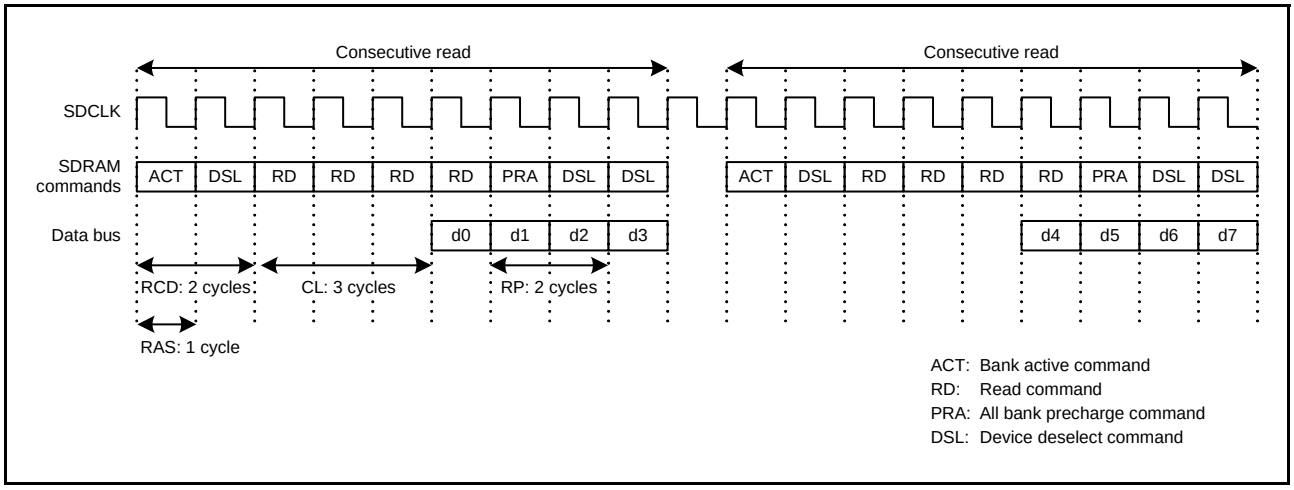


Figure 16.76 Timing Example of Consecutive Read (3)

(4) Consecutive Write Timing Examples

Figure 16.77 to Figure 16.79 show the relationship between the consecutive write timing for four data and the SDTR register settings. Table 16.20 shows the correspondence between the figures and the SDTR register settings.

Table 16.20 Correspondence between Target Figures and SDTR Register Settings (Consecutive Write Timing)

Figure No.	RAS[2:0] Settings	Number of Cycles	RCD[1:0] Settings	Number of Cycles	RP[2:0] Settings	Number of Cycles	WR Settings	Number of Cycles
Figure 16.77	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

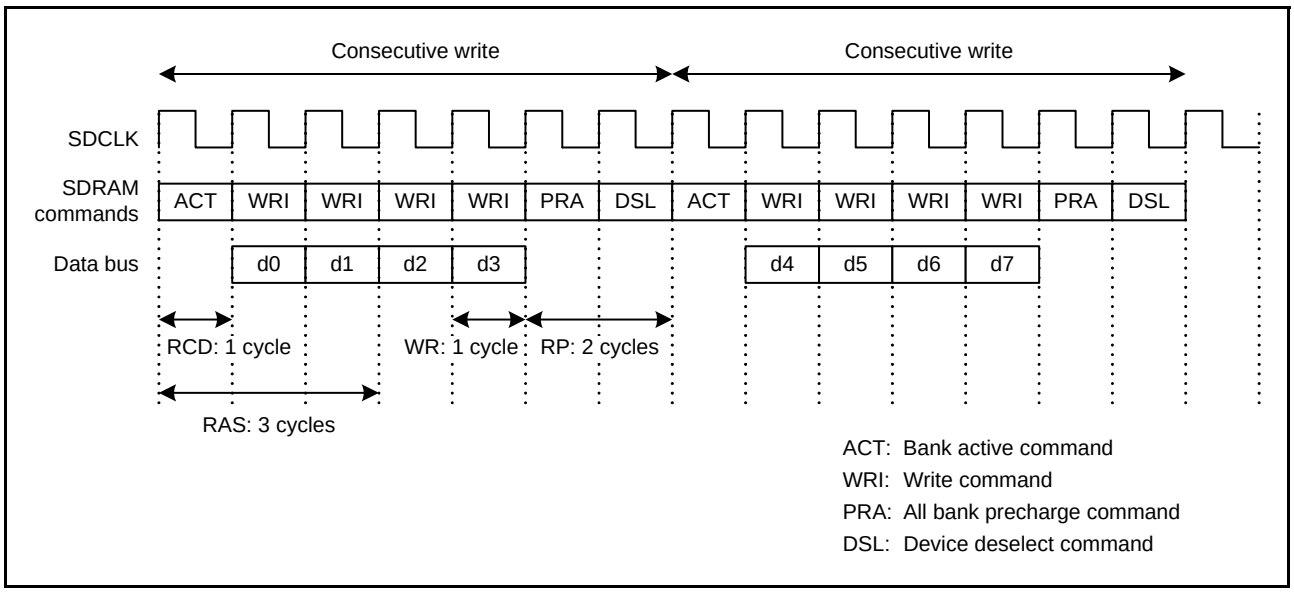


Figure 16.77 Timing Example of Consecutive Write (1)

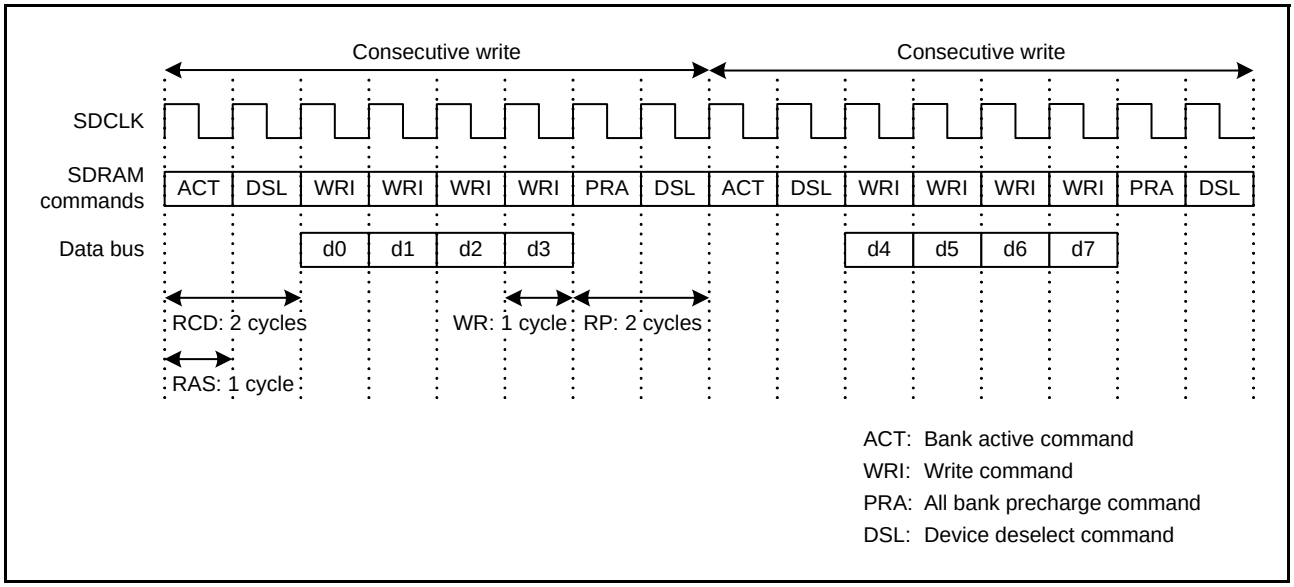


Figure 16.78 Timing Example of Consecutive Write (2)

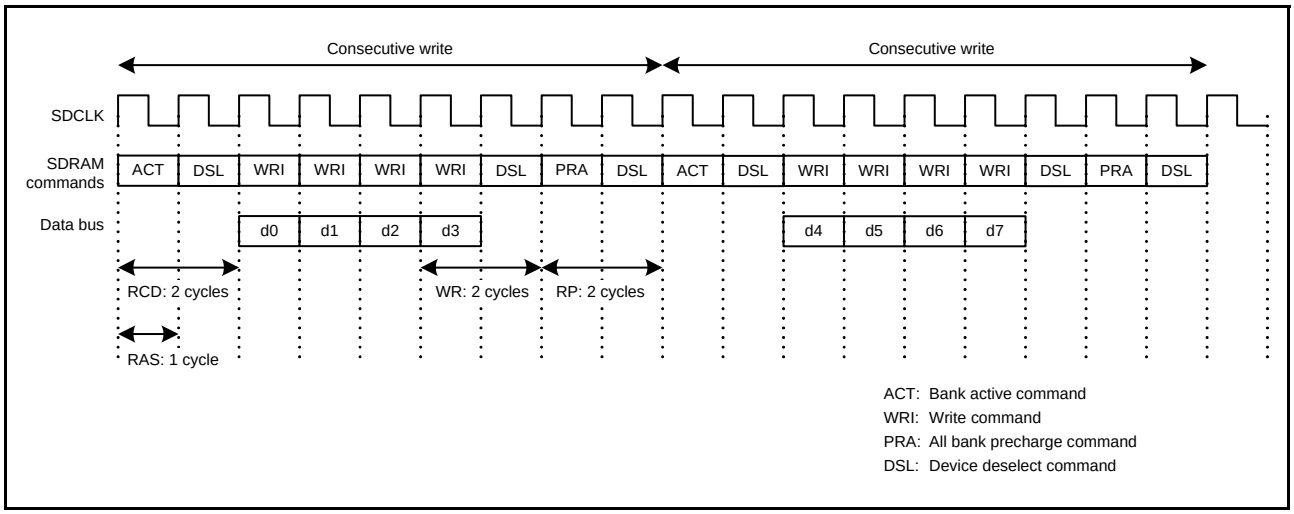


Figure 16.79 Timing Example of Consecutive Write (3)

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 16.21 shows the relationship between the SDADR.MXC[1:0] settings and the shift amount.

Table 16.21 Address Multiplexing

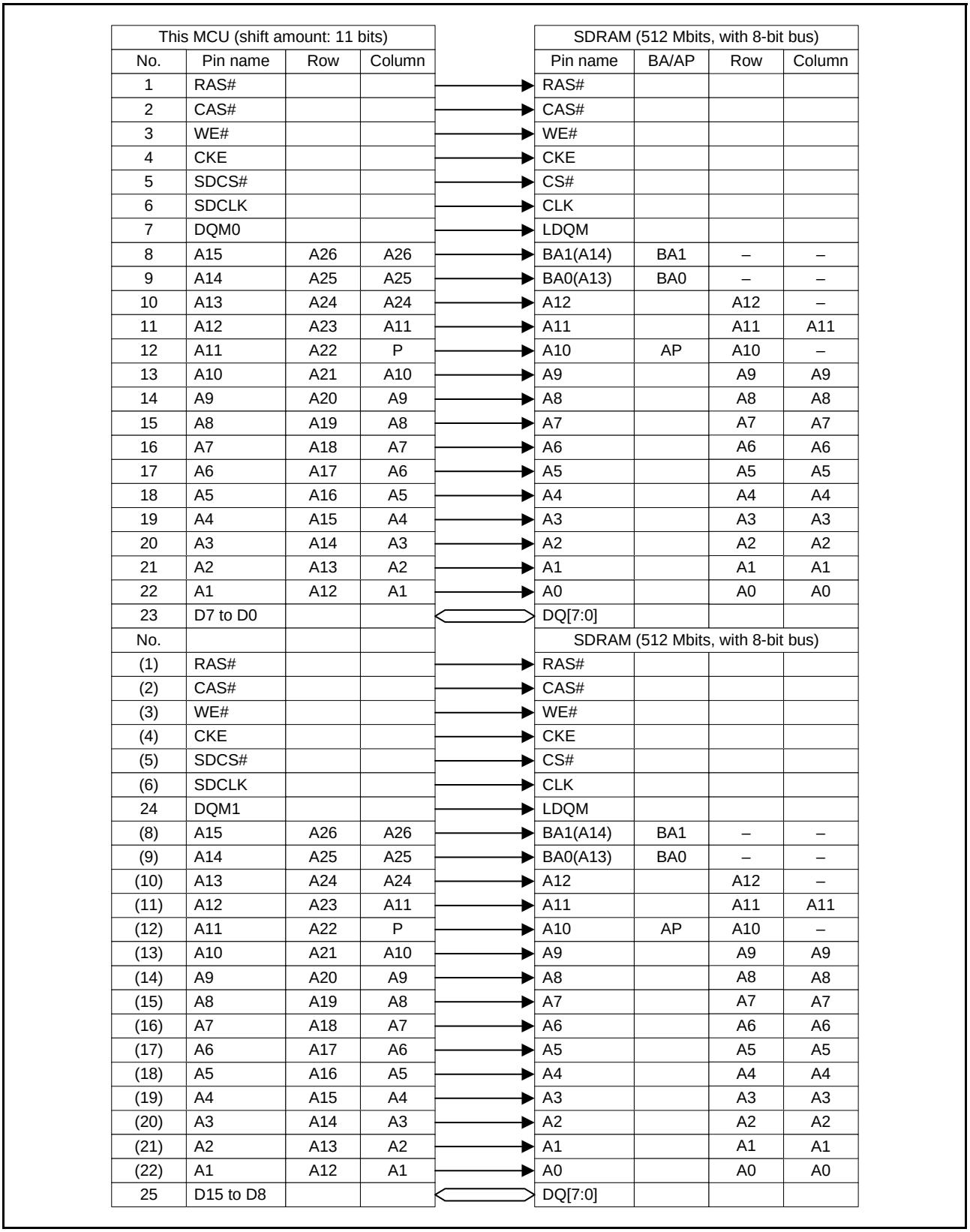
MXC [1:0]	Sift Amount	Data Bus Width	Address	Address Pins External to the Microcomputer																		
				A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
00	8 bits	8 bits	Row	A26	A25	A24	A23	A22	A21	A20	A19	A18*	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8
			Column	A26	A25	A24	A23	A22	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
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
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
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

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 16-Bit Bus Space

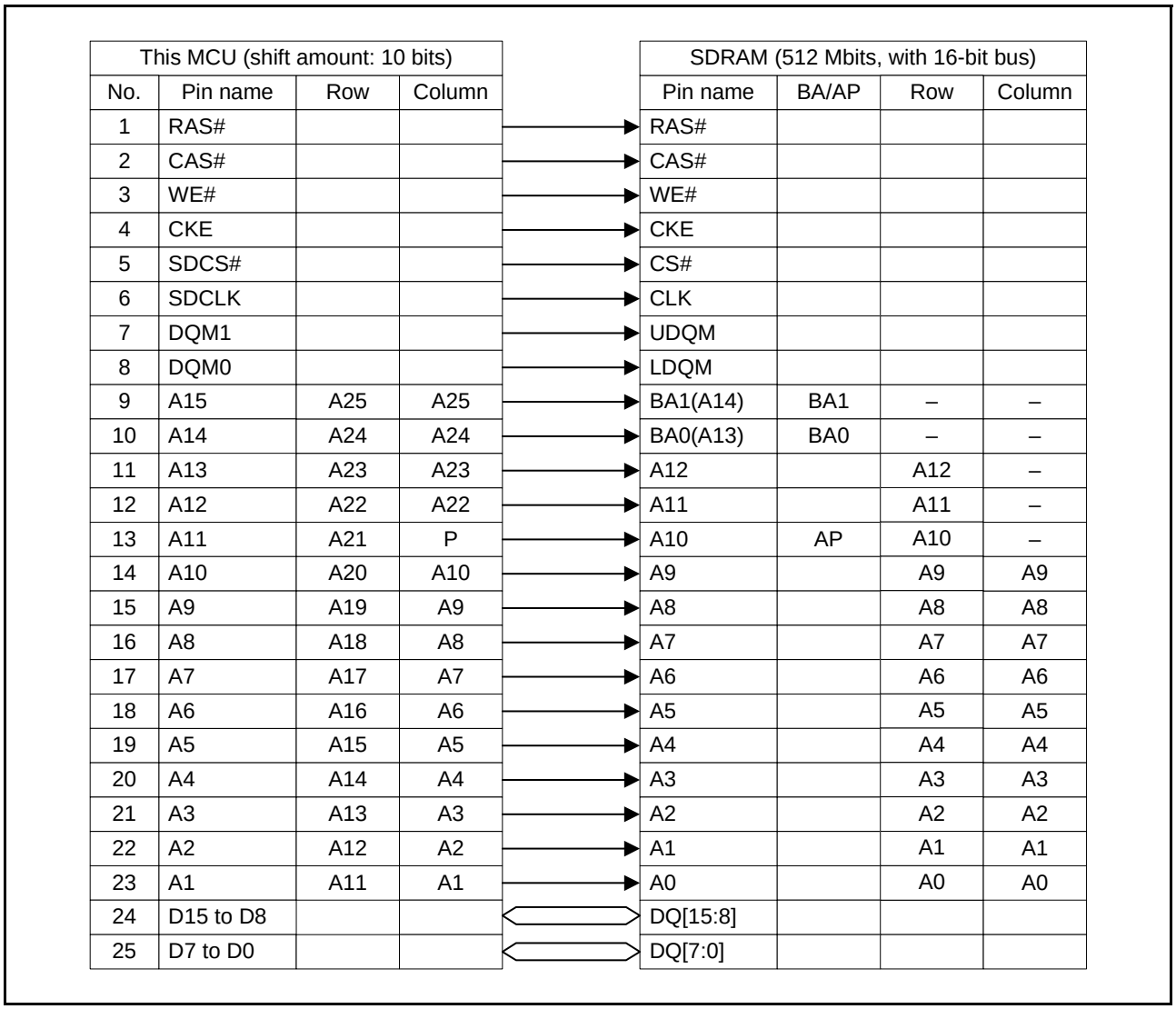
Figure 16.80 shows an example for connecting to two 512-Mbit SDRAMs with 13-bit row address, 11-bit column address and 8-bit bus.



This MCU (shift amount: 11 bits)					SDRAM (512 Mbits, with 8-bit bus)			
No.	Pin name	Row	Column		Pin name	BA/AP	Row	Column
1	RAS#			→	RAS#			
2	CAS#			→	CAS#			
3	WE#			→	WE#			
4	CKE			→	CKE			
5	SDCS#			→	CS#			
6	SDCLK			→	CLK			
7	DQM0			→	LDQM			
8	A15	A26	A26	→	BA1(A14)	BA1	–	–
9	A14	A25	A25	→	BA0(A13)	BA0	–	–
10	A13	A24	A24	→	A12		A12	–
11	A12	A23	A11	→	A11		A11	A11
12	A11	A22	P	→	A10	AP	A10	–
13	A10	A21	A10	→	A9		A9	A9
14	A9	A20	A9	→	A8		A8	A8
15	A8	A19	A8	→	A7		A7	A7
16	A7	A18	A7	→	A6		A6	A6
17	A6	A17	A6	→	A5		A5	A5
18	A5	A16	A5	→	A4		A4	A4
19	A4	A15	A4	→	A3		A3	A3
20	A3	A14	A3	→	A2		A2	A2
21	A2	A13	A2	→	A1		A1	A1
22	A1	A12	A1	→	A0		A0	A0
23	D7 to D0			⟷	DQ[7:0]			
No.					SDRAM (512 Mbits, with 8-bit bus)			
(1)	RAS#			→	RAS#			
(2)	CAS#			→	CAS#			
(3)	WE#			→	WE#			
(4)	CKE			→	CKE			
(5)	SDCS#			→	CS#			
(6)	SDCLK			→	CLK			
24	DQM1			→	LDQM			
(8)	A15	A26	A26	→	BA1(A14)	BA1	–	–
(9)	A14	A25	A25	→	BA0(A13)	BA0	–	–
(10)	A13	A24	A24	→	A12		A12	–
(11)	A12	A23	A11	→	A11		A11	A11
(12)	A11	A22	P	→	A10	AP	A10	–
(13)	A10	A21	A10	→	A9		A9	A9
(14)	A9	A20	A9	→	A8		A8	A8
(15)	A8	A19	A8	→	A7		A7	A7
(16)	A7	A18	A7	→	A6		A6	A6
(17)	A6	A17	A6	→	A5		A5	A5
(18)	A5	A16	A5	→	A4		A4	A4
(19)	A4	A15	A4	→	A3		A3	A3
(20)	A3	A14	A3	→	A2		A2	A2
(21)	A2	A13	A2	→	A1		A1	A1
(22)	A1	A12	A1	→	A0		A0	A0
25	D15 to D8			⟷	DQ[7:0]			

Figure 16.80 SDRAM Connection Example (512-Mbit × 2, with 8-Bit Bus)

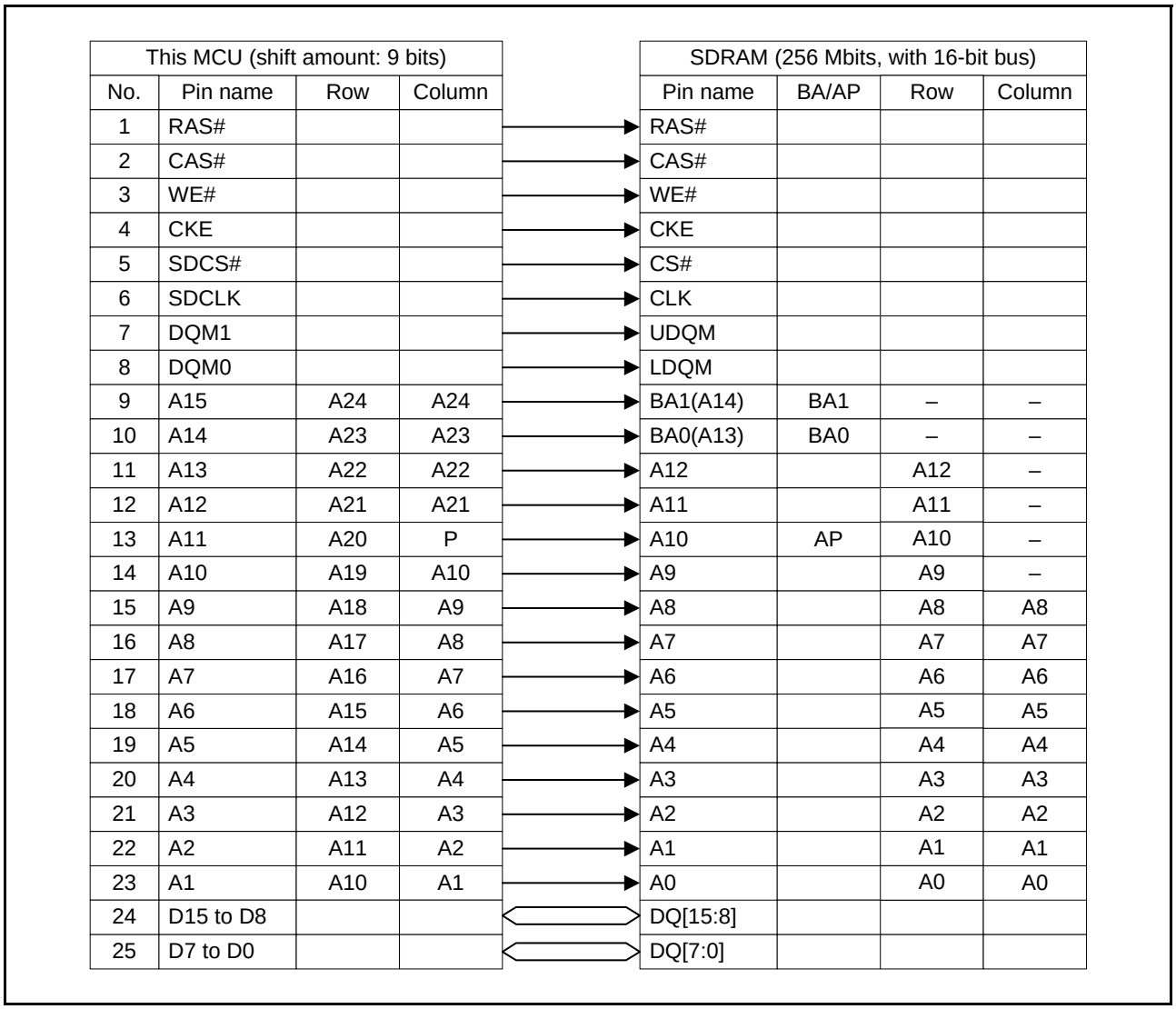
Figure 16.81 shows an example for connecting to a 512-Mbit SDRAM with 13-bit row address, 10-bit column address and 16-bit bus.



This MCU (shift amount: 10 bits)				SDRAM (512 Mbits, with 16-bit bus)			
No.	Pin name	Row	Column	Pin name	BA/AP	Row	Column
1	RAS#			RAS#			
2	CAS#			CAS#			
3	WE#			WE#			
4	CKE			CKE			
5	SDCS#			CS#			
6	SDCLK			CLK			
7	DQM1			UDQM			
8	DQM0			LDQM			
9	A15	A25	A25	BA1(A14)	BA1	–	–
10	A14	A24	A24	BA0(A13)	BA0	–	–
11	A13	A23	A23	A12		A12	–
12	A12	A22	A22	A11		A11	–
13	A11	A21	P	A10	AP	A10	–
14	A10	A20	A10	A9		A9	A9
15	A9	A19	A9	A8		A8	A8
16	A8	A18	A8	A7		A7	A7
17	A7	A17	A7	A6		A6	A6
18	A6	A16	A6	A5		A5	A5
19	A5	A15	A5	A4		A4	A4
20	A4	A14	A4	A3		A3	A3
21	A3	A13	A3	A2		A2	A2
22	A2	A12	A2	A1		A1	A1
23	A1	A11	A1	A0		A0	A0
24	D15 to D8			DQ[15:8]			
25	D7 to D0			DQ[7:0]			

Figure 16.81 SDRAM Connection Example (512-Mbit × 1, with 16-Bit Bus)

Figure 16.82 shows an example for connecting to a 256-Mbit SDRAM with 13-bit row address, 9-bit column address and 16-bit bus.



This MCU (shift amount: 9 bits)				SDRAM (256 Mbits, with 16-bit bus)			
No.	Pin name	Row	Column	Pin name	BA/AP	Row	Column
1	RAS#			RAS#			
2	CAS#			CAS#			
3	WE#			WE#			
4	CKE			CKE			
5	SDCS#			CS#			
6	SDCLK			CLK			
7	DQM1			UDQM			
8	DQM0			LDQM			
9	A15	A24	A24	BA1(A14)	BA1	–	–
10	A14	A23	A23	BA0(A13)	BA0	–	–
11	A13	A22	A22	A12		A12	–
12	A12	A21	A21	A11		A11	–
13	A11	A20	P	A10	AP	A10	–
14	A10	A19	A10	A9		A9	–
15	A9	A18	A9	A8		A8	A8
16	A8	A17	A8	A7		A7	A7
17	A7	A16	A7	A6		A6	A6
18	A6	A15	A6	A5		A5	A5
19	A5	A14	A5	A4		A4	A4
20	A4	A13	A4	A3		A3	A3
21	A3	A12	A3	A2		A2	A2
22	A2	A11	A2	A1		A1	A1
23	A1	A10	A1	A0		A0	A0
24	D15 to D8			DQ[15:8]			
25	D7 to D0			DQ[7:0]			

Figure 16.82 SDRAM Connection Example (256-Mbit × 1, with 16-Bit Bus)

16.6.15 Restrictions

(1) Prohibition of Access that Spans Areas of External Address Space

Single access that spans two areas of the external address space is prohibited, and operation of such an access is not guaranteed. Setting must be made so that two areas are not accessed at the same time by a single word or longword access.
Single access that spans two areas of the address space is also prohibited in the EXDMAC block transfer in single address mode or cluster transfer, and operation of such an access is not guaranteed. Setting must be made so that two areas are not accessed at the same time by a single access in the EXDMAC block transfer in single address mode or cluster transfer.

(2) Restrictions on RMPA and String-Manipulation Instructions

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

(3) Low Power Consumption State

In all-module clock stop mode, software standby mode, and deep software standby mode, auto-refresh operation is not available since the clock supply to SDRAMC is stopped. To retain the data in the SDRAM when the SDRAM is externally connected, use the self-refresh function. For the procedure for transition to and recovery from self-refresh mode, refer to 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 16.22.

16.7.1.2 Timeout

When the timeout detection enable bit (TOEN) in the bus error monitoring enable register (BEREN) is set to 1, bus access that is not completed within 768 cycles leads to a timeout error.

- CS areas (CS0 to CS7): Bus access is not completed (the WAIT# signal is not negated) within 768 external bus clock (BCLK) cycles from the start of the access.
  Once a timeout error occurs, accesses from the bus master are rejected for 256 BCLK cycles. If multiple external bus accesses are generated with a single request from the bus master during the transfer, the bus accesses cannot be stopped by a timeout. At this time, timeout errors may occur repeatedly.
- Internal peripheral buses (2 and 3): Bus access is not completed within 768 peripheral module clock (PCLKB) cycles from the start of the access. In this MCU, a timeout error does not occur.
  Once a timeout error occurs, accesses from the bus master are rejected for 256 PCLKB cycles.
- Internal peripheral buses (4 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. If the MSTPB15 bit is cleared, set the BEREN.TOEN bit to 1.
- Internal peripheral bus (6): Bus access is not completed within 768 FlashIF clock (FCLK) cycles from the start of the access.
  Once a timeout error occurs, accesses from the bus master are rejected for 256 FCLK cycles. In this MCU, a timeout error does not occur.
- Internal expansion bus: Bus access is not completed within 768 ICLK cycles from the start of the access.
  Once a timeout error occurs, accesses from the bus master are rejected for 256 ICLK 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 16.22 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 16.22 Types of Bus Errors

	Type of Area		Type of Error			
			Illegal Address Access		Timeout	
	On-Chip ROM		On-Chip ROM		On-Chip ROM	
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	Internal peripheral bus 5		Δ		—	
0010 0000h to 0011 FFFFh	Internal peripheral bus 6	Reserved area	—	✓	—	—
0012 0000h to 007F FFFFh			Δ	✓	—	—
0080 0000h to 00FF FFFFh	Reserved area		—	—	—	—
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 6FFF FFFFh	Reserved area		✓		—	
7000 0000h to 77FF FFFFh	Internal expansion bus (QSPI area)		Δ		[TO]	
7800 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: Availability and capacity of RAM, data flash memory, code flash memory to be included differ depending on the product. For details, refer to section 53, RAM, section 55, Flash Memory (FLASH).

16.8 Interrupt

16.8.1 Interrupt Source

An illegal address access error or detection of a timeout leads to a bus error signal for the interrupt controller.

Table 16.23 Interrupt Source

Name	Interrupt Source	DTC Trigger	DMAC Trigger
BUSERR	Illegal address access error or timeout	Not possible	Not possible

17. Memory-Protection Unit (MPU)

17.1 Overview

The RXv3 CPU incorporates a memory-protection unit that checks the addresses of CPU access to the overall address space (0000 0000h to FFFF FFFFh).

Access-control information can be set for up to eight regions, and permission for access to each region is in accord with this information. The default response to the detection of access to a region where permission has not been set is the generation of a memory-protection error.

The supported access-control information for the individual regions consists of permission to read, permission to write, and permission to execute. This access-control information is effective when the processor mode of the CPU is user mode. Memory protection is not applied when the CPU is in supervisor mode.

Table 17.1 lists the specifications of the memory-protection unit, and Figure 17.1 shows a block diagram of the memory-protection unit.

Table 17.1 Specifications of Memory Protection

Specifications	Description
Region to be covered by memory protection and processor mode	0000 0000h to FFFF FFFFh (in user mode) No memory protection in supervisor mode
Number of regions	8
Page size (smallest unit of protection)	16 bytes
Specifying addresses of individual regions	Setting the page numbers where regions start and end
Setting to make memory protection effective or ineffective in individual regions	A V (valid) bit in each region-n end page number register (REPAGEn) makes the settings effective or ineffective for the corresponding region (n = 0 to 7).
Access-control information settings for individual regions	Instruction execution: Permission to execute Operand access: Permission to read, permission to write
Start of memory-protection operations	After the memory-protection unit has been enabled, access monitoring starting up with the transition to user mode.
Memory-protection error processing	Generation of access exceptions
Addresses where memory-protection errors are generated	Address in instruction execution: The PC value is preserved on the stack. Address in operand access: The address is stored in the data memory-protection error address register (MPDEA).
Determining the reasons for memory-protection errors	The memory-protection error status register (MPESTS) holds indicators of the reason.
Background region setting	Access-control information can be set for the background region (the whole address space).
Processing where regions overlap	The access-control information for access to an overlap between regions is the logical OR of the attributes for the given regions, and permission is given priority.

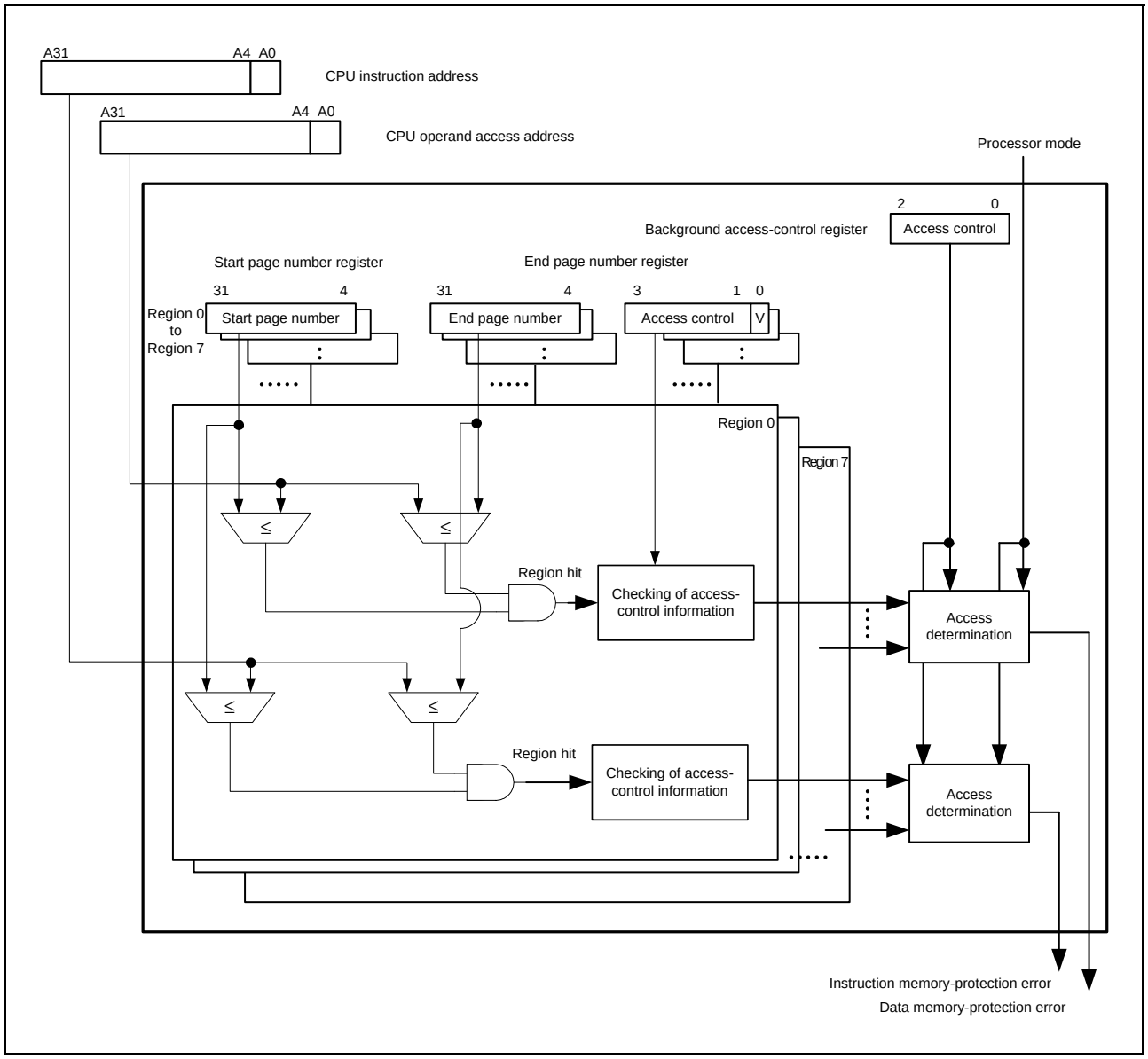


Figure 17.1 Block Diagram of the Memory-Protection Unit

17.1.1 Types of Access Control

There are three types of access control information: permission for instruction execution, permission to read operands, and permission to write operands. Violations of these types of access control are only detected when programs are running in user mode. Violations are not detected when programs are running in supervisor mode.

17.1.2 Regions for Access Control

Up to eight regions for access control are definable. Settings of the range of memory for each access-control region are made in the corresponding region-n start page number register (RSPAGEn) and region-n end page number register (REPAGEn), where n = 0 to 7.
The minimum unit for control of access is the “page”, by which the address space is divided into 16-byte units. The 28 higher-order bits ([31:4]) of the address [31:0] bits correspond to the page number.
The REPAGEn register specifies the access-control information for each area and whether the area is enabled or not.

17.1.3 Background Region

“Background region” refers to the whole address space (0000 0000h to FFFF FFFFh). Access-control information for the background region is set in the background-region access-control register (MPBAC). In contrast to the access-control information for the eight individual regions, protection information for the background region is effective as long as memory protection is enabled (the MPEN bit in the MPEN register is 1).

17.1.4 Overlap between Regions

In cases of overlap between multiple regions, the access-control information becomes the logical OR of the access-control bits for the overlapping regions (including the background region), with permission given priority.

17.1.5 Instructions and Data that Span Regions

Operations in response to the detection of memory-protection errors when instructions or data span regions for which different access-control settings have been made are undefined. Ensure that instructions and data do not span regions for which different access-control settings have been made.

17.2 Register Descriptions

17.2.1 Region-n Start Page Number Register (RSPAGEn) (n = 0 to 7)

Address(es): RSPAGE0 0008 6400h, RSPAGE1 0008 6408h, RSPAGE2 0008 6410h, RSPAGE3 0008 6418h, RSPAGE4 0008 6420h, RSPAGE5 0008 6428h, RSPAGE6 0008 6430h, RSPAGE7 0008 6438h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RSPN[27:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RSPN[27:0]												—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b4	RSPN[27:0]	Region-Start Page Number	Page number where the region starts, for use in region determination	R/W

RSPN[27:0] Bits (Region-Start Page Number)

These bits specify the page number where the region starts.

17.2.2 Region-n End Page Number Register (REPAGEn) (n = 0 to 7)

Address(es): REPAGE0 0008 6404h, REPAGE1 0008 640Ch, REPAGE2 0008 6414h, REPAGE3 0008 641Ch, REPAGE4 0008 6424h, REPAGE5 0008 642Ch, REPAGE6 0008 6434h, REPAGE7 0008 643Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	REPN[27:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REPN[27:0]												UAC[2:0]			V
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	V	Valid Bit	0: Region setting invalid 1: Region setting valid	R/W
b3 to b1	UAC[2:0]	Access Control Bits in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R/W
b31 to b4	REPN[27:0]	Region-End Page Number	Page number where the region ends, for use in region determination	R/W

V Bit (Valid Bit)

This bit enables or disables the settings for the corresponding region.
This bit is set to 0 when the region invalidation operation register (MPOPI) invalidates all access-controlled areas.

UAC[2:0] Bits (Access Control Bits in User Mode)

These bits specify the access control in user mode.

REPN[27:0] Bits (Region-End Page Number)

These bits specify the page number where the region ends.
Specify a value that is greater than or equal to the page number where the corresponding region starts. The page specified by the region-end page number is part of the target region for memory protection.

17.2.3 Memory-Protection Enable Register (MPEN)

Address(es): 0008 6500h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Bit Name	Description	R/W
b0	MPEN	Memory-Protection Enable	0: The memory protection is disabled. 1: The memory protection is enabled.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

MPEN Bit (Memory-Protection Enable)

This bit enables or disables the memory protection.

After 1 has been written to this bit, address checking for memory protection by the CPU starts on the execution of a branch instruction (RTE or RTFI) that shifts operation to the user mode.

17.2.4 Background Access Control Register (MPBAC)

Address(es): 0008 6504h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3 to b1	UBAC[2:0]	Background Access Control Bits in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R/W
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

UBAC[2:0] Bits (Background Access Control Bits in User Mode)

These bits specify the background access control in user mode.

17.2.5 Memory-Protection Error Status-Clearing Register (MPECLR)

Address(es): 0008 6508h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Bit Name	Description	R/W
b0	CLR	Error Status-Clearing	[Reading] 0: Fixed value for reading [Writing] 0: Nothing is done. 1: The DRW, DMPER and IMPER bits in MPESTS are set to 0.	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CLR Bit (Error Status-Clearing)

This bit clears the data read/write bit (DRW), the data memory-protection error generation bit (DMPER), and the instruction memory-protection error generation bit (IMPER) in the memory-protection error status register (MPESTS) to 0.

17.2.6 Memory-Protection Error Status Register (MPESTS)

Address(es): 0008 650Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	DRW	DMPER	IMPER
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IMPER	Instruction Memory-Protection Error Generation	0: No instruction memory-protection error was generated. 1: Instruction memory-protection error was generated.	R
b1	DMPER	Data Memory-Protection Error Generation	0: No data memory-protection error was generated. 1: Data memory-protection error was generated.	R
b2	DRW	Data Read/Write	0: Data were read. 1: Data were written.	R
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

IMPER Bit (Instruction Memory-Protection Error Generation)

This bit indicates the state of memory-protection error generation by instruction execution.

Setting the error status-clearing bit (CLR) in the memory-protection error status-clearing register (MPECLR) to 1 clears this bit to 0.

DMPER Bit (Data Memory-Protection Error Generation)

This bit indicates the state of memory-protection error generation by operand access.

Setting the error status-clearing bit (CLR) in the memory-protection error status-clearing register (MPECLR) to 1 clears this bit to 0.

DRW Bit (Data Read/Write)

For a memory-protection error produced by operand access, this bit indicates the read/write attribute of the access operation. This bit is only valid when the DMPER bit is 1.

Setting the error status-clearing bit (CLR) in the memory-protection error status-clearing register (MPECLR) to 1 clears this bit to 0.

17.2.7 Data Memory-Protection Error Address Register (MPDEA)

Address(es): 0008 6514h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DEA[31:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DEA[31:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	DEA[31:0]	Data Memory-Protection Error Address	Data memory-protection error address	R

DEA[31:0] Bits (Data Memory-Protection Error Address)

These bits retain the address for which operand access generated a memory-protection error.

17.2.8 Region Search Address Register (MPSA)

Address(es): 0008 6520h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SA[31:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SA[31:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	SA[31:0]	Region Search Address	Address for region searching	R/W

SA[31:0] Bits (Region Search Address)

These bits specify the address for use in comparison with region-start addresses in the region-n start page number registers (RSPAGEn) and region-end addresses in the region-n end page number registers (REPAGEn).

17.2.9 Region Search Operation Register (MPOPS)

Address(es): 0008 6524h

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

Bit	Symbol	Bit Name	Description	R/W
b0	S	Region Search Operation	[Reading] 0: Fixed value for reading [Writing] 0: Nothing is done. 1: A region-search operation proceeds.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

S Bit (Region Search Operation)

Setting this bit to 1 makes the memory-protection unit perform a region-search operation. The address specified in the region search address register (MPSA) is compared with the address information for individual regions to search for a hitting region.

The result of searching is stored in the data-hit region bits (HITD[7:0]) of the data-hit region register (MHITD). Moreover, the logical OR of the respective access control bits for hitting regions is stored in the data-hit region access control bits (UHACD[2:0]) in user mode.

17.2.10 Region Invalidation Operation Register (MPOPI)

Address(es): 0008 6526h

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

Bit	Symbol	Bit Name	Description	R/W
b0	INV	Region Invalidate Start	[Reading] 0: Fixed value for reading [Writing] 0: Nothing is done. 1: All access-controlled areas are invalidated.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

INV Bit (Region Invalidate Start)

Setting this bit to 1 clears the valid (V) bits in all of the region-n end page number registers (REPAGEn) to 0. After a V bit is set to 0, all settings other than background access-control settings are invalid.

17.2.11 Instruction-Hit Region Register (MHITI)

Address(es): 0008 6528h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	HITI[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3 to b1	UHACI[2:0]	Instruction-Hit Region Access Control Bits in User Mode	b3 0: Reading prohibited 1: Read permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution is permitted.	R
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	HITI[7:0]	Instruction-Hit Region	When the instruction memory-protection error generation bit (MPESTS.IMPER) = 1, [b23:b16] = 0000 0000b indicates that attempted access to the background region led to an instruction memory-protection error. Other than above b23 0: Instruction memory-protection error was not generated in region 7. 1: Instruction memory-protection error was generated in region 7. b22 0: Instruction memory-protection error was not generated in region 6. 1: Instruction memory-protection error was generated in region 6. b21 0: Instruction memory-protection error was not generated in region 5. 1: Instruction memory-protection error was generated in region 5. b20 0: Instruction memory-protection error was not generated in region 4. 1: Instruction memory-protection error was generated in region 4. b19 0: Instruction memory-protection error was not generated in region 3. 1: Instruction memory-protection error was generated in region 3. b18 0: Instruction memory-protection error was not generated in region 2. 1: Instruction memory-protection error was generated in region 2. b17 0: Instruction memory-protection error was not generated in region 1. 1: Instruction memory-protection error was generated in region 1. b16 0: Instruction memory-protection error was not generated in region 0. 1: Instruction memory-protection error was generated in region 0.	R
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

UHACI[2:0] Bits (Instruction-Hit Region Access Control Bits in User Mode)

These bits hold the user-mode access control bits (REPAGEn.UAC[2:0]) for the region where the instruction memory-protection error was generated.

If the error was generated in an overlap between regions, the value stored here is the logical OR of the user-mode access control bits for the corresponding regions (including the background region).

HITI[7:0] Bits (Instruction-Hit Region)

These bits indicate the region where an instruction memory-protection error was generated. These bits are set to 0000 0000b in response to the generation of an instruction memory-protection error in the background region.

17.2.12 Data-Hit Region Register (MHITD)

Address(es): 0008 652Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	HITD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3 to b1	UHACD[2:0]	Data-Hit Region Access Control Bits in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b16	HITD[7:0]	Data-Hit Region	When the data memory-protection error generation bit (MPESTS.DMPER) = 1, [b23:b16] = 0000 0000b indicates that attempted access to the background region led to a data memory-protection error. Other than above b23 0: Neither a data memory-protection error nor a search hit was generated in region 7. 1: A data memory-protection error or search hit was generated in region 7. b22 0: Neither a data memory-protection error nor a search hit was generated in region 6. 1: A data memory-protection error or search hit was generated in region 6. b21 0: Neither a data memory-protection error nor a search hit was generated in region 5. 1: A data memory-protection error or search hit was generated in region 5. b20 0: Neither a data memory-protection error nor a search hit was generated in region 4. 1: A data memory-protection error or search hit was generated in region 4. b19 0: Neither a data memory-protection error nor a search hit was generated in region 3. 1: A data memory-protection error or search hit was generated in region 3. b18 0: Neither a data memory-protection error nor a search hit was generated in region 2. 1: A data memory-protection error or search hit was generated in region 2. b17 0: Neither a data memory-protection error nor a search hit was generated in region 1. 1: A data memory-protection error or search hit was generated in region 1. b16 0: Neither a data memory-protection error nor a search hit was generated in region 0. 1: A data memory-protection error or search hit was generated in region 0.	R
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

UHACD[2:0] Bits (Data-Hit Region Access Control Bits in User Mode)

These bits hold the user-mode access control bits (REPAGEn.UAC[2:0]) that have been set for the region where a data memory-protection error was generated or the region that produced a hit in region searching.
When an error is generated in an overlap between regions or a hit was generated in region searching, the value stored here is the logical OR of the user-mode access control bits for the corresponding regions (including the background region).

HITD[7:0] Bits (Data-Hit Region)

These bits indicate the region where a data memory-protection error was generated or the region that produced a hit in a region search. These bits are set to 0000 0000b for a data memory-protection error generated in the background region.

Note: When access to a register of memory protection unit in user mode generates a data memory-protection error, the value in this register is set to 0000 0000h.

17.3 Functions

17.3.1 Memory Protection

Memory protection means monitoring, in accord with the access-control information that has been set for the individual access-control regions and the background region, whether or not access by programs running in user mode violates the access-control settings. The memory-protection unit notifies the CPU of access-control violations (or memory-protection errors) when they are detected, causing the CPU to start access-exception processing.
Memory protection is enabled by setting the memory-protection enable (MPEN) bit in the memory-protection enable (MPEN) register to 1.
An instruction memory-protection error is generated on detection of an instruction-execution violation and a data memory-protection error is generated on detection of an operand-access reading or writing violation. Operand access that leads to a data memory-protection error is not actually executed.

17.3.2 Region Search

Region search means enquiry as to which of the eight specified access regions was “hit” and how the access-control information (permission to execute, to read, and to write) is set.
When the region search operation (S) bit in the region-search operation (MPOPS) register is set to 1, the address specified in the region search address (MPSA) register is compared with the addresses for the individual regions. After a region search is executed, the data-hit region register (MHITD) indicates the logical OR of the access-control information for the region which was “hit” and for the other regions.

17.3.3 Protection of Registers Related to the Memory-Protection Unit

Registers related to the memory-protection unit are not accessible through means of access other than operand access by the CPU (i.e. by instruction fetching or DMA). The registers related to the memory-protection unit are only accessible in supervisor mode. Attempted access to registers related to the memory-protection unit in user mode through operand access by the CPU leads to a data memory-protection error regardless of whether or not memory protection is in effect at the given location.

17.3.4 Flow for Determination of Access by the Memory-Protection Function

Figure 17.2 shows the flow of determination in the case of data access and Figure 17.3 shows the flow of determination in the case of instruction access.

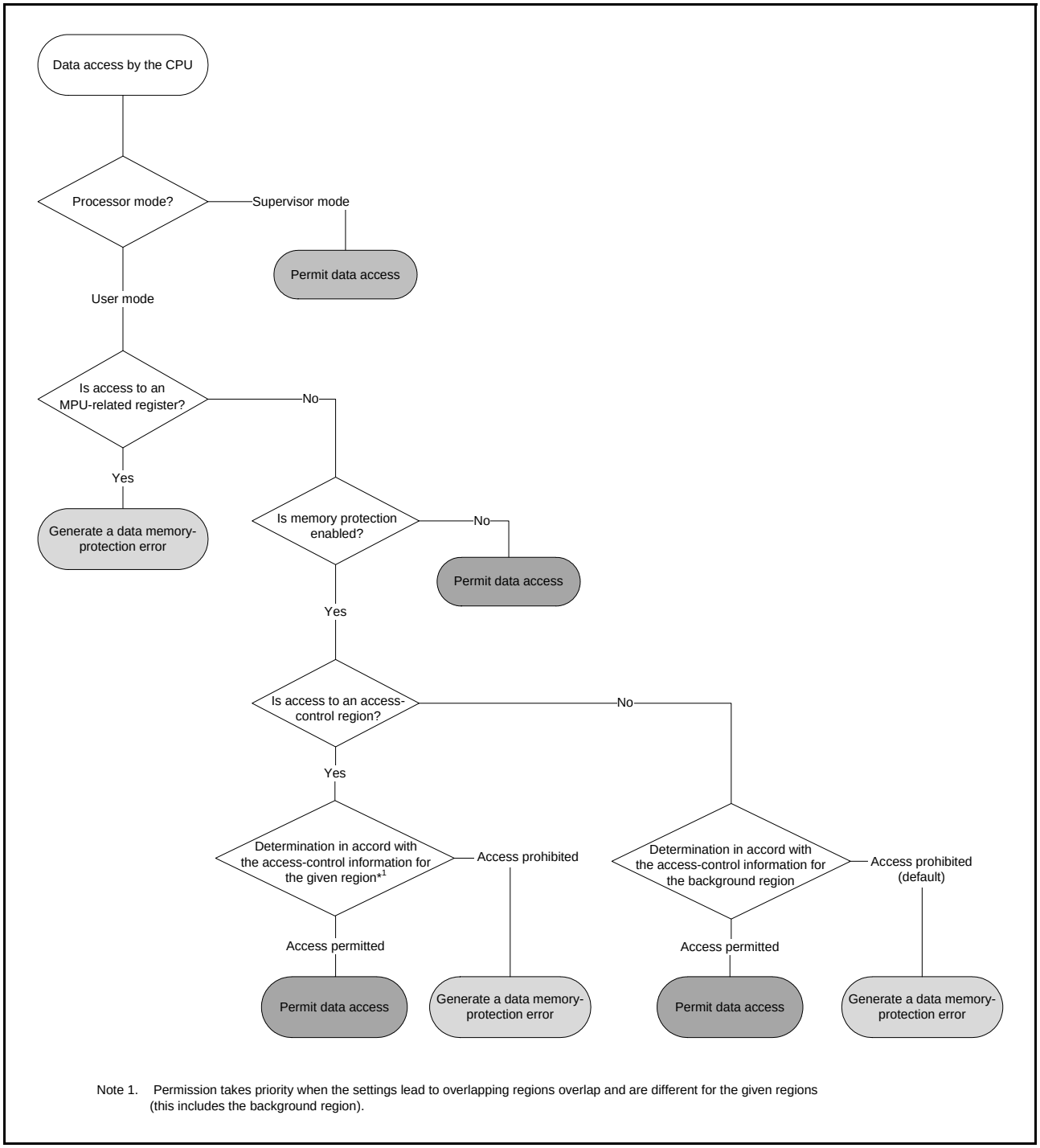
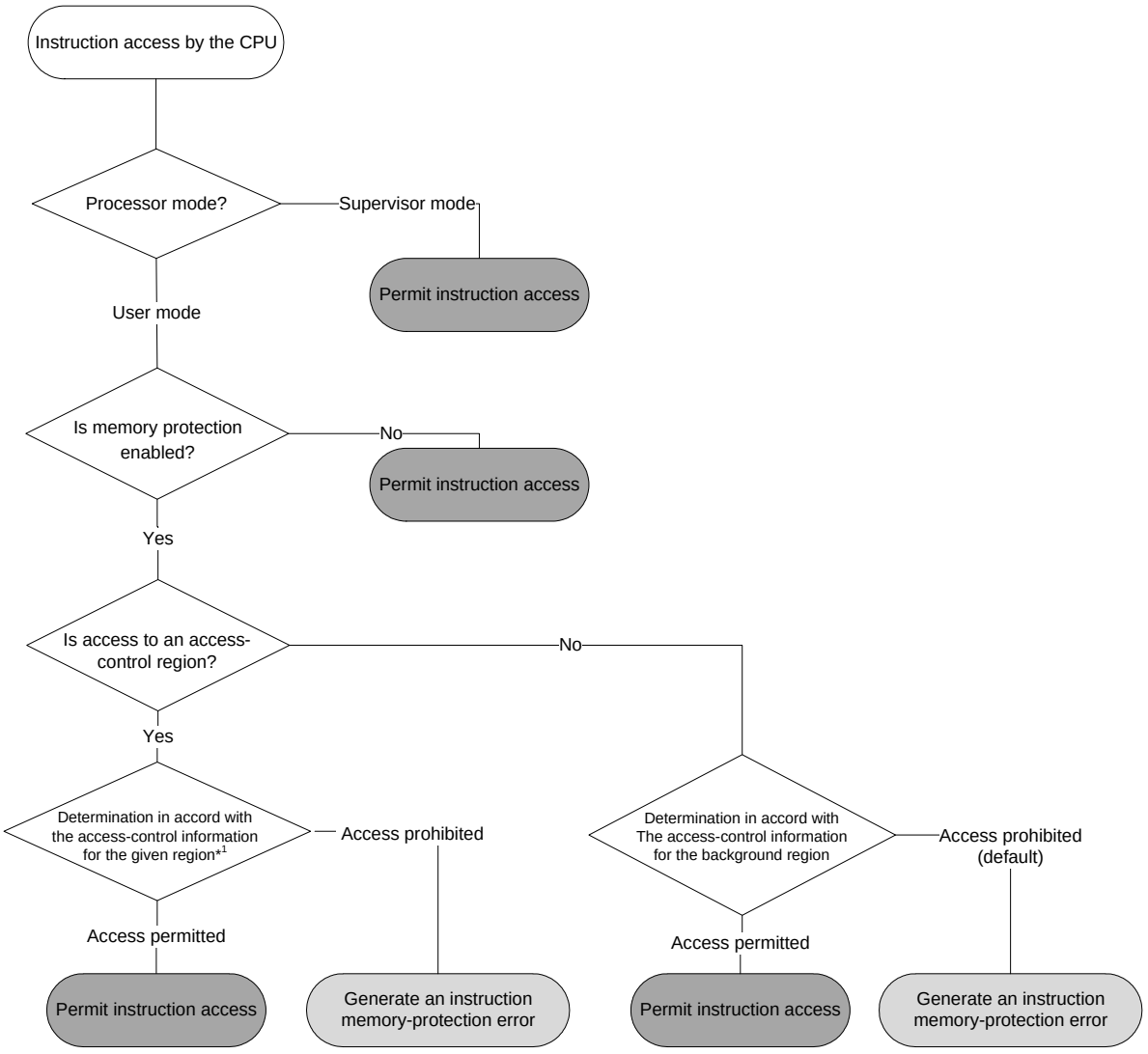


Figure 17.2 Flow of Determination for Data Access



Note 1. Permission takes priority when the settings lead to overlapping regions overlap and are not the same for the given regions (this includes the background region).

Figure 17.3 Flow of Determination for Instruction Access

17.4 Procedures for Using Memory Protection

17.4.1 Setting Access-Control Information

Access-control information for the various regions is set in supervisor mode.
Settings for up to eight access-control regions are made in the region-n start page number registers (RSPAGEn) and region-n end page number registers (REPAGEn), where n = 0 to 7.
Settings for the background access-control region are made in the background access-control register (MPBAC).

17.4.2 Enabling Memory Protection

Setting the memory-protection enable (MPEN) bit in the memory-protection enable (MPEN) register to 1 while operation is in supervisor mode enables memory protection.

17.4.3 Transition to User Mode

After updating the registers related to the memory-protection unit, read any of these registers and check that the settings have been made before the transition to user mode.
Either of the methods below can be used for the transition from supervisor mode to user mode.

- Set the processor mode setting (PM) bit in the copy of the processor status word (PSW) saved in the stack area to 1 (the setting for user mode) and then execute an RTE instruction.
- Set the PM bit in the backup processor status word (BPSW) to 1 and then execute an RTFI instruction.

Note: Using an MVTC or POPC instruction to write to the PSW.PM bit is invalid. Use an RTE or RTFI instruction to update the value of the PSW.PM bit.

The memory-protection unit starts checking instruction-execution access and operand access by the CPU on the transition to user mode.

17.4.4 Processing in Response to Memory-Protection Errors

The CPU starts access-exception processing on detection of a violation of protection set up by the access-control information (i.e. a memory-protection error). For details on CPU operations in access-exception processing, refer to section 14, Exception Handling.
To determine whether an instruction memory-protection error or data memory-protection error has been generated, check the values of the instruction memory-protection error generation (IMPER) and data memory-protection error generation (DMPER) bits in the memory-protection error status register (MPESTS) from within the exception-processing routine. After confirming the type of error, clear the memory-protection error status register (MPESTS) by writing 1 to the status clearing (CLR) bit in the memory-protection error status clearing register (MPECLR).

(1) When a data memory-protection error is generated

Access-exception processing by the CPU saves the address of the instruction that led to the memory-protection error on the stack. Furthermore, the address of the operand for which access led to a memory-protection error is stored in the data memory-protection error address register (MPDEA) and the region information for the region where the memory-protection error was generated is stored in the data-hit region register (MHITD).

- Violations of access control in access to valid regions 0 to 7

The data-hit region bits (MHITD.HITD[7:0]) with the same region number as the region where the error occurred is set to 1. In user mode, the logical “or” of the region access-control information for the location where the error occurred is set in the data-hit region access-control bits (MHITD.UHACD[2:0]).

- Violations of access control for the background region, besides access to outside valid regions 0 to 7

The data-hit region bits (MHITD.HITD[7:0]) are set to 0000 0000b. In user mode, the access-control information for the background region is set in the data-hit region access-control bits (MHITD.UHACD[2:0]).

Referring to this information can pinpoint the sources of errors.

(2) When an instruction memory-protection error is generated

Access-exception processing by the CPU saves the address of the instruction that led to the memory-protection error on the stack. Furthermore, the region information for the region where the memory-protection error was generated is stored in the instruction-hit region register (MHITI).

- Violations of access control in access to valid regions 0 to 7

The instruction-hit region bit (MHITD.HITI[7:0]) with the same region number as the region where the error occurred is set to 1. In user mode, the logical “or” of the region access-control information for the location where the error occurred is set in the instruction-hit region access-control bits (MHITI.UHACI[2:0]).

- Violations of access control for the background region, besides access to outside valid regions 0 to 7

The instruction-hit region bits (MHITI.HITI[7:0]) are set to 0000 0000b. In user mode, the access-control information for the background region is set in the instruction-hit region access-control bits (MHITI.UHACI[2:0]).

Referring to this information can pinpoint the sources of errors.

18. DMA Controller (DMACAb)

This MCU incorporates an 8-channel direct memory access controller (DMAC).
The DMAC is a module to transfer data without the CPU. When a DMA transfer request is generated, the DMAC transfers data stored at the transfer source address to the transfer destination address.

18.1 Overview

Table 18.1 lists the specifications of the DMAC, and Figure 18.1 shows a block diagram of the DMAC.

Table 18.1 Specifications of DMAC

Item		Description
Number of channels		8 (DMACm (m = 0 to 7))
Transfer space		4 Gbytes (0000 0000h to FFFF FFFFh excluding reserved areas)
Maximum transfer volume		64M data (Maximum number of transfers in block transfer mode: 1024 data × 65536 blocks)
DMA request source		• Request source selectable for each channel Software trigger Interrupt requests from peripheral modules or trigger input to external interrupt input pins*1
Channel priority		Channel 0 > Channel 1 > Channel 2 > Channel 3... > Channel 7 (Channel 0: Highest)
Transfer data	Single data	Bit length: 8, 16, 32 bits
	Block size	Number of data: 1 to 1024
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: 1024
	Block transfer mode	• One block data transfer by one DMA transfer request • Maximum settable block size: 1024 data
Selective functions	Extended repeat area function	• Function in which data can be transferred by repeating the address values in the specified range with the upper bit values in the transfer address register fixed • Area of 2 bytes to 128 Mbytes separately settable as extended repeat area for transfer source and destination
Interrupt request	Transfer end interrupt	Generated when the specified number of transfers is completed in normal transfer mode Generated when the specified repeat count of transfers is completed in repeat transfer mode Generated when the specified block count of transfers is completed in block transfer mode
	Transfer escape end interrupt	Generated when the repeat size of data transfer is completed or the extended repeat area overflows.
Event link function		An event link request is generated after each data transfer (for block transfer, after each block is transferred).
Power consumption reduction function		Module-stop state can be set.

Note 1. For details on DMA request sources, refer to Table 15.5, Interrupt Vector Table in section 15, Interrupt Controller (ICUE).

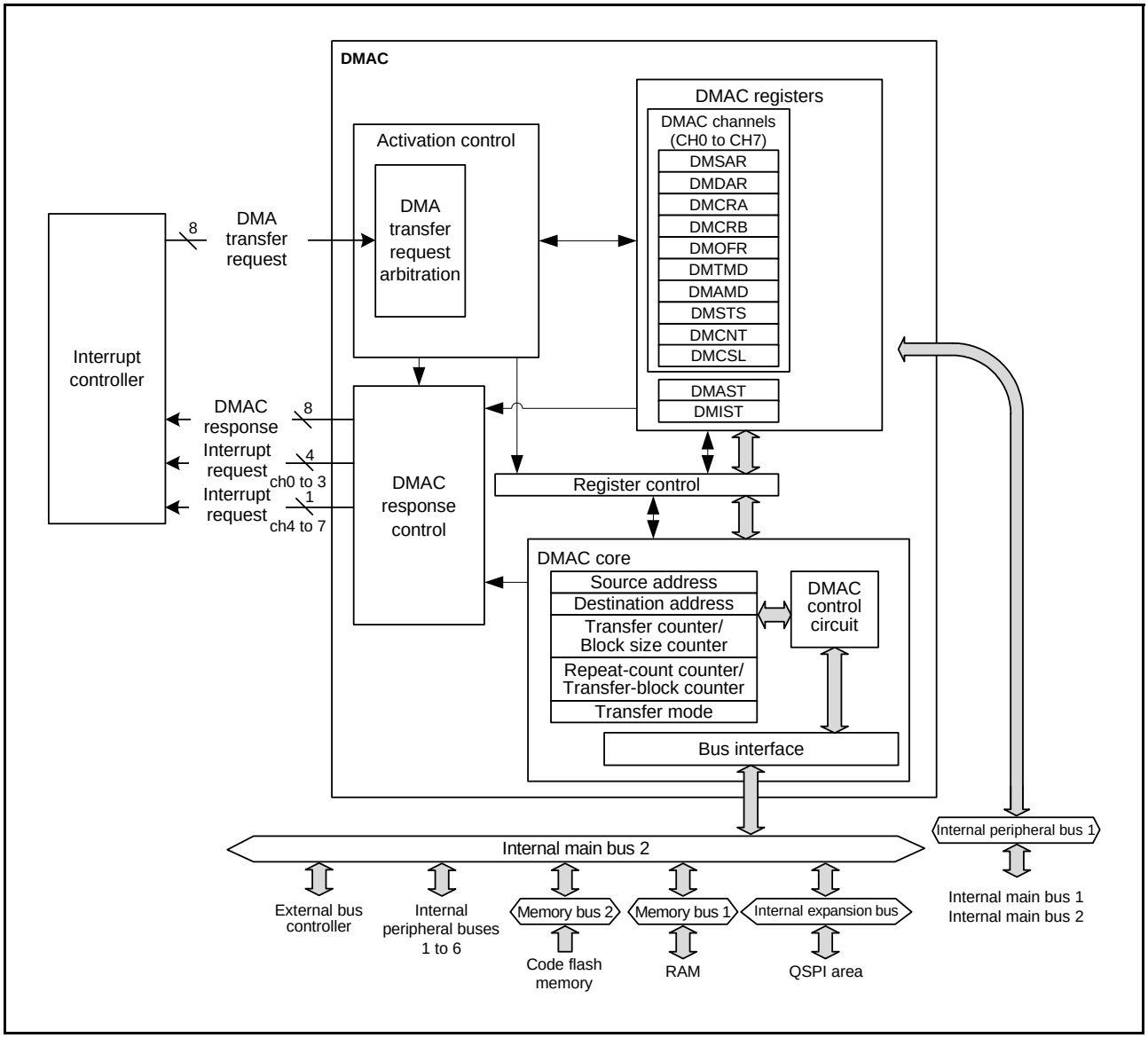


Figure 18.1 Block Diagram of DMAC

18.2 Register Descriptions

18.2.1 DMA Source Address Register (DMSAR)

Address(es): DMAC0.DMSAR 0008 2000h, DMAC1.DMSAR 0008 2040h, DMAC2.DMSAR 0008 2080h, DMAC3.DMSAR 0008 20C0h, DMAC4.DMSAR 0008 2100h, DMAC5.DMSAR 0008 2140h, DMAC6.DMSAR 0008 2180h, DMAC7.DMSAR 0008 21C0h

	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 FFFF FFFFh (4 Gbytes)	R/W

Set the DMSAR register while DMAC stops (DMAST.DMST bit = 0) or DMA transfer is disabled (DMCNT.DTE bit = 0).

18.2.2 DMA Destination Address Register (DMDAR)

Address(es): DMAC0.DMDAR 0008 2004h, DMAC1.DMDAR 0008 2044h, DMAC2.DMDAR 0008 2084h, DMAC3.DMDAR 0008 20C4h, DMAC4.DMDAR 0008 2104h, DMAC5.DMDAR 0008 2144h, DMAC6.DMDAR 0008 2184h, DMAC7.DMDAR 0008 21C4h

	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 FFFF FFFFh (4 Gbytes)	R/W

Set the DMDAR register while DMAC stops (DMAST.DMST bit = 0) or DMA transfer is disabled (DMCNT.DTE bit = 0).

18.2.3 DMA Transfer Count Register (DMCRA)

Address(es): DMAC0.DMCRA 0008 2008h, DMAC1.DMCRA 0008 2048h, DMAC2.DMCRA 0008 2088h, DMAC3.DMCRA 0008 20C8h, DMAC4.DMCRA 0008 2108h, DMAC5.DMCRA 0008 2148h, DMAC6.DMCRA 0008 2188h, DMAC7.DMCRA 0008 21C8h

- Normal transfer mode

DMCRAH

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
—	—	—	—	—	—										
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Value after reset:

DMCRAL

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	0	0	0

Value after reset:

- Repeat transfer mode, block transfer mode

DMCRAH

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
—	—	—	—	—	—										
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Value after reset:

DMCRAL

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	0	0	0	0	0	0	0	0	0

Value after reset:

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 the DMCRAH and DMCRAL registers in repeat transfer mode and block transfer mode.

(1) Normal Transfer Mode (DMACm.DMTMD.MD[1:0] Bits = 00b)

The DMCRAL register 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).

The DMCRAH register is not used in normal transfer mode. Write 0000h to the DMCRAH register.

(2) Repeat Transfer Mode (DMACm.DMTMD.MD[1:0] Bits = 01b)

The DMCRAH register specifies the repeat size and the DMCRAL register 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 the DMCRAH and DMCRAL registers.
Setting bits 15 to 10 in the DMCRAL register is invalid. Write 0 to these bits.
The value in the DMCRAL register is decremented by one each time data is transferred until it reaches 000h, at which the value in the DMCRAH register is loaded into the DMCRAL register.

(3) Block Transfer Mode (DMACm.DMTMD.MD[1:0] Bits = 10b)

The DMCRAH register specifies the block size and the DMCRAL register 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 the DMCRAH and DMCRAL registers.
Setting bits 15 to 10 in the DMCRAL register is invalid. Write 0 to these bits.
The value in the DMCRAL register is decremented by one each time data is transferred until it reaches 000h, at which the value in the DMCRAH register is loaded into the DMCRAL register.

18.2.4 DMA Block Transfer Count Register (DMCRB)

Address(es): DMAC0.DMCRB 0008 200Ch, DMAC1.DMCRB 0008 204Ch, DMAC2.DMCRB 0008 208Ch, DMAC3.DMCRB 0008 20CCh, DMAC4.DMCRB 0008 210Ch, DMAC5.DMCRB 0008 214Ch, DMAC6.DMCRB 0008 218Ch, DMAC7.DMCRB 0008 21CCh

	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
b15 to b0	Specifies the block count or repeat count of transfers.	0001h to FFFFh (1 to 65535) 0000h (65536)	R/W

This register specifies the transfer block count in block transfer mode and the repeat count in repeat transfer mode.
In repeat transfer mode, the value is decremented by one when the final data of one repeat size is transferred.
In block transfer mode, the value is decremented by one when the final data of one block size is transferred.
In normal transfer mode, the DMCRB register is not used. The setting is invalid.

18.2.5 DMA Transfer Mode Register (DMTMD)

Address(es): DMAC0.DMTMD 0008 2010h, DMAC1.DMTMD 0008 2050h, DMAC2.DMTMD 0008 2090h, DMAC3.DMTMD 0008 20D0h, DMAC4.DMTMD 0008 2110h, DMAC5.DMTMD 0008 2150h, DMAC6.DMTMD 0008 2190h, DMAC7.DMTMD 0008 21D0h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	DCTG[1:0]	Transfer Request Source Select	b1 b0 0 0: Software 0 1: Interrupts*1 from peripheral modules or external interrupt input pins 1 0: Setting prohibited 1 1: Setting prohibited	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	SZ[1:0]	Transfer Data Size Select	b9 b8 0 0: 8 bits 0 1: 16 bits 1 0: 32 bits 1 1: Setting prohibited	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13, b12	DTS[1:0]	Repeat Area Select	b13 b12 0 0: The destination is specified as the repeat area or block area. 0 1: The source is specified as the repeat area or block area. 1 0: The repeat area or block area is not specified. 1 1: Setting prohibited	R/W
b15, b14	MD[1:0]	Transfer Mode Select	b15 b14 0 0: Normal transfer 0 1: Repeat transfer 1 0: Block transfer 1 1: Setting prohibited	R/W

Note 1. DMA request source is selected using the ICU.DMRSRm registers. For details on DMA request sources, refer to Table 15.5, Interrupt Vector Table in section 15, Interrupt Controller (ICUE).

DTS[1:0] Bits (Repeat Area Select)

These bits select either the source or destination as the repeat area in repeat or block transfer mode. In normal transfer mode, setting these bits is invalid.

18.2.6 DMA Interrupt Setting Register (DMINT)

Address(es): DMAC0.DMINT 0008 2013h, DMAC1.DMINT 0008 2053h, DMAC2.DMINT 0008 2093h, DMAC3.DMINT 0008 20D3h, DMAC4.DMINT 0008 2113h, DMAC5.DMINT 0008 2153h, DMAC6.DMINT 0008 2193h, DMAC7.DMINT 0008 21D3h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DARIE	Destination Address Extended Repeat Area Overflow Interrupt Enable	0: Disables an interrupt request for an extended repeat area overflow on the destination address 1: Enables an interrupt request for an extended repeat area overflow on the destination address	R/W
b1	SARIE	Source Address Extended Repeat Area Overflow Interrupt Enable	0: Disables an interrupt request for an extended repeat area overflow on the source address 1: Enables an interrupt request for an extended repeat area overflow on the source address	R/W
b2	RPTIE	Repeat Size End Interrupt Enable	0: Disables the repeat size end interrupt request. 1: Enables the repeat size end interrupt request.	R/W
b3	ESIE	Transfer Escape End Interrupt Enable	0: Disables the transfer escape end interrupt request. 1: Enables the transfer escape end interrupt request.	R/W
b4	DTIE	Transfer End Interrupt Enable	0: Disables the transfer end interrupt request. 1: Enables the transfer end interrupt request.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DARIE Bit (Destination Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the destination address occurs while this bit is set to 1, the DMCNT.DTE bit is set to 0. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that an interrupt by an extended repeat area overflow on the destination address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 in the DMACm.DMCNT.DTE bit of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the destination address, this bit is ignored.

SARIE Bit (Source Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the source address occurs while this bit is set to 1, the DMCNT.DTE bit is set to 0. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that an interrupt by an extended repeat area overflow on the source address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 in the DMACm.DMCNT.DTE bit of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the source address, this bit is ignored.

RPTIE Bit (Repeat Size End Interrupt Enable)

When this bit is set to 1 in repeat transfer mode, the DMCNT.DTE bit is set to 0 after completion of a 1-repeat size data transfer. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DMTMD.DTS[1:0] bits are 10b (= repeat area or block area is not specified).

When this bit is set to 1 in block transfer mode, the DMCNT.DTE bit is set to 0 after completion of a 1-block data transfer in the same way as repeat transfer mode. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DMTMD.DTS[1:0] bits 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 DMSTS.ESIF flag is set to 1 with this bit set to 1. The transfer escape end interrupt is cleared by setting this bit or the DMSTS.ESIF flag to 0.

DTIE Bit (Transfer End Interrupt Enable)

This bit enables or disables the transfer end interrupt request to be generated on completion of a specified number of data transfers.

The transfer end interrupt is generated when the DMSTS.DTIF flag is set to 1 with this bit set to 1. The transfer end interrupt is cleared by setting this bit or the DMSTS.DTIF flag to 0.

18.2.7 DMA Address Mode Register (DMAMD)

Address(es): DMAC0.DMAMD 0008 2014h, DMAC1.DMAMD 0008 2054h, DMAC2.DMAMD 0008 2094h, DMAC3.DMAMD 0008 20D4h, DMAC4.DMAMD 0008 2114h, DMAC5.DMAMD 0008 2154h, DMAC6.DMAMD 0008 2194h, DMAC7.DMAMD 0008 21D4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SM[1:0]		—	SARA[4:0]					DM[1:0]		—	DARA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	DARA[4:0]	Destination Address Extended Repeat Area	Specifies the extended repeat area on the destination address. For details on the settings, refer to Table 18.2.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7, b6	DM[1:0]	Destination Address Update Mode	b7 b6 0 0: Destination address is fixed. 0 1: Offset addition*1 1 0: Destination address is incremented. 1 1: Destination address is decremented.	R/W
b12 to b8	SARA[4:0]	Source Address Extended Repeat Area	Specifies the extended repeat area on the source address. For details on the settings, refer to Table 18.2.	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b15, b14	SM[1:0]	Source Address Update Mode	b15 b14 0 0: Source address is fixed. 0 1: Offset addition*1 1 0: Source address is incremented. 1 1: Source address is decremented.	R/W

Note 1. Offset addition can be specified only for DMAC0.

DARA[4:0] Bits (Destination Address Extended Repeat Area)

These bits specify the extended repeat area on the destination address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 Mbytes. That is, the interval between settings is any power of two. When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer destination, do not specify the extended repeat area on the destination address. When repeat transfer or block transfer is selected, and when DMACm.DMTMD.DTS[1:0] = 00b (the transfer destination is specified as the repeat area or block area), write 00000b in the DARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the DMINT.DARIE bit set to 1. Table 18.2 lists the settings and the corresponding extended repeat areas.

DM[1:0] Bits (Destination Address Update Mode)

These bits select the mode of updating the destination address.

When increment is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, and 10b, the destination address is incremented by 1, 2, and 4, respectively.

When decrement is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, and 10b, the destination address is decremented by 1, 2, and 4, respectively.

When offset addition is selected, the offset specified by the DMAC0.DMOFR register is added to the address. Offset addition can be specified only for DMAC0.

SARA[4:0] Bits (Source Address Extended Repeat Area)

These bits specify the extended repeat area on the source address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 Mbytes.

When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer source, do not specify the extended repeat area on the source address. When repeat transfer or block transfer is selected, and when DMACm.DMTMD.DTS[1:0] = 01b (the transfer source is specified as the repeat area or block area), write 00000b in the SARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the DMINT.SARIE bit set to 1. Table 18.2 lists the settings and the corresponding extended repeat areas.

SM[1:0] Bits (Source Address Update Mode)

These bits select the mode of updating the source address.

When increment is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, and 10b, the source address is incremented by 1, 2, and 4, respectively.

When decrement is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, and 10b, the source address is decremented by 1, 2, and 4, respectively.

When offset addition is selected, the offset specified by the DMAC0.DMOFR register is added to the address.

Offset addition can be specified only for DMAC0.

Table 18.2 SARA[4:0] or DARA[4:0] Settings and Corresponding Repeat Areas

SARA[4:0] or DARA[4:0]	Extended Repeat Area
00000b	Not specified
00001b	2 bytes specified as extended repeat area by the lower 1 bit of the address
00010b	4 bytes specified as extended repeat area by the lower 2 bits of the address
00011b	8 bytes specified as extended repeat area by the lower 3 bits of the address
00100b	16 bytes specified as extended repeat area by the lower 4 bits of the address
00101b	32 bytes specified as extended repeat area by the lower 5 bits of the address
00110b	64 bytes specified as extended repeat area by the lower 6 bits of the address
00111b	128 bytes specified as extended repeat area by the lower 7 bits of the address
01000b	256 bytes specified as extended repeat area by the lower 8 bits of the address
01001b	512 bytes specified as extended repeat area by the lower 9 bits of the address
01010b	1 Kbyte specified as extended repeat area by the lower 10 bits of the address
01011b	2 Kbytes specified as extended repeat area by the lower 11 bits of the address
01100b	4 Kbytes specified as extended repeat area by the lower 12 bits of the address
01101b	8 Kbytes specified as extended repeat area by the lower 13 bits of the address
01110b	16 Kbytes specified as extended repeat area by the lower 14 bits of the address
01111b	32 Kbytes specified as extended repeat area by the lower 15 bits of the address
10000b	64 Kbytes specified as extended repeat area by the lower 16 bits of the address
10001b	128 Kbytes specified as extended repeat area by the lower 17 bits of the address
10010b	256 Kbytes specified as extended repeat area by the lower 18 bits of the address
10011b	512 Kbytes specified as extended repeat area by the lower 19 bits of the address
10100b	1 Mbyte specified as extended repeat area by the lower 20 bits of the address
10101b	2 Mbytes specified as extended repeat area by the lower 21 bits of the address
10110b	4 Mbytes specified as extended repeat area by the lower 22 bits of the address
10111b	8 Mbytes specified as extended repeat area by the lower 23 bits of the address
11000b	16 Mbytes specified as extended repeat area by the lower 24 bits of the address
11001b	32 Mbytes specified as extended repeat area by the lower 25 bits of the address
11010b	64 Mbytes specified as extended repeat area by the lower 26 bits of the address
11011b	128 Mbytes specified as extended repeat area by the lower 27 bits of the address
11100b to 11111b	Setting prohibited.

18.2.8 DMA Offset Register (DMOFR)

Address(es): DMAC0.DMOFR 0008 2018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the offset when offset addition is selected as the address update mode for transfer source or destination.	0000 0000h to 00FF FFFFh (0 bytes to (16 M – 1) bytes) FF00 0000h to FFFF FFFFh (–16 Mbytes to –1 byte)	R/W

Write to this register while the DMAC operation is stopped or DMA transfer is disabled (not during data transfer). Setting bits 31 to 25 is invalid; a value of bit 24 is extended to bits 31 to 25. Reading DMOFR returns the extended value.

18.2.9 DMA Transfer Enable Register (DMCNT)

Address(es): DMAC0.DMCNT 0008 201Ch, DMAC1.DMCNT 0008 205Ch, DMAC2.DMCNT 0008 209Ch, DMAC3.DMCNT 0008 20DCh, DMAC4.DMCNT 0008 211Ch, DMAC5.DMCNT 0008 215Ch, DMAC6.DMCNT 0008 219Ch, DMAC7.DMCNT 0008 21DCh

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

Bit	Symbol	Bit Name	Description	R/W
b0	DTE	DMA Transfer Enable	0: Disables DMA transfer. 1: Enables DMA transfer.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DTE Bit (DMA Transfer Enable)

When the DMAST.DMST bit is set to 1 (DMAC module start) and this bit is set to 1 (DMA transfer is enabled), DMA transfer can be started for the corresponding channel.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When 0 is written to this bit.
- When the specified total volume of data transfer is completed.
- When DMA transfer is stopped by the repeat size end interrupt.
- When DMA transfer is stopped by the extended repeat area overflow interrupt.

18.2.10 DMA Software Start Register (DMREQ)

Address(es): DMAC0.DMREQ 0008 201Dh, DMAC1.DMREQ 0008 205Dh, DMAC2.DMREQ 0008 209Dh, DMAC3.DMREQ 0008 20DDh, DMAC4.DMREQ 0008 211Dh, DMAC5.DMREQ 0008 215Dh, DMAC6.DMREQ 0008 219Dh, DMAC7.DMREQ 0008 21DDh

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

Bit	Symbol	Bit Name	Description	R/W
b0	SWREQ	DMA Software Start	0: DMA transfer is not requested. 1: DMA transfer is requested.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	CLRS	DMA Software Start Bit Auto Clear Select	0: SWREQ bit is cleared after DMA transfer is started by software. 1: SWREQ bit is not cleared after DMA transfer is started by software.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SWREQ Bit (DMA Software Start)

When 1 is written to this bit, a DMA transfer request is generated. After DMA transfer is started in response to the request, this bit is set to 0 if the CLRS bit is set to 0. This bit is not set to 0 while the CLRS bit is set to 1. In this case, a DMA transfer request can be issued again after completion of a transfer.

Note that, however, setting this bit is valid and DMA transfer by software is enabled only when the DMTMD.DCTG[1:0] bits are set to 00b (DMA request source is software).

Setting this bit is invalid when the DMTMD.DCTG[1:0] bits are set to a value other than 00b.

To start DMA transfer by software with the CLRS bit being 0, ensure that the SWREQ bit is 0, and then write 1 to the SWREQ bit.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When a DMA transfer request by software is accepted and DMA transfer is started while the CLRS bit is set to 0 (the SWREQ bit is cleared after DMA transfer is started by software).
- When 0 is written to this bit.

CLRS Bit (DMA Software Start Bit Auto Clear Select)

This bit specifies whether to clear the SWREQ bit to 0 after DMA transfer is started in response to the DMA transfer request generated by setting the SWREQ bit to 1. With this bit set to 0, the SWREQ bit is set to 0 after DMA transfer is started. With this bit set to 1, the SWREQ bit is not set to 0. In this case, a DMA transfer request can be issued again after completion of a transfer.

18.2.11 DMA Status Register (DMSTS)

Address(es): DMAC0.DMSTS 0008 201Eh, DMAC1.DMSTS 0008 205Eh, DMAC2.DMSTS 0008 209Eh, DMAC3.DMSTS 0008 20DEh, DMAC4.DMSTS 0008 211Eh, DMAC5.DMSTS 0008 215Eh, DMAC6.DMSTS 0008 219Eh, DMAC7.DMSTS 0008 21DEh

	b7	b6	b5	b4	b3	b2	b1	b0
	ACT	—	—	DTIF	—	—	—	ESIF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESIF	Transfer Escape End Interrupt Flag	0: A transfer escape end interrupt has not been generated. 1: A transfer escape end interrupt has been generated.	R/W*1
b3 to b1	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b4	DTIF	Transfer End Interrupt Flag	0: A transfer end interrupt has not been generated. 1: A transfer end interrupt has been generated.	R/W*1
b6, b5	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	ACT	DMA Active Flag	0: DMAC operation is suspended. 1: DMAC is operating.	R

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

ESIF Flag (Transfer Escape End Interrupt Flag)

This flag indicates that the transfer escape end interrupt has been generated.

[Setting conditions]

- When 1-repeat size data transfer is completed in repeat transfer mode with the DMINT.RPTIE bit set to 1.
- When 1-block data transfer is completed in block transfer mode with the DMINT.RPTIE bit set to 1.
- When an extended repeat area overflow on the source address occurs while the DMINT.SARIE bit is set to 1 and the DMAMD.SARA[4:0] bits are set to a value other than 00000b (extended repeat area is specified on the transfer source address)
- When an extended repeat area overflow on the destination address occurs while the DMINT.DARIE bit is set to 1 and the DMAMD.DARA[4:0] bits are set to a value other than 00000b (extended repeat area is specified on the transfer destination address)

[Clearing conditions]

- When 0 is written to this bit.
- When 1 is written to the DMCNT.DTE bit.

DTIF Flag (Transfer End Interrupt Flag)

This flag indicates that the transfer end interrupt has been generated.

[Setting conditions]

- When the specified number of unit-transfers are completed in normal transfer mode (the value of the DMCRAL register becoming 0 on completion of transfer)
- When the specified number of repeat transfer operations are completed in repeat transfer mode (the value of the DMCRB register becoming 0 on completion of transfer)
- When the specified number of blocks have been transferred in block transfer mode (the value of the DMCRB register becoming 0 on completion of transfer)

[Clearing conditions]

- When 0 is written to this bit
- When 1 is written to the DMCNT.DTE bit

ACT Flag (DMA Active Flag)

This flag indicates whether the DMAC is in the idle or active state.

[Setting condition]

- When the DMAC starts data transfer operation

[Clearing condition]

- When data transfer in response to one transfer request is completed

18.2.12 DMA Request Source Flag Control Register (DMCSL)

Address(es): DMAC0.DMCSL 0008 201Fh, DMAC1.DMCSL 0008 205Fh, DMAC2.DMCSL 0008 209Fh, DMAC3.DMCSL 0008 20DFh, DMAC4.DMCSL 0008 211Fh, DMAC5.DMCSL 0008 215Fh, DMAC6.DMCSL 0008 219Fh, DMAC7.DMCSL 0008 21DFh

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

Bit	Symbol	Bit Name	Description	R/W
b0	DISEL	Interrupt Select	0: At the beginning of transfer, clear the interrupt status flag of the request source to 0. 1: At the end of transfer, the interrupt status flag of the request source issues an interrupt to the CPU.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DISEL Bit (Interrupt Select)

This bit selects whether the interrupt status flag of the request source is set to 0 or issues an interrupt request to the CPU, at the beginning of DMA transfer.

When DMTMD.DCTG[1:0] = 00b (trigger by software), the setting of the DISEL bit does not affect the operation.

18.2.13 DMAC Module Start Register (DMAST)

Address(es): 0008 2200h

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

Bit	Symbol	Bit Name	Description	R/W
b0	DMST	DMAC Module Start	0: DMAC module stop 1: DMAC module start	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DMST Bit (DMAC Module Start)

When this bit is set to 1, DMAC is ready to accept transfer requests for all channels.

When 1 is written to the DMACm.DMCNT.DTE bit (DMA transfer is enabled) of multiple channels and then this bit is set to 1 (DMAC module start), the corresponding multiple channels can be placed in the transfer request acceptable state at the same time.

When the DMST bit is set to 0 during DMA transfer, DMA transfer is suspended after completion of the current data transfer corresponding to a single transfer request. DMA transfer can be resumed by setting the DMST bit to 1 again.

[Setting condition]

- When 1 is written to this bit

[Clearing condition]

- When 0 is written to this bit

18.2.14 DMAC74 Interrupt Status Monitor Register (DMIST)

Address(es): 0008 2204h

	b7	b6	b5	b4	b3	b2	b1	b0
	DMIS7	DMIS6	DMIS5	DMIS4	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b4	DMIS4	DMAC4 Interrupt Status Flag	0: DMAC4 interrupt is not requested. 1: DMAC4 interrupt is requested.	R
b5	DMIS5	DMAC5 Interrupt Status Flag	0: DMAC5 interrupt is not requested. 1: DMAC5 interrupt is requested.	R
b6	DMIS6	DMAC6 Interrupt Status Flag	0: DMAC6 interrupt is not requested. 1: DMAC6 interrupt is requested.	R
b7	DMIS7	DMAC7 Interrupt Status Flag	0: DMAC7 interrupt is not requested. 1: DMAC7 interrupt is requested.	R

DMISm Flag (DMACm Interrupt Status Flag) (m = 4 to 7)

This flag monitors the DMACm interrupt request. Writing to this flag will be ignored.

While the DMACm.DMINT.DTIE bit is 1 and the DMACm.DMSTS.DTIF flag is 1, or the DMACm.DMINT.ESIE bit is 1 and the DMACm.DMSTS.ESIF flag is 1, the DMIST.DMISm flag is set to 1.

18.3 Operation

18.3.1 Transfer Mode

(1) Normal Transfer Mode

In normal transfer mode, one data is transferred by one transfer request. A maximum of 65535 can be set as the number of transfer operations using the DMACm.DMCRAL register. 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 the DMACm.DMCRB register is invalid in normal transfer mode. Except in free running mode, a transfer end interrupt request can be generated after completion of the specified number of transfer operations.
Table 18.3 summarizes the register update operation in normal transfer mode, and Figure 18.2 shows the operation in normal transfer mode.

Table 18.3 Register Update Operation in Normal Transfer Mode

Register	Function	Update Operation after Completion of a Transfer by One Transfer Request
DMACm.DMSAR	Transfer source address	Increment/decrement/fixed/offset addition*1
DMACm.DMDAR	Transfer destination address	Increment/decrement/fixed/offset addition*1
DMACm.DMCRAL	Transfer counter	Decremented by one/not updated (in free running mode)
DMACm.DMCRAH	—	Not updated (Not used in normal transfer mode)
DMACm.DMCRB	—	Not updated (Not used in normal transfer mode)

Note 1. Offset addition can be specified only for DMAC0.

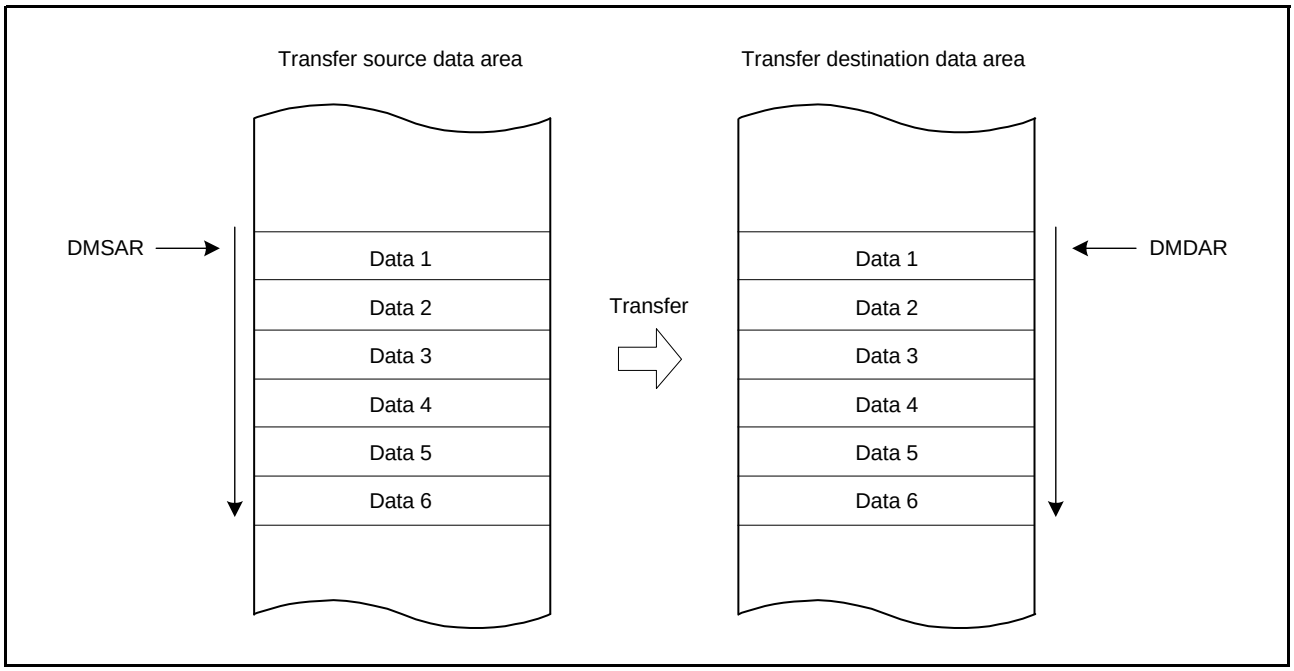


Figure 18.2 Operation in Normal Transfer Mode

(2) Repeat Transfer Mode

In repeat transfer mode, one data is transferred by one transfer request.
A maximum of 1K data can be set as a total repeat transfer size using the DMACm.DMCRA register.
A maximum of 64K can be set as the number of repeat transfer operations using the DMACm.DMCRB register; therefore, a maximum of 64M data (1K data × 64K counts of repeat transfer operations) can be set as a total data transfer size.
Either the transfer source or transfer destination can be specified as a repeat area. When transfer of the repeat size data is completed, the address of the specified repeat area (DMSAR or DMDAR of the DMACm) returns to the transfer start address. When data of the specified repeat size has all been transferred in repeat transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DMACm.DMCNT.DTE bit in the repeat size end interrupt handling.
A transfer end interrupt request can be generated after completion of the specified number of repeat transfer operations.
Table 18.4 summarizes the register update operation in repeat transfer mode, and Figure 18.3 shows the operation in repeat transfer mode.

Table 18.4 Register Update Operation in Repeat Transfer Mode

Register	Function	Update Operation after Completion of a Transfer by One Transfer Request	
		When the DMACm.DMCRAL register is not 1	**When the DMACm.DMCRAL register is 1 (Transfer of the Last Data in Repeat Size)**
DMACm.DMSAR	Transfer source address	Increment/decrement/fixed/offset addition*1	• DMACm.DMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition*1 • DMACm.DMTMD.DTS[1:0] = 01b Initial value of DMACm.DMSAR • DMACm.DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
DMACm.DMDAR	Transfer destination address	Increment/decrement/fixed/offset addition*1	• DMACm.DMTMD.DTS[1:0] = 00b Initial value of DMACm.DMDAR • DMACm.DMTMD.DTS[1:0] = 01b Increment/decrement/fixed/offset addition*1 • DMACm.DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
DMACm.DMCRAH	Repeat size	Not updated	Not updated
DMACm.DMCRAL	Transfer counter	Decremented by one	DMACm.DMCRAH
DMACm.DMCRB	Repeat-count counter	Not updated	Decremented by one

Note 1. Offset addition can be specified only for DMAC0.

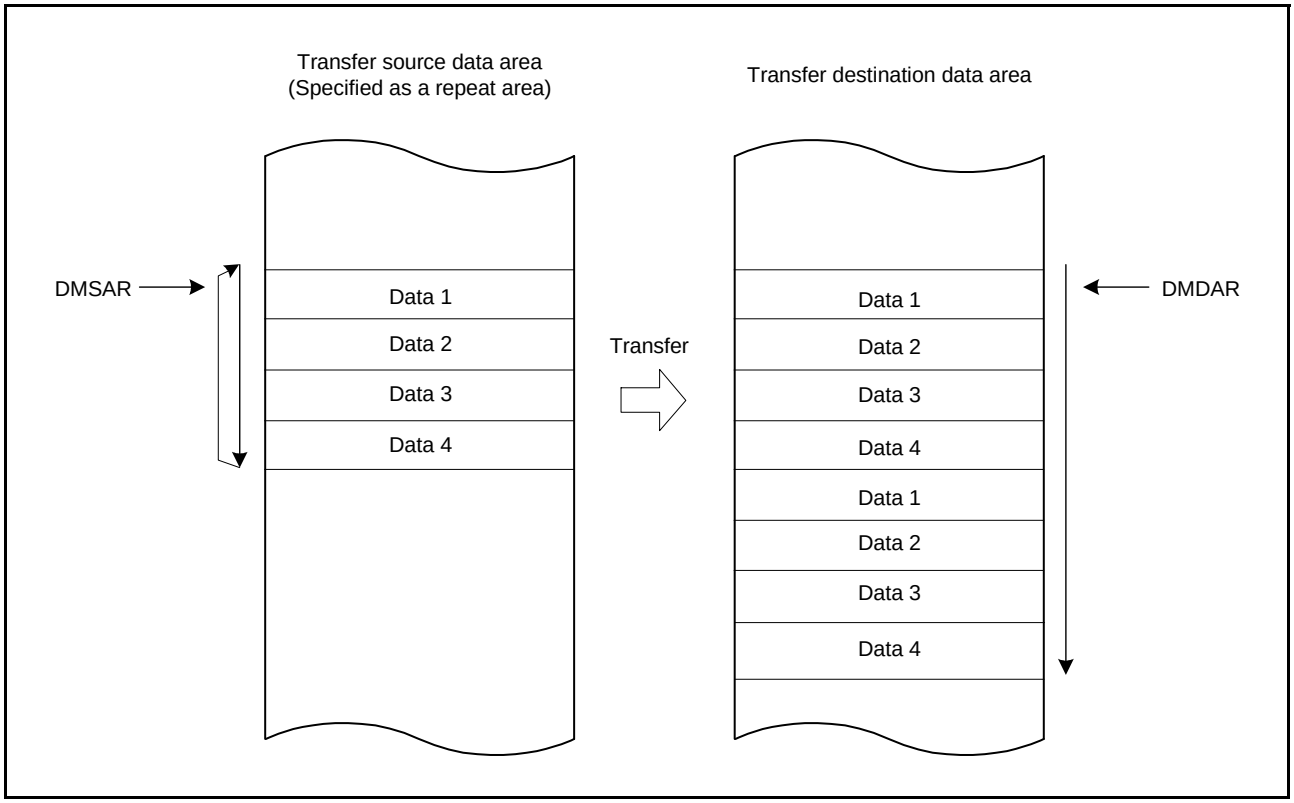


Figure 18.3 Operation in Repeat Transfer Mode

(3) Block Transfer Mode

In block transfer mode, a single block data is transferred by one transfer request.
A maximum of 1K data can be set as a total block transfer size using the DMACm.DMCRA register.
A maximum of 64K can be set as the number of block transfer operations using the DMACm.DMCRB register; therefore, a maximum of 64M data (1K data × 64K counts of block transfer operations) can be set as a total data transfer size.
Either the transfer source or transfer destination can be specified as a block area. When transfer of a single block data is completed, the address of the specified block area (DMSAR or DMDAR of the DMACm) returns to the transfer start address. When a single block data has all been transferred in block transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DMACm.DMCNT.DTE bit in the repeat size end interrupt handling.
Transfer end interrupt request can be generated after completion of the specified number of block transfer operations.
Table 18.5 summarizes the register update operation in block transfer mode, and Figure 18.4 shows the operation in block transfer mode.

Table 18.5 Register Update Operation in Block Transfer Mode

Register	Function	Update Operation after Completion of Single-Block Transfer by One Transfer Request
DMACm.DMSAR	Transfer source address	• DMACm.DMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition*1 • DMACm.DMTMD.DTS[1:0] = 01b Initial value of DMACm.DMSAR • DMACm.DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
DMACm.DMDAR	Transfer destination address	• DMACm.DMTMD.DTS[1:0] = 00 b Initial value of DMACm.DMDAR • DMACm.DMTMD.DTS[1:0] = 01 b Increment/decrement/fixed/offset addition*1 • DMACm.DMTMD.DTS[1:0] = 10 b Increment/decrement/fixed/offset addition*1
DMACm.DMCRAH	Block size	Not updated
DMACm.DMCRAL	Block size counter	DMACm.DMCRAH
DMACm.DMCRB	Transfer-block counter	Decremented by one

Note 1. Offset addition can be specified only for DMAC0.

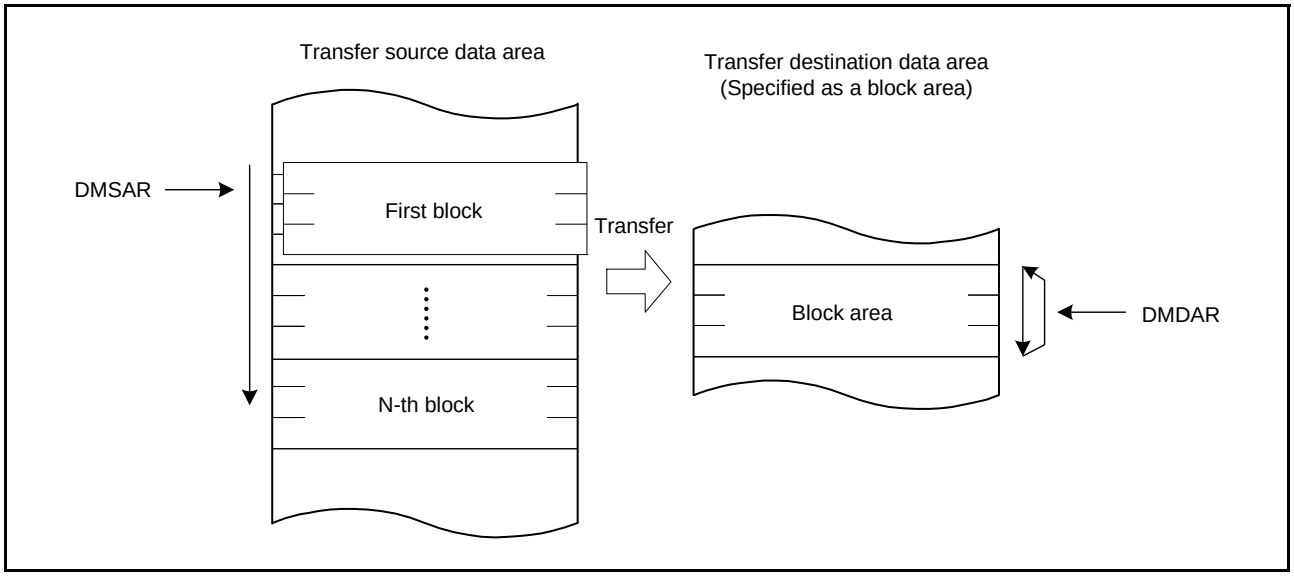


Figure 18.4 Operation in Block Transfer Mode

18.3.2 Extended Repeat Area Function

The DMAC supports a function to specify the extended repeat areas on the transfer source and destination addresses. With the extended repeat areas set, the address registers repeatedly indicate the addresses of the specified extended repeat areas.

The extended repeat areas can be specified separately to the transfer source address register (DMSAR) and transfer destination address register (DMDAR) of DMACm.

The extended repeat area on the source address is specified by the DMACm.DMAMD.SARA[4:0] bits. The extended repeat area on the destination address is specified by the DMACm.DMAMD.DARA[4:0] bits. 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 DMACm.DMINT.SARIE bit is set to 1, the DMACm.DMSTS.ESIF flag is set to 1 and the DMACm.DMCNT.DTE bit is set to 0 to stop DMA transfer. At this time, if the DMACm.DMINT.ESIE bit is set to 1, an interrupt by an extended repeat area overflow is requested. When the DMACm.DMINT.DARIE bit 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 DMACm.DMCNT.DTE bit in the interrupt handling.

Figure 18.5 shows an example of the extended repeat area operation.

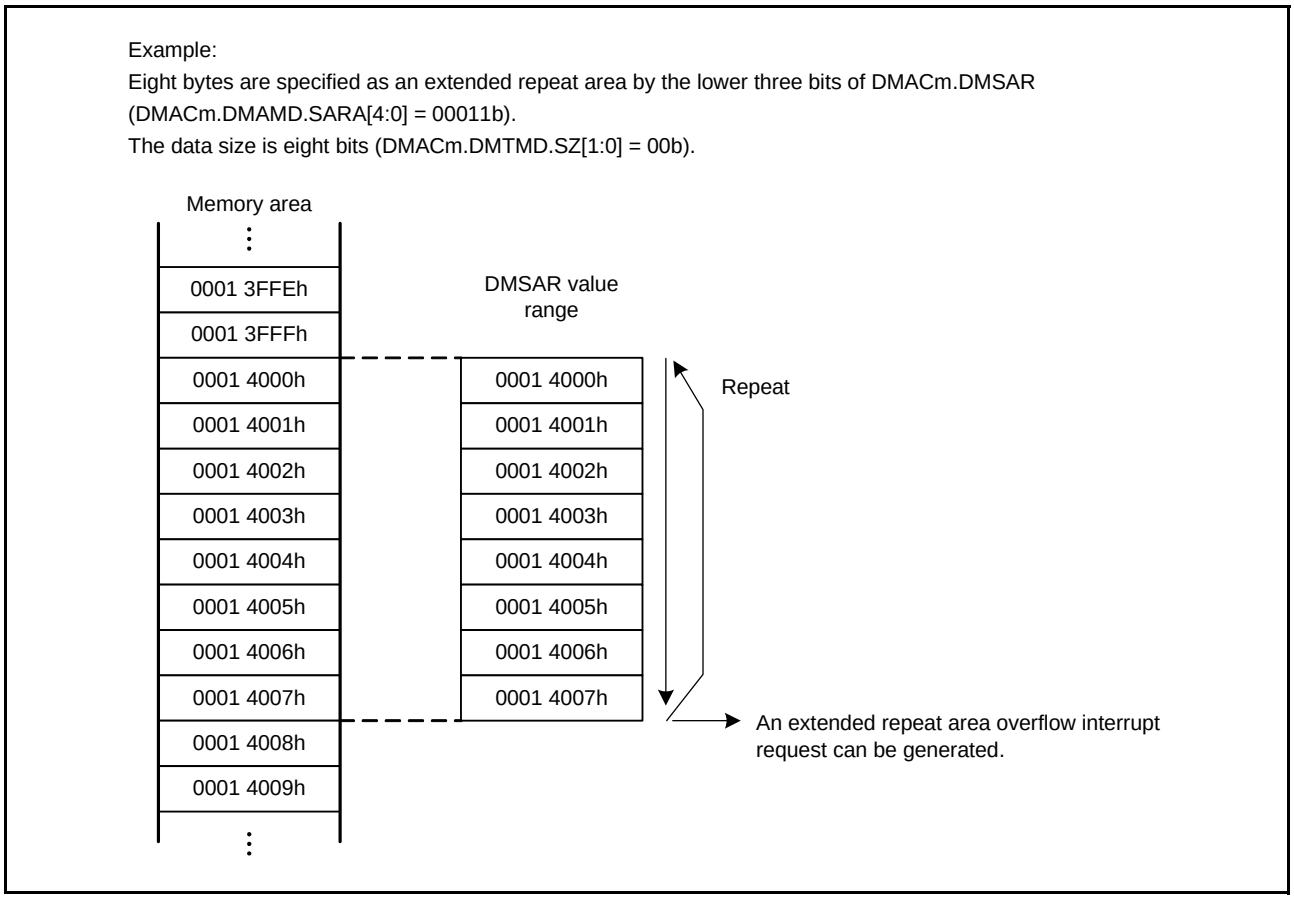


Figure 18.5 Example of Extended Repeat Area Operation

When an interrupt by an extended repeat area overflow is used in block transfer mode, the following should be taken into consideration.
When a transfer is stopped by an interrupt by an extended repeat area overflow, the address register must be set so that the block size is a power of 2 or the block size boundary is aligned with the extended repeat area boundary. When an overflow on the extended repeat area occurs during a transfer of one block, the interrupt by the overflow is suspended until transfer of the block is completed, and the transfer overruns.

Figure 18.6 shows an example when the extended repeat area function is used in block transfer mode.

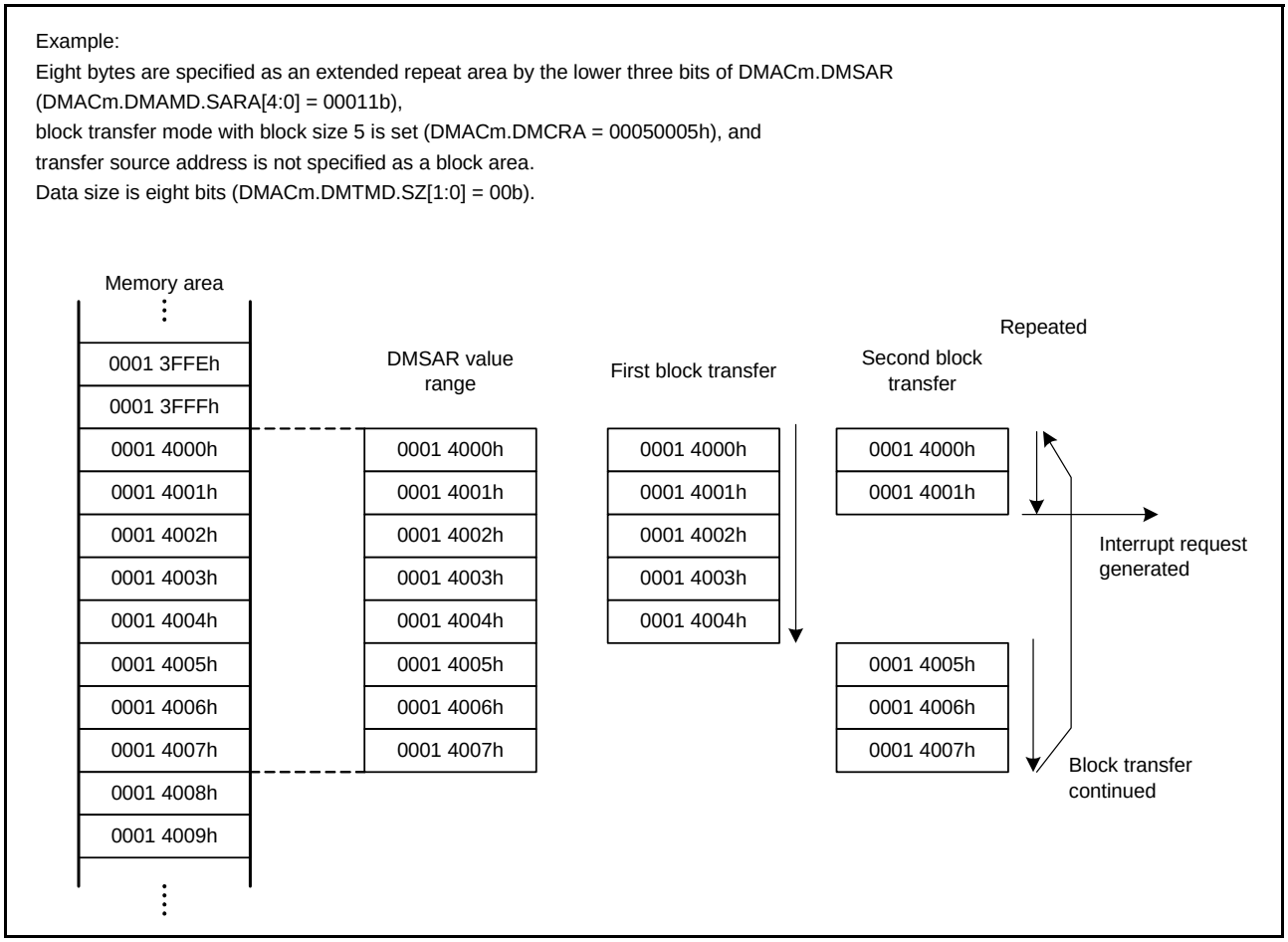


Figure 18.6 Example of Extended Repeat Area Function in Block Transfer Mode

18.3.3 Address Update Function using Offset

The source and destination addresses can be updated by fixing, increment, decrement, or offset addition. When the offset addition is selected, the offset specified by the DMA offset register (DMAC0.DMOFR) 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 the DMAC0.DMOFR register. In this case, the negative value must be 2's complement.

Address update function using offset can be specified only for the DMAC0 channel.

Table 18.6 lists the address update method in each address update mode.

Table 18.6 Address Update Method in Each Address Update Mode

Address Update Mode	Settings of DMACm.DMAMD.SM[1:0] and DMACm.DMAMD.DM[1:0] for Address Update Modes	Address Update Method (for Different DMACm.DMTMD.SZ[1:0] Settings)		
		SZ[1:0] = 00b	SZ[1:0] = 01b	SZ[1:0] = 10b
Address fixed	00b	Fixed		
Offset addition	01b	+DMACm.DMOFR*1		
Increment	10b	+1	+2	+4
Decrement	11b	–1	–2	–4

Note 1. When setting a negative value in the DMA offset register, the value must be 2's complement. The 2's complement is obtained by the following formula.
2's complement of a negative offset value = ~ (offset) + 1 (~: bit inversion)

(1) Basic Transfer Using Offset Addition

Figure 18.7 shows an example of address updating using offset addition.

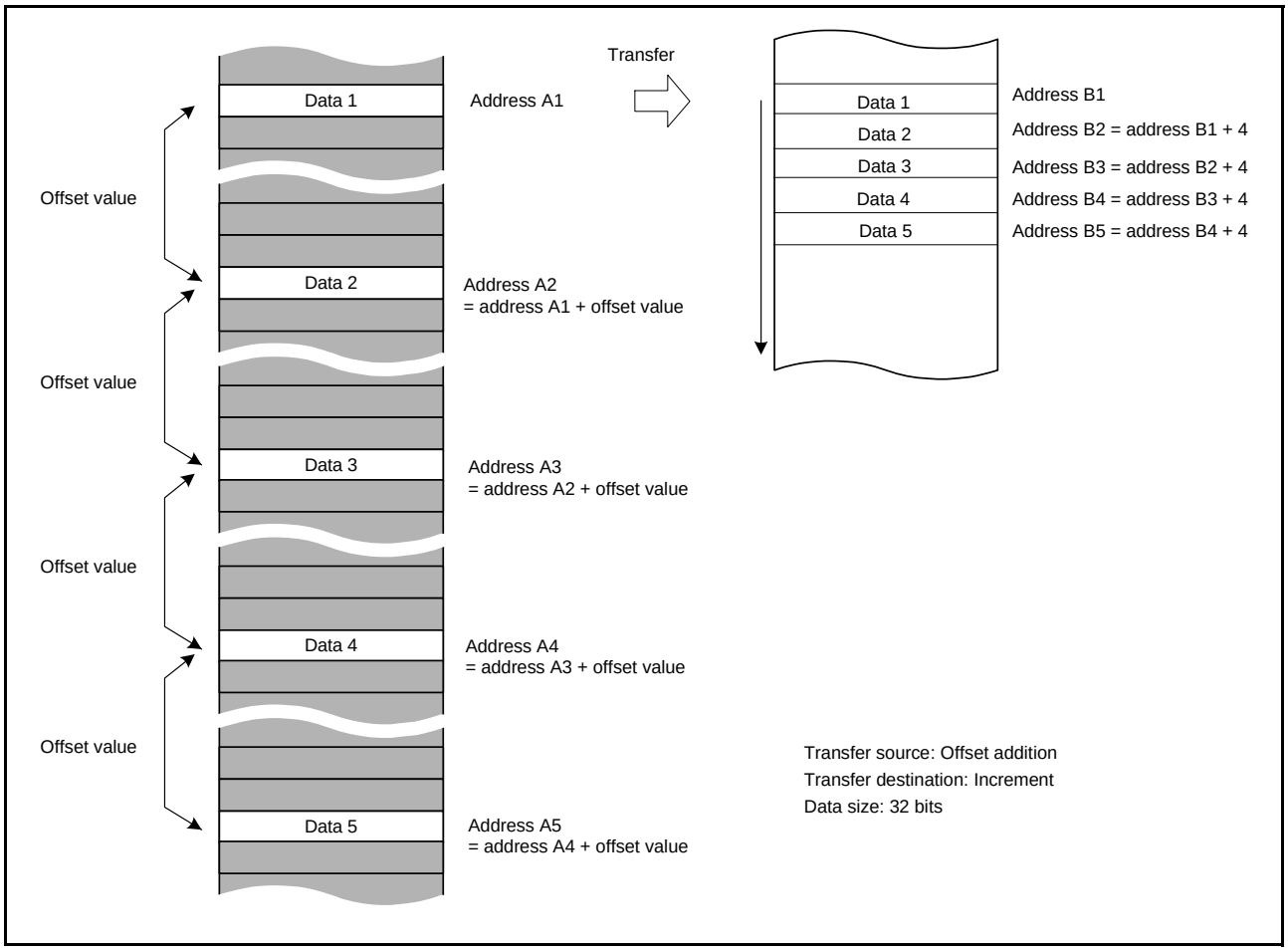


Figure 18.7 Example of Address Updating by Offset Addition

In Figure 18.7, the transfer data is 32 bits long, and offset addition and increment are set as the transfer source address update mode and transfer destination address update mode, respectively. The second and subsequent data is each read from the transfer source address obtained by adding the offset value to the previous address. The data read from the addresses at the specified intervals is written to the continuous locations on the destination.

(2) Example of XY Conversion Using Offset Addition

Figure 18.8 shows the XY conversion using offset addition in repeat transfer mode.
Settings are as follows:

- DMAC0.DMAMD register: Transfer source address update mode: Offset addition
- DMAC0.DMAMD register: Transfer destination address update mode: Destination address is incremented.
- DMAC0.DMTMD register: Transfer data size select: 32 bits
- DMAC0.DMTMD register: Transfer mode select: Repeat transfer
- DMAC0.DMTMD register: Repeat area select: The source is specified as the repeat area.
- DMAC0.DMOFR register: Offset address: 10h
- DMAC0.DMCRA register: Repeat size: 4h
- DMAC0.DMINT register: The repeat size end interrupt is enabled.

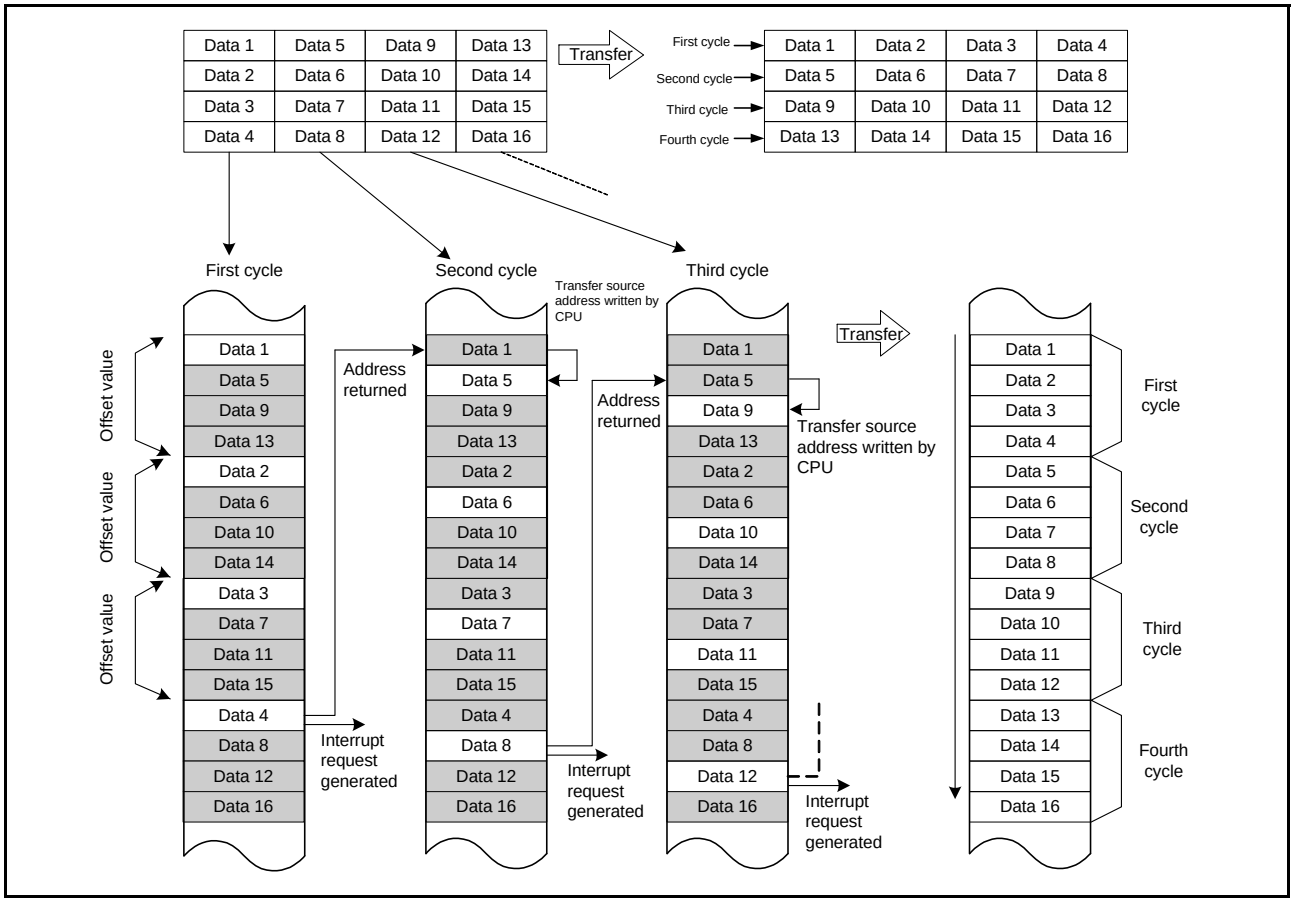


Figure 18.8 XY Conversion Operation Using Offset Addition in Repeat Transfer Mode

When a transfer starts, the offset value is added to the transfer source address every time data is transferred. The transfer data is written to the destination continuous addresses. When data 4 is transferred, which means that the repeat size of transfers is completed, the transfer source address returns to the transfer start address (address of data 1 on the transfer source) and a repeat size end interrupt is requested. While this interrupt stops the transfer temporarily, perform the following.

- DMAC0.DMSAR register: Rewrite the DMA transfer source address to the address of data 5 (with the above example, the data 1 address + 4).
- DMAC0.DMCNT register: Set the DTE bit to 1.

The DMA transfer is resumed from the state when the DMA transfer is stopped. After that, the operations described above are repeated until the transfer source data is transposed to the destination area (XY conversion).

Figure 18.9 shows a flowchart of the XY conversion.

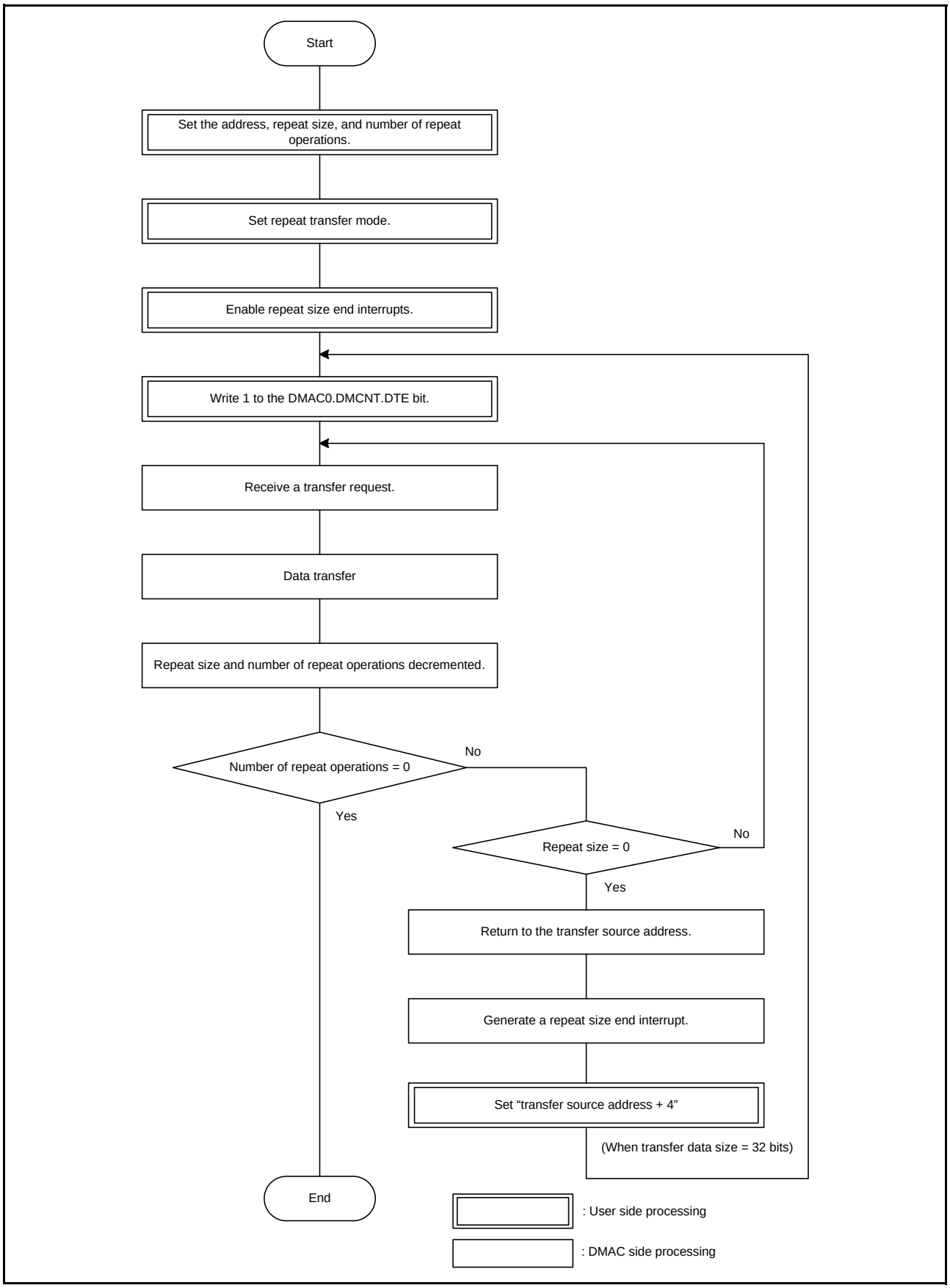


Figure 18.9 XY Conversion Flowchart Using Offset Addition in Repeat Transfer Mode

18.3.4 Request Sources

Software, the interrupt requests from the peripheral modules, and the external interrupt requests can be specified as the DMA request sources. Setting the DMACm.DMTMD.DCTG[1:0] bits selects the request source.

(1) Trigger by Software

Setting the DMACm.DMTMD.DCTG[1:0] bits to 00b enables the trigger by software.
To start DMA transfer by software, set the DMACm.DMTMD.DCTG[1:0] bits to 00b, and then set the DMACm.DMCNT.DTE bit to 1 (DMA transfer is enabled) and the DMACm.DMREQ.SWREQ bit to 1 (DMA transfer is requested) with the DMAST.DMST bit set to 1 (DMAC module start).
When the DMAC is triggered by software while the DMACm.DMREQ.CLRS bit is 0, the DMACm.DMREQ.SWREQ bit is set to 0 after data transfer is started in response to a DMA transfer request.
When the DMAC is triggered by software while the CLRS bit is 1, the SWREQ bit is not set to 0 after data transfer is started. In this case, a DMA transfer request is issued again after completion of a transfer.

(2) Trigger by Interrupt Requests from On-Chip Peripheral Modules or External Interrupt Requests

Interrupt requests from the on-chip peripheral modules and external interrupt requests can be specified as the DMA request sources. The request source can be selected separately for each channel using the ICU.DMRSRm registers (m = 0 to 7).
The DMA transfer is triggered when an interrupt request from the on-chip peripheral module or an external interrupt request is generated while the DMACm.DMTMD.DCTG[1:0] bits are set to 01b (interrupts from the peripheral modules and the external interrupt pins are selected), the DMACm.DMCNT.DTE bit is set to 1 (DMA transfer is enabled), and the DMAST.DMST bit is set to 1 (DMAC module start).

For interrupt requests specified as DMA request sources, refer to Table 15.5, Interrupt Vector Table, in section 15, Interrupt Controller (ICUE).

18.3.5 Operation Timing

Figure 18.10 and Figure 18.11 show DMAC operation timing examples.

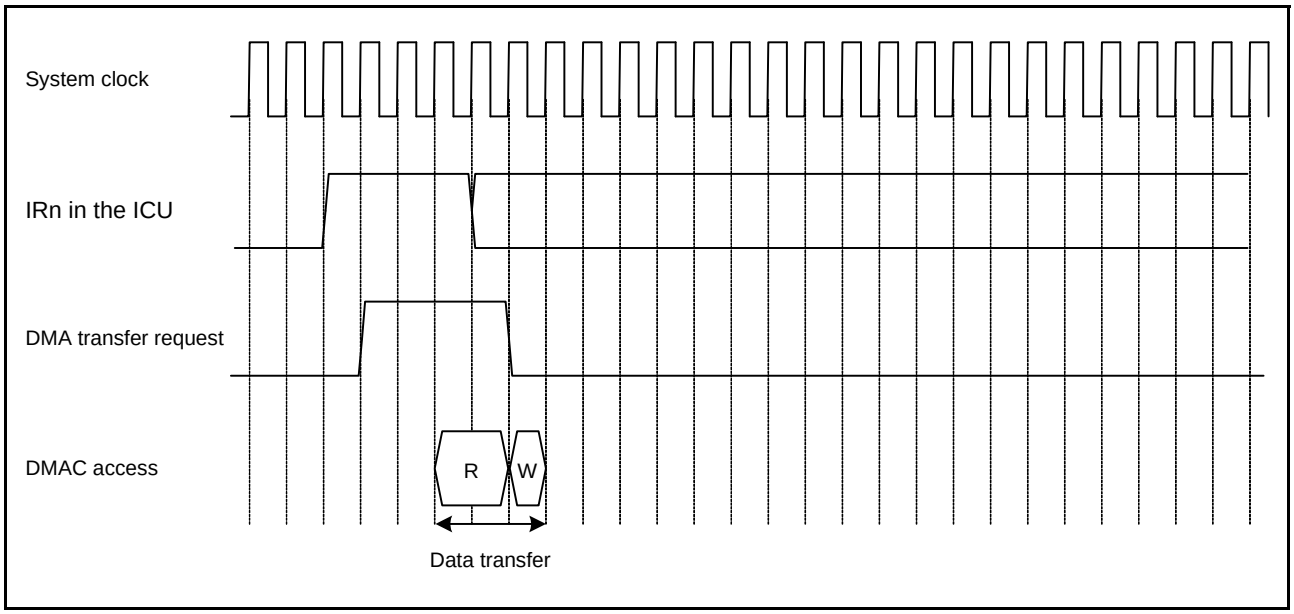


Figure 18.10 DMAC Operation Timing Example (1) (Trigger by Interrupt from Peripheral Module/External Interrupt Input Pin, Normal Transfer Mode, Repeat Transfer Mode)

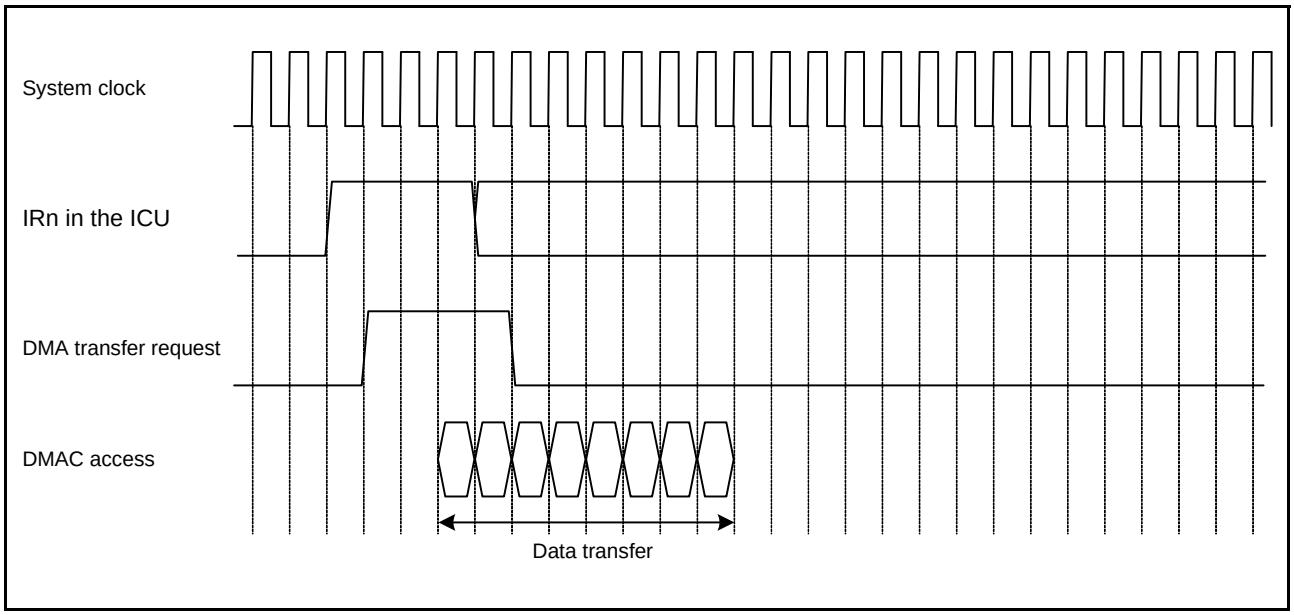


Figure 18.11 DMAC Operation Timing Example (2) (Trigger by Interrupt from Peripheral Module/External Interrupt Input Pin, Block Transfer Mode, Block Size = 4)

18.3.6 DMAC Execution Cycles

Table 18.7 lists execution cycles in one DMAC data transfer operation.

Table 18.7 DMAC Execution Cycles

Transfer Mode	Data Transfer (Read)	Data Transfer (Write)
Normal	Cr+1	Cw
Repeat	Cr+1	Cw
Block*1	P × Cr	P × Cw

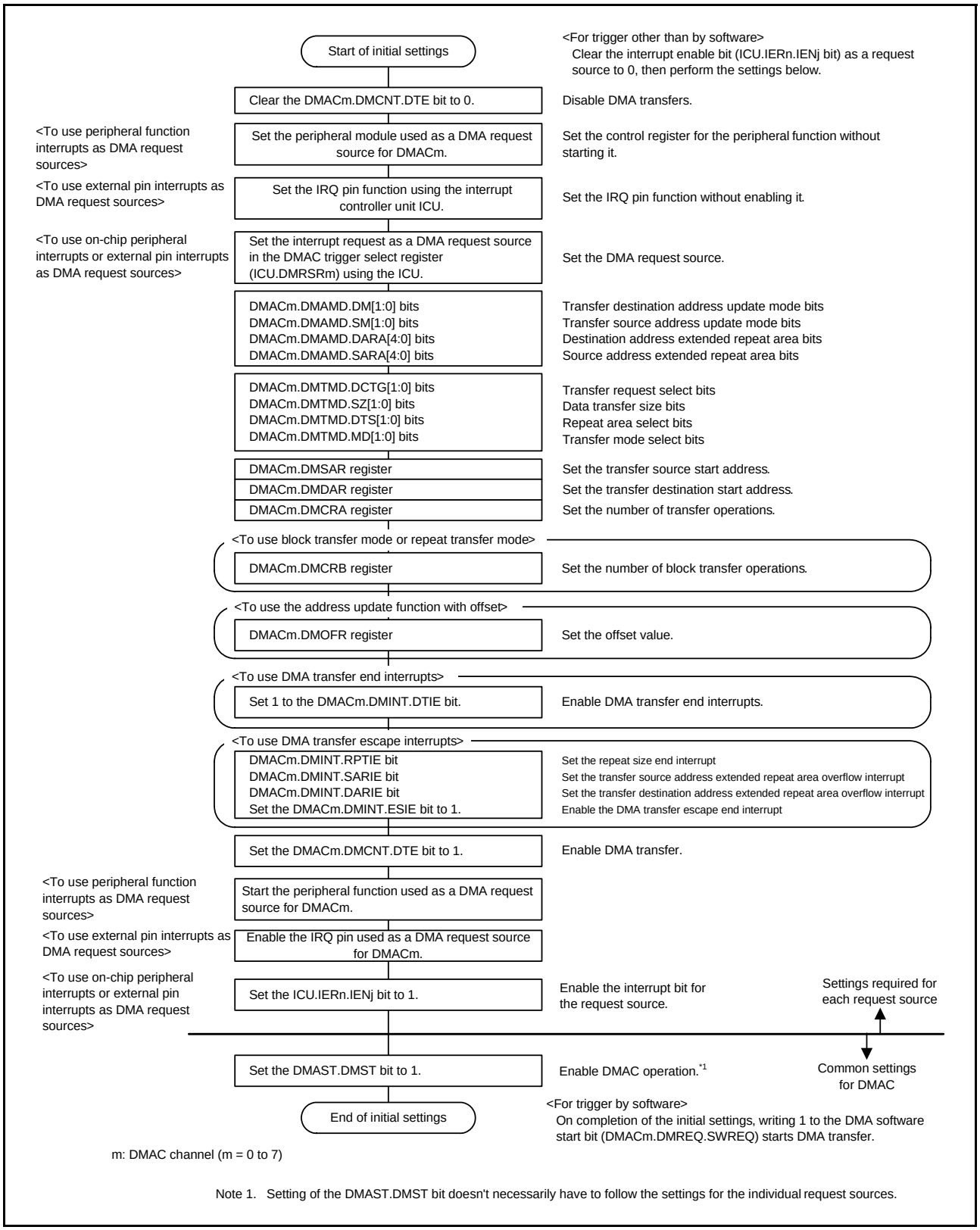
Note 1. This is the case when the block size is 2 or more. When the block size is 1, normal transfer cycle is applied.

P: Block size (DMCRAH register setting)
Cr: Data read destination access cycle
Cw: Data write destination access cycle

Cr and Cw depend on the access destination. For the number of cycles for each access destination, refer to section 53, RAM, section 55, Flash Memory (FLASH), section 5, I/O Registers, and section 16.2.7, External Bus.
The unit for +1 in “Data Transfer (Read)” column is one system clock cycle (ICLK).
For the operation example, refer to section 18.3.5, Operation Timing.

18.3.7 Activating the DMAC

Figure 18.12 shows the register setting procedure.



Note 1. Setting of the DMAST.DMST bit doesn't necessarily have to follow the settings for the individual request sources.

Figure 18.12 Register Setting Procedure

18.3.8 Starting DMA Transfer

Setting the DMACm.DMCNT.DTE bit to 1 (DMA transfer enabled) and setting the DMAST.DMST bit to 1 (DMAC module start) enable DMA transfer of channel m (m = 0 to 7).
Another transfer request cannot be accepted during the transfer of other DMAC channel or DTC. When the proceeding transfer is completed, channel arbitration is performed where a DMA transfer request of the highest priority channel is accepted and DMA transfer of the channel starts. When DMA transfer starts, the DMACm.DMSTS.ACT flag is set to 1 (the DMAC is in the active state).

18.3.9 Registers during DMA Transfer

The DMAC registers are updated by a DMA transfer. The value to be updated differs according to the other settings and the transfer state. The registers to be updated are DMSAR, DMDAR, DMCRA, DMCRB, DMCNT, and DMSTS of DMACm.

(1) DMA Source Address Register (DMACm.DMSAR)

When data has been transferred in response to one transfer request, the contents of this register are updated to the address to be accessed by the next transfer request.
For details on register update operation in each transfer mode, refer to Table 18.3 to Table 18.5.

(2) DMA Destination Address Register (DMACm.DMDAR)

When data has been transferred in response to one transfer request, the contents of this register are updated to the address to be accessed by the next transfer request.
For details on register update operation in each transfer mode, refer to Table 18.3 to Table 18.5.

(3) DMA Transfer Count Register (DMACm.DMCRA)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.
For details on register update operation in each transfer mode, refer to Table 18.3 to Table 18.5.

(4) DMA Block Transfer Count Register (DMACm.DMCRB)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.
For details on register update operation in each transfer mode, refer to Table 18.3 to Table 18.5.

(5) DMA Transfer Enable Bit (DMACm.DMCNT.DTE)

Although the DMACm.DMCNT.DTE bit enables or disables data transfer by the register write access, it is automatically set to 0 by the DMAC according to the DMA transfer state.
The conditions for clearing this bit by the DMAC are as follows:

- When the specified total volume of data transfer is completed
- When DMA transfer is stopped by the repeat size end interrupt
- When DMA transfer is stopped by the extended repeat area overflow interrupt

Writing to the registers for the channels when the corresponding DMACm.DMCNT.DTE bit is set to 1 is prohibited (except for the DMACm.DMCNT register). In this case, writing must be performed after the bit is set to 0.

(6) DMA Active Flag (DMACm.DMSTS.ACT)

The DMACm.DMSTS.ACT flag indicates whether the DMACm is in the idle or active state.
This flag is set to 1 when the DMAC starts data transfer, and is set to 0 when data transfer in response to one transfer request is completed.
Even when DMA transfer is stopped by writing 0 to the DMACm.DMCNT.DTE bit during DMA transfer, this flag remains 1 until DMA transfer is completed.

(7) Transfer End Interrupt Flag (DMACm.DMSTS.DTIF)

The DMACm.DMSTS.DTIF flag is set to 1 after DMA transfer of the total transfer size of data is completed.
When both this flag and the DMACm.DMINT.DTIE bit 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 DMACm.DMSTS.ACT flag is set to 0 indicating the DMA transfer end.
This flag is automatically set to 0 when the DMACm.DMCNT.DTE bit is set to 1 during the interrupt handling.

(8) Transfer Escape End Interrupt Flag (DMACm.DMSTS.ESIF)

The DMACm.DMSTS.ESIF flag is set to 1 when a repeat size end interrupt or extended repeat area overflow interrupt is requested. When this bit and the DMACm.DMINT.ESIE bit 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 DMACm.DMSTS.ACT flag is set to 0 indicating the DMA transfer end.
This flag is automatically set to 0 when the DMACm.DMCNT.DTE bit 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, refer to section 15, Interrupt Controller (ICUE).

18.3.10 Channel Priority

When multiple DMA transfer requests are present, the DMAC determines the priority of channels that have DMA transfer requests.
The channel priority is fixed as channel 0 > channel 1 > channel 2 > channel 3 > channel 4 > channel 5 > channel 6 > channel 7 (channel 0: highest).
When a DMA transfer request is generated during data transfer, channel arbitration is started after the final data has been transferred, and DMA transfer of the higher-priority channel starts.

18.4 Ending DMA Transfer

The operation for ending DMA transfer depends on the transfer end conditions. When DMA transfer ends, the DMACm.DMCNT.DTE bit and the DMACm.DMSTS.ACT flag are changed from 1 to 0, indicating that DMA transfer has ended.

18.4.1 Transfer End by Completion of Specified Total Number of Transfer Operations

(1) In Normal Transfer Mode (DMACm.DMTMD.MD[1:0] = 00b)

When the value of the DMACm.DMCRAL register changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMACm.DMCNT.DTE bit is set to 0 and the DMACm.DMSTS.DTIF flag is set to 1 at the same time. If the DMACm.DMINT.DTIE bit 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 the DMACm.DMCRB register changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMACm.DMCNT.DTE bit is set to 0 and the DMACm.DMSTS.DTIF flag is set to 1 at the same time. If the DMACm.DMINT.DTIE bit 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 the DMACm.DMCRB register changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMACm.DMCNT.DTE bit is set to 0 and the DMACm.DMSTS.DTIF flag is set to 1 at the same time. If the DMACm.DMINT.DTIE bit 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, refer to section 15, Interrupt Controller (ICUE).

18.4.2 Transfer End by Repeat Size End Interrupt

In repeat transfer mode, a repeat size end interrupt is requested when transfer of a 1-repeat size of data is completed while the DMACm.DMINT.RPTIE bit is set to 1. When the interrupt is requested to complete DMA transfer, the DMACm.DMCNT.DTE bit is set to 0 and the DMACm.DMSTS.ESIF flag is set to 1. If the DMACm.DMINT.ESIE bit 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 DMACm.DMCNT.DTE bit.

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, refer to section 15, Interrupt Controller (ICUE).

18.4.3 Transfer End by Interrupt on Extended Repeat Area Overflow

When an overflow on the extended repeat area occurs while the extended repeat area is specified and the DMACm.DMINT.SARIE or DMACm.DMINT.DARIE bit 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 DMACm.DMCNT.DTE bit is set to 0, and the DMACm.DMSTS.ESIF flag is set to 1. If the DMACm.DMINT.ESIE bit 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, refer to section 15, Interrupt Controller (ICUE).

18.5 Interrupts

Each DMAC channel can output an interrupt request to the CPU or the DTC after transfer in response to one request is completed. When the transfer destination is the external bus or the on-chip peripheral bus, an interrupt request is generated upon completion of data write to the write buffer not to the actual transfer destination.

Table 18.8 lists the relation among the interrupt sources, the interrupt status flags, and the interrupt enable bits. Figure 18.13 shows the schematic logic diagram of interrupt outputs (DMAC0 to DMAC3). Figure 18.14 shows the schematic logic diagram of interrupt outputs (DMAC4 to DMAC7). Figure 18.15 shows the DMAC interrupt handling routine to resume/terminate DMA transfer.

Table 18.8 Relation among Interrupt Sources, Interrupt Status Flags, and Interrupt Enable Bits

Interrupt Sources		Interrupt Enable Bits	Interrupt Status Flags	Request Output Enable Bits
Transfer end		—	DMACm.DMSTS.DTIF	DMACm.DMINT.DTIE
Escape transfer end	Repeat size end	DMACm.DMINT.RPTIE	DMACm.DMSTS.ESIF	DMACm.DMINT.ESIE
	Source address extended repeat area overflow	DMACm.DMINT.SARIE		
	Destination address extended repeat area overflow	DMACm.DMINT.DARIE		

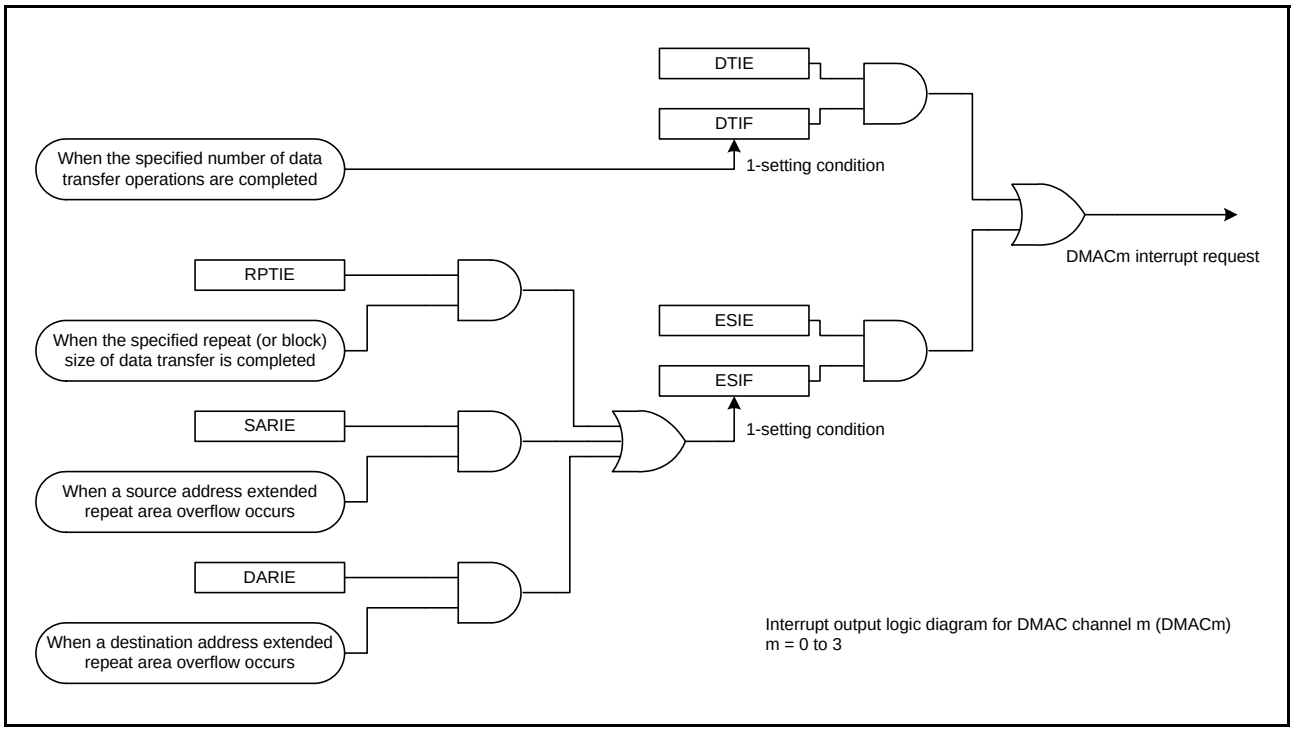


Figure 18.13 Schematic Logic Diagram of Interrupt Outputs (DMAC0 to DMAC3)

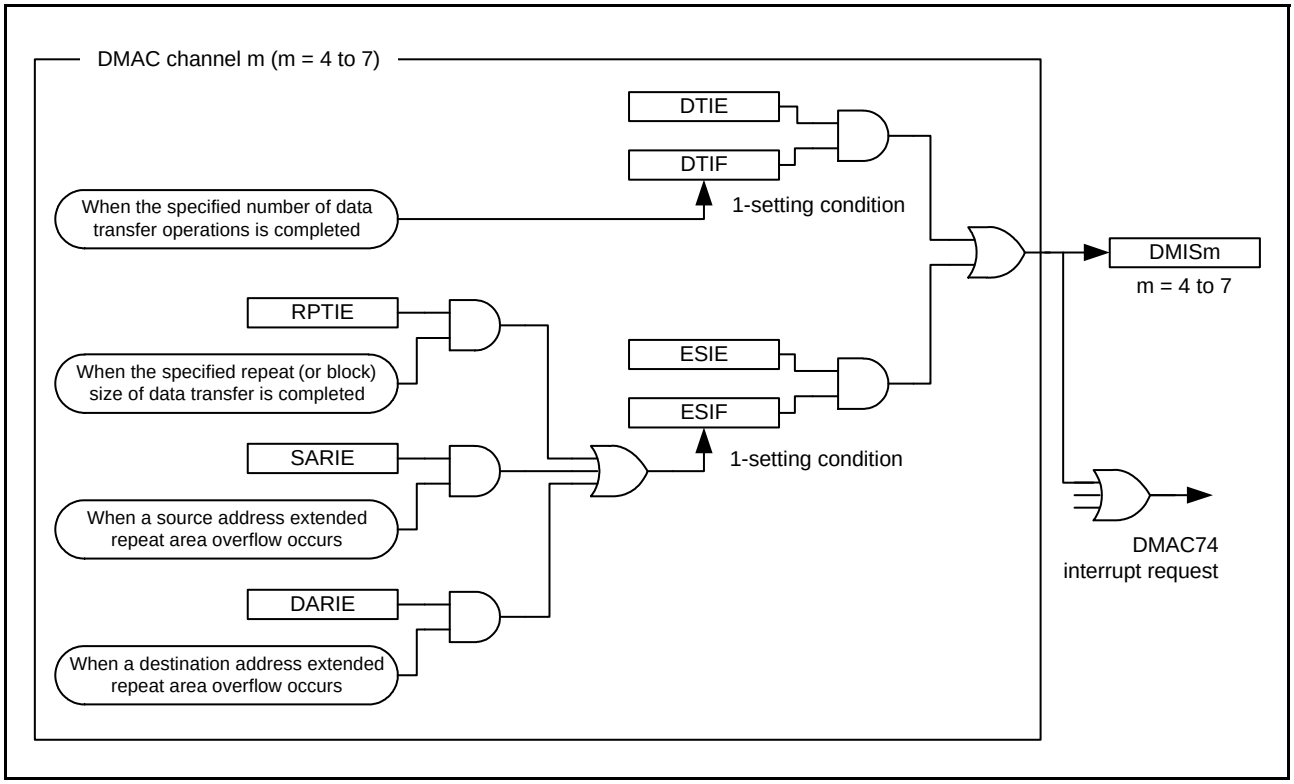


Figure 18.14 Schematic Logic Diagram of Interrupt Outputs (DMAC4 to DMAC7)

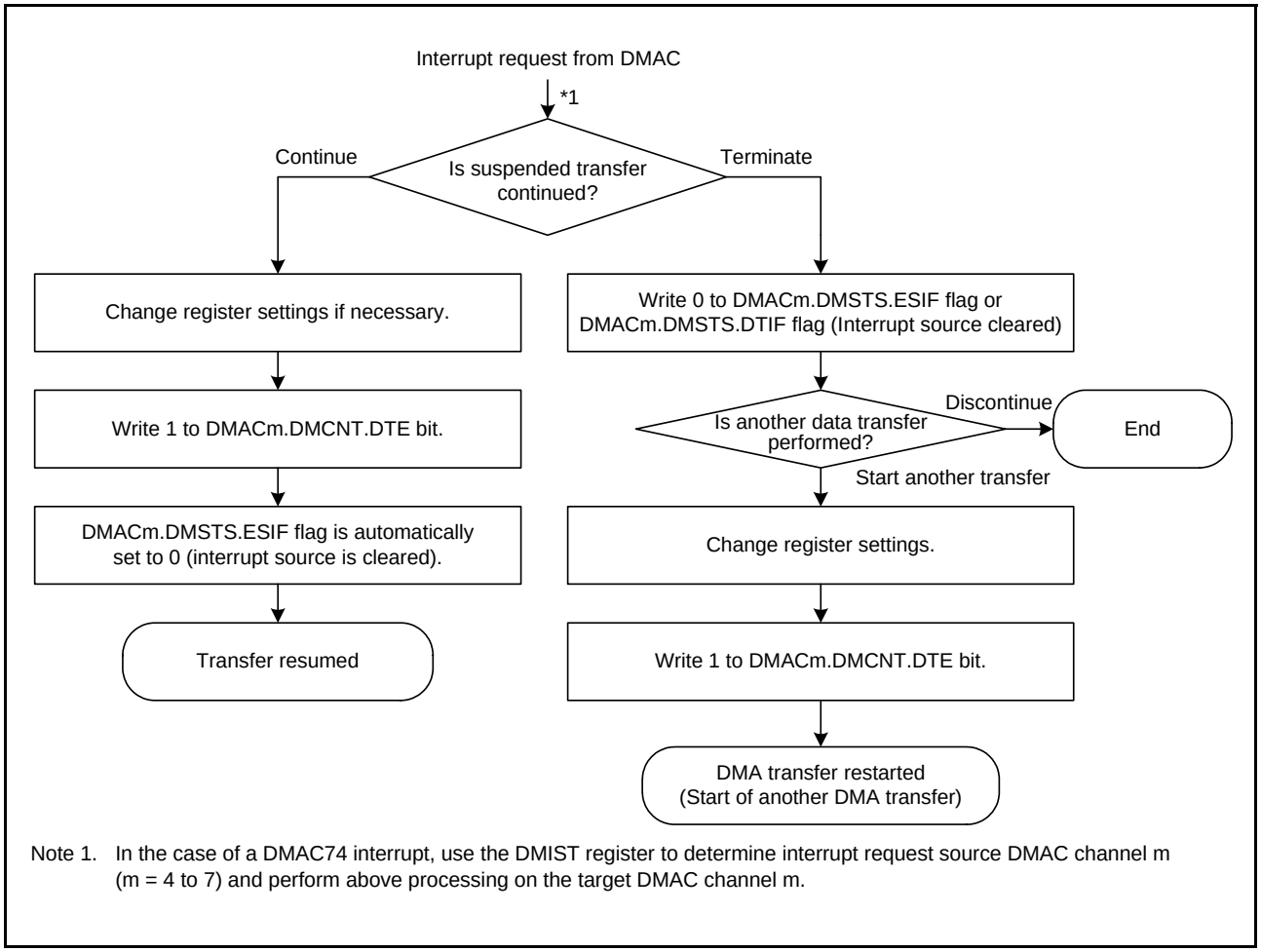
Specifically, the different procedures are used for canceling an interrupt to restart DMA transfer in the following two cases: (1) discontinuing or terminating DMA transfer and (2) continuing DMA transfer.

(1) When Discontinuing or Terminating DMA Transfer

Write 0 to the DMACm.DMSTS.DTIF flag to clear a transfer end interrupt, and to the DMACm.DMSTS.ESIF flag 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 DMACm.DMCNT.DTE bit to 1 (DMA transfer enabled).

(2) When Continuing DMA Transfer

Write 1 to the DMACm.DMCNT.DTE bit. The DMACm.DMSTS.ESIF flag is automatically set to 0 (interrupt source cleared), and DMA transfer is resumed.



Note 1. In the case of a DMAC74 interrupt, use the DMIST register to determine interrupt request source DMAC channel m (m = 4 to 7) and perform above processing on the target DMAC channel m.

Figure 18.15 DMAC Interrupt Handling Routine to Resume/Terminate DMA Transfer

18.6 Event Link

Each DMAC channel outputs an event link request signal each time the channel completes data transfer (or block transfer in block transfer mode). However, when the transfer destination is the external bus or internal peripheral bus, an event link request signal is generated when the writing to the write buffer is accepted.

18.7 Low-Power Consumption Function

Before transition to the module-stop state, all-module clock stop mode, software standby mode, or deep software standby mode, set the DMAST.DMST bit to 0 (the DMAC suspended), and then perform the following.

(1) Module-Stop Function

Writing 1 to the MSTPCRA.MSTPA28 bit (transition to the module-stop state) 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, refer to 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 DMAST.DMST bit to 1 again.
To use a request that is generated in all-module clock-stop mode and software standby mode as an interrupt request to the CPU but not as a DMAC startup request, specify the CPU as the interrupt request destination in accordance with the description in section 15, Interrupt Controller (ICUE), and then execute the WAIT instruction.

18.8 Usage Notes

18.8.1 DMA Transfer to External Devices

In DMA transfer to an external device, the DMACm.DMSTS.ACT flag may be set to 0 (DMAC transfer suspended) during the period from the beginning of the final data write to the end of the external bus access.

18.8.2 DMA Transfer to Peripheral Modules

In DMA transfer to a peripheral module, the DMACm.DMSTS.ACT flag may be set to 0 (DMAC transfer suspended) during the period from the beginning of the final data write to the end of the peripheral bus access.

18.8.3 Access to the Registers during DMA Transfer

Do not write to the DMSAR, DMDAR, DMCRA, DMCRB, DMTMD, DMINT, DMAMD, DMOFR, and DMCSL registers of DMACm while the DMSTS.ACT flag of the same channel is set to 1 (DMAC active state) or the DMCNT.DTE bit of the same channel is set to 1 (DMA transfer enabled).

18.8.4 DMA Transfer to Reserved Areas

DMA transfer to the reserved areas is prohibited. If such an access is made, transfer results are not guaranteed. For details on the reserved areas, refer to section 4, Address Space.

18.8.5 Interrupt Request by the DMA Request Source Flag Control Register (DMCSL) at the End of Each Transfer

While the DMACm.DMCSL.DISEL bit is 1, an interrupt request is issued to the CPU at the end of each transfer that has been triggered by one DMA request. Unlike the transfer end interrupt that the DMAC outputs or the escape end interrupt, the interrupt of this type is issued to the CPU at the end of DMA transfer without clearing the interrupt status flag of the DMA request source to 0 by changing the interrupt request destination to the CPU. In this case, since the interrupt status flag is not set to 0 at the end of DMAC transfer, it should be set to 0 by the CPU interrupt routine.
The interrupt flag is cleared when the CPU interrupt is accepted.
For the change of the settings on the interrupt flag or the interrupt request destination, refer to section 15, Interrupt Controller (ICUE). For the DMACm.DMCSL.DISEL bit setting, refer to section 18.2.12, DMA Request Source Flag Control Register (DMCSL).

18.8.6 Setting of DMAC Trigger Select Register of the Interrupt Controller (ICU.DMRSRm)

The DMAC trigger select register (ICU.DMRSRm) should be set while the DMA transfer enable bit (DMACm.DMCNT.DTE) is 0 (DMA transfer is disabled). Moreover, the DTC transfer request enable register (ICU.DTCERn) 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 registers, refer to section 15, Interrupt Controller (ICUE).

18.8.7 Suspending or Restarting DMA Transfer

To suspend a DMA transfer request, write 0 to the interrupt enable bit for the request source (ICU.IERn.IENj bit). To restart the DMA transfer, write 1 to the ICU.IERn.IENj bit with the setting shown in section 18.3.7, Activating the DMAC.

19. EXDMA Controller (EXDMACa)

This MCU incorporates a 2-channel direct memory access controller (EXDMAC) designed exclusively for external bus transfer. The EXDMAC is a module to transfer data without the CPU. When a DMA transfer request is generated, the EXDMAC transfers data stored at the transfer source address to the transfer destination address.

19.1 Overview

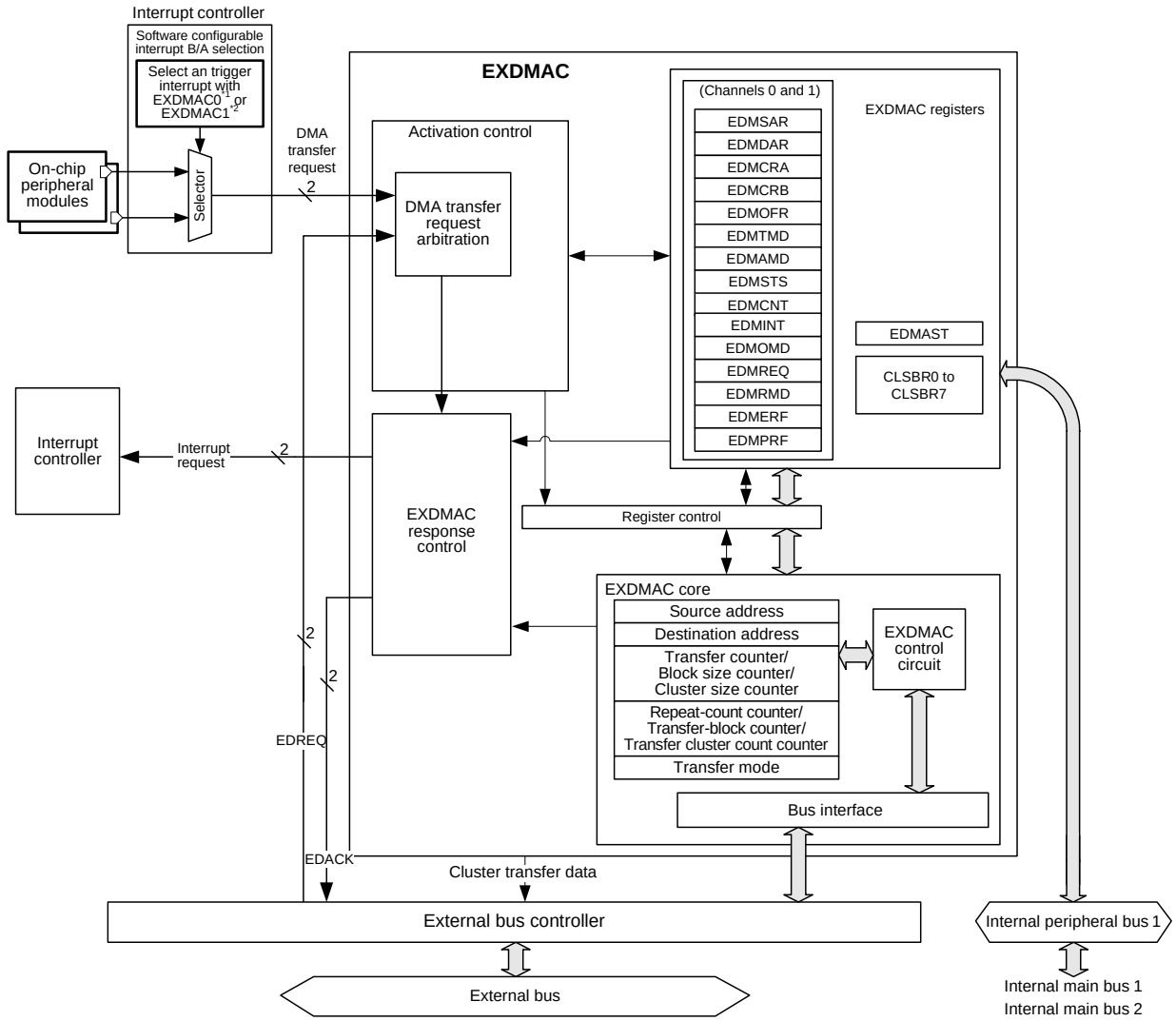
Table 19.1 lists the specifications of the EXDMAC, and Figure 19.1 shows a block diagram of the EXDMAC.

Table 19.1 Specifications of EXDMAC (1/2)

Item		Description
Number of channels		2 (EXDMAC0 and EXDMAC1)
Transfer space		512 Mbytes (External areas at addresses 0000 0000h to 0FFF FFFFh and F000 0000h to FFFF FFFFh excluding reserved areas)
Maximum transfer volume		1 M data (Maximum number of transfer operations in block transfer mode: 1,024 data × 1,024 blocks)
DMA request source		• Request source selectable from the following three sources for each channel Software trigger External DMA transfer request input DMA transfer request from peripheral modules (TPU1.TRGA or MTU1.TRGA) (Channel 0: a software configurable interrupt B request from TPU1.TRGA selected in ICU.SLIBR144 or a software configurable interrupt A request from MTU1.TRGA selected in ICU.SLIAR208; Channel 1: a software configurable interrupt B request from TPU1.TRGA selected in ICU.SLIBR145 or a software configurable interrupt A request from MTU1.TRGA selected in ICU.SLIAR209)
Channel priority		Channel 0 > Channel 1 (Channel 0: Highest)
Transfer data	Single data	Bit length: 8, 16, 32 bits
	Block size	Number of data: 1 to 1,024 data
	Cluster size	Number of data: 1 to 8 data
Transfer mode	Normal transfer mode	• One data transfer by one DMA transfer request • Free running mode (setting in which total number of data transfer operations is not specified) settable
	Repeat transfer mode	• One data transfer by one DMA transfer request • Program returns to the transfer start address on completion of the repeat size of data transfer specified for the transfer source or destination. • Maximum settable repeat size: 1,024 data
	Block transfer mode	• One block data transfer by one DMA transfer request • Maximum settable block size: 1,024 data
	Cluster transfer mode	• One cluster data transfer by one DMA transfer request • Maximum settable cluster size: 8 data (32 bytes)
Address mode	Single address mode	• Transfers data by accessing the transfer source or destination peripheral device with the EDACKn signal (n = 0, 1) and specifying the address of the other peripheral device. • Available in normal transfer mode, repeat transfer mode, and block transfer mode.
	Dual address mode	• Transfers data by specifying the addresses of transfer source and destination. • Available in normal transfer mode, repeat transfer mode, block transfer mode, and cluster transfer mode.
Selective functions	Extended repeat area function	• Function in which data can be transferred by repeating the address values in the specified range with the upper bit values in the transfer address register fixed • Area of 2 bytes to 128 Mbytes separately settable as extended repeat area for transfer source and destination

Table 19.1 Specifications of EXDMAC (2/2)

Item		Description
Interrupt request	Transfer end interrupt	Generated when the specified number of transfers is completed in normal transfer mode Generated when the specified repeat count of transfers is completed in repeat transfer mode Generated when the specified block count of transfers is completed in block transfer mode Generated when the specified cluster count of transfers is completed in cluster transfer mode
	Transfer escape end interrupt	Generated when the repeat size of data transfer is completed or the extended repeat area overflows.
Low-power consumption function		The module-stop state can be set.



Note 1. A software configurable interrupt B request from TPU1.TRGA selected in ICU.SLIBR144 or a software configurable interrupt A request from MTU1.TRGA selected in ICU.SLIAR208

Note 2. A software configurable interrupt B request from TPU1.TRGA selected in ICU.SLIBR145 or a software configurable interrupt A request from MTU1.TRGA selected in ICU.SLIAR209

Figure 19.1 Block Diagram of EXDMAC

Table 19.2 lists the input/output pins of the EXDMAC.

Table 19.2 Pin Configuration of EXDMAC

Channel	Pin Name	I/O	Description
EXDMAC0	EDREQ0	Input	EXDMAC0 external DMA transfer request
	EDACK0	Output	EXDMAC0 single address transfer acknowledge
EXDMAC1	EDREQ1	Input	EXDMAC1 external DMA transfer request
	EDACK1	Output	EXDMAC1 single address transfer acknowledge

19.2 Register Descriptions

19.2.1 EXDMA Source Address Register (EDMSAR)

Address(es): EXDMAC0.EDMSAR 0008 2800h, EXDMAC1.EDMSAR 0008 2840h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer source start address.	0000 0000h to 0FFF FFFFh (256 Mbytes) F000 0000h to FFFF FFFFh (256 Mbytes)	R/W

This register is used to set the start address of the transfer source.
Set the EDMSAR register while EXDMAC stops (EDMAST.DMST bit = 0) or DMA transfer is disabled (EDMCNT.DTE bit = 0).
Setting bits 31 to 29 is invalid; a value of bit 28 is extended to bits 31 to 29. Reading the EDMSAR register returns the extended value.

19.2.2 EXDMA Destination Address Register (EDMDAR)

Address(es): EXDMAC0.EDMDAR 0008 2804h, EXDMAC1.EDMDAR 0008 2844h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer destination start address.	0000 0000h to 0FFF FFFFh (256 Mbytes) F000 0000h to FFFF FFFFh (256 Mbytes)	R/W

This register is used to set the start address of the transfer destination.
Set the EDMDAR register while EXDMAC stops (EDMAST.DMST bit = 0) or DMA transfer is disabled (EDMCNT.DTE bit = 0).
Setting bits 31 to 29 is invalid; a value of bit 28 is extended to bits 31 to 29. Reading the EDMDAR register returns the extended value.

19.2.3 EXDMA Transfer Count Register (EDMCRA)

Address(es): EXDMAC0.EDMCRA 0008 2808h, EXDMAC1.EDMCRA 0008 2848h

· Normal transfer mode

	EDMCRAH															
	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

· Repeat transfer mode, block transfer mode

	EDMCRAH															
	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Note: The function differs depending on the transfer mode.

· Cluster transfer mode

	EDMCRAH															
	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Symbol	Bit Name	Description	R/W
EDMCRAL	Lower bits of transfer count	Specifies the number of transfer operations.	R/W
EDMCRAH	Upper bits of transfer count		R/W

Note: Set the same value for EDMCRAH and EDMCRAL in repeat transfer mode, block transfer mode, and cluster transfer mode.

This register is used to specify the transfer count of the DMA. The function of this register depends on transfer mode.

(1) Normal transfer mode (EXDMACn.EDMTMD.MD[1:0] bits = 00b)

EDMCRAL functions as a 16-bit transfer counter.
The number of transfer operations is one when the setting is 0001h, and 65535 when it is FFFFh. The value is decremented by one each time data is transferred.
When the setting is 0000h, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running mode).
EDMCRAH is not used in normal transfer mode. Write 0000h to EDMCRAH.

(2) Repeat transfer mode (EXDMACn.EDMTMD.MD[1:0] bits = 01b)

EDMCRAH specifies the repeat size and EDMCRAL functions as a 10-bit transfer counter.
The number of transfer operations is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. In repeat transfer mode, a value in the range of 000h to 3FFh (the number of transfer operations: 1 to 1024) can be set for EDMCRAH and EDMCRAL.
Setting bits 15 to 10 in EDMCRAL is invalid. Write 0 to these bits.
The value in EDMCRAL is decremented by one each time data is transferred until it reaches 000h, at which the value in EDMCRAH is loaded into EDMCRAL.

(3) Block transfer mode (EXDMACn.EDMTMD.MD[1:0] bits = 10b)

EDMCRAH specifies the block size and EDMCRAL functions as a 10-bit block size counter.
The block size is 1 when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. In block transfer mode, a value in the range of 000h to 3FFh (the number of transfer operations: 1 to 1024) can be set for EDMCRAH and EDMCRAL.
Setting bits 15 to 10 in EDMCRAL is invalid. Write 0 to these bits.
The value in EDMCRAL is decremented by one each time data is transferred until it reaches 000h, at which the value in EDMCRAH is loaded into EDMCRAL.

(4) Cluster transfer mode (EXDMACn.EDMTMD.MD[1:0] bits = 11b)

EDMCRAH specifies the cluster size and EDMCRAL functions as a 3-bit cluster size counter.
The cluster size is 1 when the setting is 001h, 7 when it is 007h, and 8 when it is 000h. In cluster transfer mode, a value in the range of 000h to 007h (the number of transfer operations: 1 to 8) can be set for EDMCRAH and EDMCRAL.
Setting bits 15 to 3 in EDMCRAL is invalid. Write 0 to these bits.
The value in EDMCRAL is decremented by one each time data is transferred until it reaches 000h, at which the value in EDMCRAH is loaded into EDMCRAL.

19.2.4 EXDMA Block Transfer Count Register (EDMCRB)

Address(es): EXDMAC0.EDMCRB 0008 280Ch, EXDMAC1.EDMCRB 0008 284Ch

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

Bit	Description	Setting Range	R/W
b9 to b0	Specifies the block count of transfers in block transfer mode, the repeat count in repeat transfer mode, or the transfer cluster count in cluster transfer mode.	001h to 3FFh (1 to 1023) 000h (1024)	R/W
b15 to b10	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to specify the transfer block count in block transfer mode, the repeat count in repeat transfer mode, or the transfer cluster count in cluster transfer mode.

Decremented (by 1) when the last data of 1 repeat size is transferred in repeat transfer mode.

Decremented (by 1) when the last data of 1 block size is transferred in block transfer mode.

Decremented (by 1) when the last data of 1 cluster size is transferred in cluster transfer mode.

In normal transfer mode, EDMCRB is not used and setting this register is invalid.

19.2.5 EXDMA Transfer Mode Register (EDMTMD)

Address(es): EXDMAC0.EDMTMD 0008 2810h, EXDMAC1.EDMTMD 0008 2850h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	DCTG[1:0]	Transfer Request Source Select	b1 b0 0 0: Software 0 1: Setting prohibited 1 0: External DMA transfer request pin (EDREQn) 1 1: DMA transfer request from peripheral modules (TPU1.TRGA or MTU1.TRGA) *1	R/W
b7 to b2	—	Reserved	These bits are 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: Cluster transfer	R/W

Note 1. EXDMAC0: A software configurable interrupt B request from TPU1.TRGA selected in ICU.SLIBR144 or a software configurable interrupt A request from MTU1.TRGA selected in ICU.SLIAR208
EXDMAC1: A software configurable interrupt B request from TPU1.TRGA selected in ICU.SLIBR145 or a software configurable interrupt A request from MTU1.TRGA selected in ICU.SLIAR209
For the setting procedure, refer to section 19.5.1, Request Sources, (3).

This register is used to set DMA transfer mode.

DCTG[1:0] Bits (Transfer Request Source Select)

These bits select the startup source of the EXDMAC from software, the external DMA transfer request pin, or a DMA transfer request generated from a peripheral module.

SZ[1:0] Bits (Transfer Data Size Select)

These bits select data size for a single data transfer from 8, 16, and 32 bits.

DTS[1:0] Bits (Repeat Area Select)

DTS[1:0] select either the source or destination as the repeat area in repeat, block, or cluster transfer mode. In normal transfer mode, setting these bits is invalid.

MD[1:0] Bits (Transfer Mode Select)

These bits specify DMA transfer mode from normal, repeat, block, and cluster.

19.2.6 EXDMA Output Setting Register (EDMOMD)

Address(es): EXDMAC0.EDMOMD 0008 2812h, EXDMAC1.EDMOMD 0008 2852h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	DACKS	DACKE	DACKW	DACKSEL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DACKSEL	EDACKn Pin Toggling Select	0: EDACKn pin toggle is disabled. 1: EDACKn pin toggle is enabled.	R/W
b1	DACKW	EDACKn Pin Negate Wait	0: The EDACKn pin is negated at the same time as the RD# or WRn# pin is negated. 1: The EDACKn is negated one BCLK cycle before the RD# pin is negated or one BCLK cycle after the WRn# pin is negated.	R/W
b2	DACKE	EDACKn Pin Output Enable	0: EDACKn output is disabled. 1: EDACKn output is enabled.	R/W
b3	DACKS	EDACKn Pin Output Polarity Select	0: EDACKn pin polarity is active low. 1: EDACKn pin polarity is active high.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to set the output signal of the EXDMAC.

DACKSEL Bit (EDACKn Pin Toggling Select)

This bit is used to disable or enable toggled outputs on the EDACKn pin during normal, repeated, or block transfer to and from the SDRAM area in single-address mode.

If the DACKSEL bit is 0 during normal, repeated, or block transfer to and from the SDRAM area in single-address mode (i.e. while the EDMAMD.AMS bit is 1), the signal on the EDACKn pin is asserted throughout the interval where the data is valid (data-valid interval). If the DACKSEL bit is 1, the signal on the EDACKn pin is only asserted for one-half of the SDCLK period in the latter half of the data-valid interval.

The value of the DACKSEL pin has no effect in the case of CS areas.

The value of the DACKSEL pin also has no effect for transfer in dual-address mode and in cluster mode.

In these cases, there is no output on the EDACKn pin.

DACKW Bit (EDACKn Pin Negate Wait)

DACKW selects the EDACKn pin negation timing in single address mode during normal, repeat, or block transfer to/from the CS area.

During the above transfer in single address mode (EDMAMD.AMS bit = 1), the EDACKn pin is negated at the same time as the RD# or WRn# pin is negated if this bit is 0; and the EDACKn pin is negated one BCLK cycle before the RD# pin is negated, or one BCLK cycle after the WRn# pin is negated if this bit is 1. In the SDRAM area, setting this bit is invalid. EDACKn pin negation timing cannot be changed. Setting this bit is also invalid in dual address mode and during cluster transfer. In these cases, the EDACKn pin does not provide output.

DACKE Bit (EDACKn Pin Output Enable)

DACKE enables or disables EDACKn pin output. Setting this bit is invalid in dual address mode and during cluster transfer (EDACKn pin output not provided).

19.2.7 EXDMA Interrupt Setting Register (EDMINT)

Address(es): EXDMAC0.EDMINT 0008 2813h, EXDMAC1.EDMINT 0008 2853h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DARIE	Destination Address Extended Repeat Area Overflow Interrupt Enable	0: Disables an interrupt request for an extended repeat area overflow on the destination address 1: Enables an interrupt request for an extended repeat area overflow on the destination address	R/W
b1	SARIE	Source Address Extended Repeat Area Overflow Interrupt Enable	0: Disables an interrupt request for an extended repeat area overflow on the source address 1: Enables an interrupt request for an extended repeat area overflow on the source address	R/W
b2	RPTIE	Repeat Size End Interrupt Enable	0: Disables the repeat size end interrupt request. 1: Enables the repeat size end interrupt request.	R/W
b3	ESIE	Transfer Escape End Interrupt Enable	0: Disables the transfer escape end interrupt request. 1: Enables the transfer escape end interrupt request.	R/W
b4	DTIE	Transfer End Interrupt Enable	0: Disables the transfer end interrupt request. 1: Enables the transfer end interrupt request.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to set the interrupt request output of the EXDMAC.

DARIE Bit (Destination Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the destination address occurs while this bit is set to 1, the EDMCNT.DTE bit is set to 0 (DMA transfer is disabled). At the same time, the EDMSTS.ESIF flag is set to 1 to indicate that an interrupt by an extended repeat area overflow on the destination address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 (DMA transfer is enabled) in the EDMCNT.DTE bit of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the destination address, this bit is ignored.

SARIE Bit (Source Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the source address occurs while this bit is set to 1, the DTE bit in EDMCNT is set to 0 (DMA transfer is disabled). At the same time, the EDMSTS.ESIF flag is set to 1 to indicate that an interrupt by an extended repeat area overflow on the source address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 (DMA transfer is enabled) in the EDMCNT.DTE bit of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the source address, this bit is ignored.

RPTIE Bit (Repeat Size End Interrupt Enable)

When this bit is set to 1 in repeat transfer mode, the EDMCNT.DTE bit is set to 0 (DMA transfer is disabled) after completion of a 1-repeat size data transfer. At the same time, the EDMSTS.ESIF flag is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the EDMTMD.DTS[1:0] bits are 10b (repeat area or block area is not specified).
When this bit is set to 1 in block transfer mode, the EDMCNT.DTE bit is set to 0 after completion of a 1-block data transfer in the same way as repeat transfer mode. At the same time, the EDMSTS.ESIF flag is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the EDMTMD.DTS[1:0] bits are 10b.
When this bit is set to 1 in cluster transfer mode, the EDMCNT.DTE bit is set to 0 after completion of a 1-cluster data transfer in the same way as repeat transfer mode. At the same time, the EDMSTS.ESIF flag is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the EDMTMD.DTS[1:0] bits are 10b.

ESIE Bit (Transfer Escape End Interrupt Enable)

This bit enables or disables the transfer escape end interrupt requests (repeat size end interrupt request and extended repeat area overflow interrupt request) that are generated during DMA transfer.
The transfer escape end interrupt is generated when the ESIF flag in EXDMACn.EDMSTS is set to 1 with this bit set to1. The transfer escape end interrupt is cleared by setting this bit or the EDMSTS.ESIF flag to 0.

DTIE Bit (Transfer End Interrupt Enable)

This bit enables or disables the transfer end interrupt request to be generated on completion of a specified number of data transfers.
The transfer end interrupt is generated when the EDMSTS.DTIF flag is set to 1 with this bit set to 1. The transfer end interrupt is cleared by setting this bit or the EDMSTS.DTIF flag to 0.

19.2.8 EXDMA Address Mode Register (EDMAMD)

Address(es): EXDMAC0.EDMAMD 0008 2814h, EXDMAC1.EDMAMD 0008 2854h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AMS	DIR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SM[1:0]		—	SARA[4:0]					DM[1:0]		—	DARA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	DARA[4:0]	Destination Address Extended Repeat Area Set	Specifies the extended repeat area on the destination address. For details on the settings, refer to Table 19.3.	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 Set	b7 b6 0 0: Destination address is fixed. 0 1: Offset addition*1 1 0: Destination address is incremented. 1 1: Destination address is decremented.	R/W
b12 to b8	SARA[4:0]	Source Address Extended Repeat Area Set	Specifies the extended repeat area on the source address. For details on the settings, refer to Table 19.3.	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 Set	b15 b14 0 0: Destination address is fixed. 0 1: Offset addition*1 1 0: Destination address is incremented. 1 1: Destination address is decremented.	R/W
b16	DIR	Single Address Direction Select	0: Data is transferred in single address mode using the EDMSAR register value as the transfer source address. EDACKn is output to the transfer destination. 1: Data is transferred in single address mode using the EDMDAR register value as the transfer destination address. EDACKn is output to the transfer destination.	R/W
b17	AMS	Address Mode Select	0: Dual address mode 1: Single address mode	R/W
b31 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Offset addition can be specified only for EXDMAC0.

This register is used to set address mode of the EXDMACn.

DARA[4:0] Bits (Destination Address Extended Repeat Area Set)

These bits specify the extended repeat area on the destination address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 21 (2 bytes) and 217 (128 Mbytes).

When the lower address overflows the extended repeat area by address increment, the top address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the bottom address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer destination, do not specify the extended repeat area on the destination address. When repeat transfer, block transfer, or cluster transfer is selected, or when EXDMACn.EDMTMD.DTS[1:0] = 00b (the transfer destination is specified as the repeat area or block area), write 00000b in the DARA[4:0] bits.

An interrupt can be requested when an overflow occurs in the extended repeat area with the DARIE bit in EDMINT set to 1. Table 19.3 lists the settings and the corresponding extended repeat areas.

DM[1:0] Bits (Destination Address Update Mode Set)

These bits select the mode of updating the destination address.
When increment is selected and the SZ[1:0] bits in EDMTMD are set to 00b, 01b, and 10b, the destination address is incremented by 1, 2, and 4, respectively.
When decrement is selected and the SZ[1:0] bits in EDMTMD are set to 00b, 01b, and 10b, the destination address is decremented by 1, 2, and 4, respectively.
When offset addition is selected, the offset specified by the EXDMAC0.EDMOFR register is added to the address. Offset addition can be specified only for EXDMAC0.

SARA[4:0] Bits (Source Address Extended Repeat Area Set)

These bits specify the extended repeat area on the source address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 21 (2 bytes) and 217 (128 Mbytes).
When the lower address overflows the extended repeat area by address increment, the top address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the bottom address of the extended repeat area is set.
When the repeat area or block area is specified as a transfer destination, do not specify the extended repeat area on the destination address. When repeat transfer, block transfer, or cluster transfer is selected, or when EXDMACn.EDMTMD.DTS[1:0] = 01b (the transfer destination is specified as the repeat area or block area), write 00000b in the SARA[4:0] bits.
An interrupt can be requested when an overflow occurs in the extended repeat area with the SARIE bit in EDMINT set to 1. Table 19.3 lists the settings and the corresponding extended repeat areas.

SM[1:0] Bits (Source Address Update Mode Set)

These bits select the mode of updating the source address.
When increment is selected and the SZ[1:0] bits in EDMTMD are set to 00b, 01b, and 10b, the destination address is incremented by 1, 2, and 4, respectively.
When decrement is selected and the SZ[1:0] bits in EDMTMD are set to 00b, 01b, and 10b, the destination address is decremented by 1, 2, and 4, respectively.
When offset addition is selected, the offset specified by the EXDMAC0.EDMOFR register is added to the address. Offset addition can be specified only for EXDMAC0.

DIR Bit (Single Address Direction Select)

This bit selects the transfer destination or source, to which the addresses should be output in single address mode.

- Normal/Repeat/Block Transfer

When this bit is set to 0, data is transferred in single address mode using the EDMSAR register value as the transfer source address. Here, EDACKn can be output to the transfer destination device by setting the DACKE bit in EDMOMD to 1 (EDACKn output is enabled).
When this bit is set to 1, data is transferred in single address mode using the EDMDAR register value as the transfer destination address. Here, EDACKn can be output to the transfer source device by setting the DACKE bit in EDMOMD to 1 (EDACKn output is enabled). Setting the DIR bit is valid when the AMS bit in EDMAMD is 1 (single address mode); setting the DIR bit is invalid when the AMS bit is 0 (dual address mode).

- Cluster Transfer

When this bit is set to 0, data is transferred in cluster transfer read address mode using the EDMSAR register value as the transfer source address. Here, data can be transferred to the cluster buffers from the external device.
When this bit is set to 1, data is transferred in cluster transfer write address mode using the EDMDAR register value as

the transfer destination address. Here, data can be transferred to the external device from the cluster buffers. Setting the DIR bit is valid when the AMS bit in EDMAMD is 1 (single address mode); setting the DIR bit is invalid when the AMS bit is 0 (dual address mode).

AMS Bit (Address Mode Select)

This bit selects the address mode.

- Normal/Repeat/Block Transfer

When this bit is set to 0, dual address mode is selected and when set to 1, single address mode is selected.
When using single address mode, select the transfer source or destination device to which the addresses should be output using the DIR bit in EDMAMD.

- Cluster Transfer

When this bit is set to 0, dual address mode is selected and when set to 1, read or write address mode is selected. Select read or write address mode using the DIR bit in EDMAMD.

Table 19.3 Settings and Range of Extended Repeat Areas

SARA[4:0] or DARA[4:0]	Extended Repeat Area
00000b	Not specified
00001b	2 bytes specified as extended repeat area by the lower 1 bit of the address
00010b	4 bytes specified as extended repeat area by the lower 2 bits of the address
00011b	8 bytes specified as extended repeat area by the lower 3 bits of the address
00100b	16 bytes specified as extended repeat area by the lower 4 bits of the address
00101b	32 bytes specified as extended repeat area by the lower 5 bits of the address
00110b	64 bytes specified as extended repeat area by the lower 6 bits of the address
00111b	128 bytes specified as extended repeat area by the lower 7 bits of the address
01000b	256 bytes specified as extended repeat area by the lower 8 bits of the address
01001b	512 bytes specified as extended repeat area by the lower 9 bits of the address
01010b	1 Kbyte specified as extended repeat area by the lower 10 bits of the address
01011b	2 Kbytes specified as extended repeat area by the lower 11 bits of the address
01100b	4 Kbytes specified as extended repeat area by the lower 12 bits of the address
01101b	8 Kbytes specified as extended repeat area by the lower 13 bits of the address
01110b	16 Kbytes specified as extended repeat area by the lower 14 bits of the address
01111b	32 Kbytes specified as extended repeat area by the lower 15 bits of the address
10000b	64 Kbytes specified as extended repeat area by the lower 16 bits of the address
10001b	128 Kbytes specified as extended repeat area by the lower 17 bits of the address
10010b	256 Kbytes specified as extended repeat area by the lower 18 bits of the address
10011b	512 Kbytes specified as extended repeat area by the lower 19 bits of the address
10100b	1 Mbyte specified as extended repeat area by the lower 20 bits of the address
10101b	2 Mbytes specified as extended repeat area by the lower 21 bits of the address
10110b	4 Mbytes specified as extended repeat area by the lower 22 bits of the address
10111b	8 Mbytes specified as extended repeat area by the lower 23 bits of the address
11000b	16 Mbytes specified as extended repeat area by the lower 24 bits of the address
11001b	32 Mbytes specified as extended repeat area by the lower 25 bits of the address
11010b	64 Mbytes specified as extended repeat area by the lower 26 bits of the address
11011b	128 Mbytes specified as extended repeat area by the lower 27 bits of the address
11100b to 11111b	Setting prohibited

19.2.9 EXDMA Offset Register (EDMOFR)

Address(es): EXDMAC0.EDMOFR 0008 2818h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Description	Setting Range	R/W
b31 to b0	Specifies the offset when offset addition is selected as the address update mode for transfer source or destination.	0000 0000h to 00FF FFFFh (0 bytes to (16M – 1) bytes) FF00 0000h to FFFF FFFFh (–16M bytes to –1 byte)	R/W

This register is used to set the address offset value.

Write to this register while the EXDMAC operation is stopped or DMA transfer is disabled (not during data transfer).

Setting bits 31 to 25 is invalid; a value of bit 24 is extended to bits 31 to 25. Reading EDMOFR returns the extended value.

19.2.10 EXDMA Transfer Enable Register (EDMCNT)

Address(es): EXDMAC0.EDMCNT 0008 281Ch, EXDMAC1.EDMCNT 0008 285Ch

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

Bit	Symbol	Bit Name	Description	R/W
b0	DTE	DMA Transfer Enable	0: Disables DMA transfer. 1: Enables DMA transfer.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register enables or disables the DMA transfer to the corresponding channel.

DTE Bit (DMA Transfer Enable)

When the EDMAST.DMST bit is set to 1 (EXDMAC module start) and this bit is set to 1 (DMA transfer is enabled), DMA transfer can be started for the corresponding channel.

When the DTE bit is set to 0 during DMA transfer, DMA transfer is suspended after completion of the current data transfer corresponding to a single transfer request. DMA transfer is resumed by setting the DTE bit to 1 again.

While the DTE bit is 1, writing to registers other than the DTE bit of the EXDMAC channel is prohibited.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When 0 is written to this bit.
- When the specified total volume of data transfer is completed.
- When DMA transfer is stopped by the repeat size end interrupt.
- When DMA transfer is stopped by the extended repeat area overflow interrupt.

19.2.11 EXDMA Software Start Register (EDMREQ)

Address(es): EXDMAC0.EDMREQ 0008 281Dh, EXDMAC1.EDMREQ 0008 285Dh

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

Bit	Symbol	Bit Name	Description	R/W
b0	SWREQ	DMA Software Start	0: DMA transfer is not requested. 1: DMA transfer is requested.	R/W
b3 to b1	—	Reserved	These bits are 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

This register is used to execute the DMA by software.

SWREQ Bit (DMA Software Start)

When 1 is written to this bit, a DMA transfer request is generated. After DMA transfer is started in response to the request, this bit is set to 0 if the CLRS bit is set to 0. This bit is not set to 0 while the CLRS bit is set to 1. In this case, a DMA transfer request can be issued again after completion of a transfer.

Note that, however, setting this bit is valid and DMA transfer by software is enabled only when the DCTG[1:0] bits in EDMTMD are set to 00b (DMA request source is software).

Setting this bit is invalid when the DCTG[1:0] bits in EDMTMD are set to a value other than 00b.

To start DMA transfer by software with CLRS set to 0, check that the SWREQ bit is 0 and then write 1 to SWREQ.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When a DMA transfer request by software is accepted and DMA transfer is started while the CLRS bit is set to 0 (the SWREQ bit is cleared after DMA transfer is started by software).
- When 0 is written to this bit.

CLRS Bit (DMA Software Start Bit Auto Clear Select)

This bit specifies whether to clear the SWREQ bit to 0 after DMA transfer is started in response to the DMA transfer request generated by setting the SWREQ bit to 1. With this bit set to 0, the SWREQ bit is set to 0 after DMA transfer is started. With this bit set to 1, the SWREQ bit is not set to 0. In this case, a DMA transfer request can be issued again after completion of a transfer.

19.2.12 EXDMA Status Register (EDMSTS)

Address(es): EXDMAC0.EDMSTS 0008 281Eh, EXDMAC1.EDMSTS 0008 285Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	ACT	—	—	DTIF	—	—	—	ESIF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESIF	Transfer Escape End Interrupt Flag	0: A transfer escape end interrupt has not been generated. 1: A transfer escape end interrupt has been generated.	R/W
b3 to b1	—	Reserved	These bits are 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 read as 0. Writing to these bits has no effect.	R
b7	ACT	EXDMA Active Flag	0: EXDMAC operation is suspended. 1: EXDMAC is operating.	R

This register indicates the state of the DMA.

ESIF Flag (Transfer Escape End Interrupt Flag)

This flag indicates that the transfer escape end interrupt has been generated.

[Setting conditions]

- When 1-repeat size data transfer is completed in repeat transfer mode with the EDMINT.RPTIE bit set to 1.
- When 1-block data transfer is completed in block transfer mode with the EDMINT.RPTIE bit set to 1.
- When 1-cluster data transfer is completed in cluster transfer mode with the EDMINT.RPTIE bit set to 1.
- When an extended repeat area overflow on the source address occurs while the EDMINT.SARIE bit is set to 1 and the EDMAMD.SARA[4:0] bits are set to a value other than 00000b (extended repeat area is specified on the transfer source address)
- When an extended repeat area overflow on the destination address occurs while the EDMINT.DARIE bit is set to 1 and the EDMAMD.DADR[4:0] bits are set to a value other than 00000b (extended repeat area is specified on the transfer destination address)

[Clearing conditions]

- When 0 is written to this bit.
- When 1 is written to the EDMCNT.DTE bit.

DTIF Flag (Transfer End Interrupt Flag)

This flag indicates that the transfer end interrupt has been generated.

[Setting conditions]

- When the specified number of unit-transfers are completed in normal transfer mode (the value of EDMCRAL becoming 0 on completion of transfer)
- When the specified number of repeat transfer operations are completed in repeat transfer mode (the value of EDMCRB becoming 0 on completion of transfer)
- When the specified number of blocks have been transferred in block transfer mode (the value of EDMCRB becoming 0 on completion of transfer)
- When the specified number of clusters have been transferred in cluster transfer mode (the value of EDMCRB becoming 0 on completion of transfer)

[Clearing conditions]

- When 0 is written to this bit
- When 1 is written to the DTE bit in EDMCNT

ACT Flag (EXDMA Active Flag)

- This flag indicates whether the EXDMACn is in the idle or active state.

[Setting condition]

- When the EXDMACn starts data transfer operation

[Clearing condition]

- When data transfer in response to one transfer request is completed

19.2.13 EXDMA External Request Sense Mode Register (EDMRMD)

Address(es): EXDMAC0.EDMRMD 0008 2820h, EXDMAC1.EDMRMD 0008 2860h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	DREQS[1:0]	Request Input Sense Mode Set	b1 b0 0 0: Rising edge 0 1: Falling edge 1 0: Low level 1 1: (Setting prohibited)	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to set sense mode of the EDREQn pin.

DREQS[1:0] Bits (Request Input Sense Mode Set)

These bits specify the sense mode for the external DMA transfer request signal (EDREQn pin).

19.2.14 EXDMA External Request Flag Register (EDMERF)

Address(es): EXDMAC0.EDMERF 0008 2821h, EXDMAC1.EDMERF 0008 2861h

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

Bit	Symbol	Bit Name	Description	R/W
b0	EREQ	External Request Flag	0: No request 1: Requested	R/(W) *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 0 has no effect.

This register detects a request generated by the EDREQn pin.

EREQ Flag (External Request Flag)

This flag indicates the DMA transfer request from the external DMA transfer request signal (the EDREQn pin).

[Setting conditions]

- When the level on the EDREQn pin changes from 0 to 1 while EXDMACn.EDMRMD.DREQS[1:0] = 00b
- When the level on the EDREQn pin changes from 1 to 0 while EXDMACn.EDMRMD.DREQS[1:0] = 01b
- When the level on the EDREQn pin is 0 while EXDMACn.EDMRMD.DREQS[1:0] = 10b (low level)

[Clearing conditions]

- When the DMA transfer is started while EXDMACn.EDMRMD.DREQS[1:0] = 00b (rising edge) or 01b (falling edge) and then the DMA transfer is started
- When 1 is written to this flag while EXDMACn.EDMRMD.DREQS[1:0] = 00b (rising edge) or 01b (falling edge)
- When the EDREQn pin is set to 1 while EXDMACn.EDMRMD.DREQS[1:0] = 10b (low level)

19.2.15 EXDMA Peripheral Request Flag Register (EDMPRF)

Address(es): EXDMAC0.EDMPRF 0008 2822h, EXDMAC1.EDMPRF 0008 2862h

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

Bit	Symbol	Bit Name	Description	R/W
b0	PREQ	Peripheral Module Request Flag	0: No request 1: Requested	R/(W) *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 0 has no effect.

This register detects a DMA transfer request generated by a peripheral module.

PREQ Flag (Peripheral Module Request Flag)

This flag detects the DMA transfer request from the peripheral modules.

[Setting condition]

- When the DMA transfer request is generated from the peripheral modules

[Clearing conditions]

- When the DMA transfer request is generated from the peripheral modules and the DMA transfer is started
- When 1 is written to this flag

19.2.16 EXDMAC Module Start Register (EDMAST)

Address(es): 0008 2A00h

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

Bit	Symbol	Bit Name	Description	R/W
b0	DMST	EXDMAC Module Start	0: EXDMAC module stop 1: EXDMAC module start	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register enables or disables the startup of all channels of the EXDMAC.

DMST Bit (EXDMAC Module Start)

When this bit is set to 1, EXDMAC is ready to accept transfer requests for all the channels.
When 1 is written to the DTE bit in EDMCNT (DMA transfer is enabled) of all the EXDMACn channels and then this bit is set to 1, all the channels can be placed in the transfer request acceptable state at the same time.
When the DMST bit is set to 0 during DMA transfer, DMA transfer for all channels is suspended after completion of the current data transfer corresponding to a single transfer request. DMA transfer is resumed by setting the DMST bit to 1 again.
[Setting condition]

- When 1 is written to this bit

[Clearing condition]

- When 0 is written to this bit

19.2.17 Cluster Buffer Register y (CLSBRy) (y = 0 to 7)

Address(es): CLSBR0 0008 2BE0h, CLSBR1 0008 2BE4h, CLSBR2 0008 2BE8h, CLSBR3 0008 2BECh, CLSBR4 0008 2BF0h, CLSBR5 0008 2BF4h, CLSBR6 0008 2BF8h, CLSBR7 0008 2BFCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Description	R/W
b31 to b0	Buffer area for cluster transfer.	R/W

CLSBRy are buffer registers for cluster transfer.
During cluster transfer, transferred data is sequentially stored in CLSBRy starting from CLSBR0. The cluster-transferred data or data written by the CPU is retained until another cluster transfer or data write by the CPU. When reading the cluster-transferred data with the CPU, confirm that cluster transfer has been completed and only refer to the data of the specified size for cluster; the other data is invalid.
During cluster transfer, the same CLSBRy is used for all the channels. If a conflict occurs between the write to CLSBRy by the CPU and cluster transfer, transferred data is not guaranteed. If a channel is set to cluster transfer in read or write address mode and another channel is set to cluster transfer, data to be transferred may be erroneously modified.
Data is stored in cluster buffers in the different manner depending on the transfer size setting (EDMTMD.SZ[1:0] bits).

(1) Transfer Size is 8 Bits (EXDMACn.EDMTMD.SZ[1:0] = 00b)

Data is stored in the lower 8 bits in the cluster buffers. Here, the upper 24 bits are invalid. When the maximum cluster size is set to 8, 8-byte data is one cluster.
Data is stored in CLSBR in the order of CLSBR0 to CLSBRj (j = cluster size value – 1).

(2) Transfer Size is 16 Bits (EXDMACn.EDMTMD.SZ[1:0] = 01b)

Data is stored in the lower 16 bits in the cluster buffers. Here, the upper 16 bits are invalid. When the maximum cluster size is set to 8, 16-byte data is one cluster.
Data is stored in CLSBR in the order of CLSBR0 to CLSBRj (j = cluster size value – 1).

(3) Transfer Size is 32 Bits (EXDMACn.EDMTMD.SZ[1:0] = 10b)

Data is stored in all the 32 bits in the cluster buffers. When the maximum cluster size is set to 8, 32-byte data is one cluster.
Data is stored in CLSBR in the order of CLSBR0 to CLSBRj (j = cluster size value – 1).

19.3 Operation

19.3.1 Transfer Mode

(1) Normal Transfer Mode

In normal transfer mode, one data is transferred by one transfer request. A maximum of 65535 can be set as the number of transfer operations using EDMCRA of EXDMACn. When the EXDMACn.EDMCRAL bits are set to 0000h, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running mode). Setting EDMCRB of EXDMACn is invalid in normal transfer mode. Except in free running mode, a transfer end interrupt request can be generated after completion of the specified number of transfer operations.
Table 19.4 summarizes the register update operation in normal transfer mode, and Figure 19.2 shows the operation in normal transfer mode.

Table 19.4 Register Update Operation in Normal Transfer Mode

Register	Function	Update Operation after Completion of a Transfer by One Transfer Request
EXDMACn.EDMSAR	Transfer source address	Increment/decrement/fixed/offset addition*1
EXDMACn.EDMDAR	Transfer destination address	Increment/decrement/fixed/offset addition*1
EXDMACn.EDMCRAL	Transfer counter	Decremented by one/not updated (in free running mode)
EXDMACn.EDMCRAH	—	Not updated (Not used in normal transfer mode)
EXDMACn.EDMCRB	—	Not updated (Not used in normal transfer mode)

Note 1. Offset addition can be specified only for EXDMAC0.

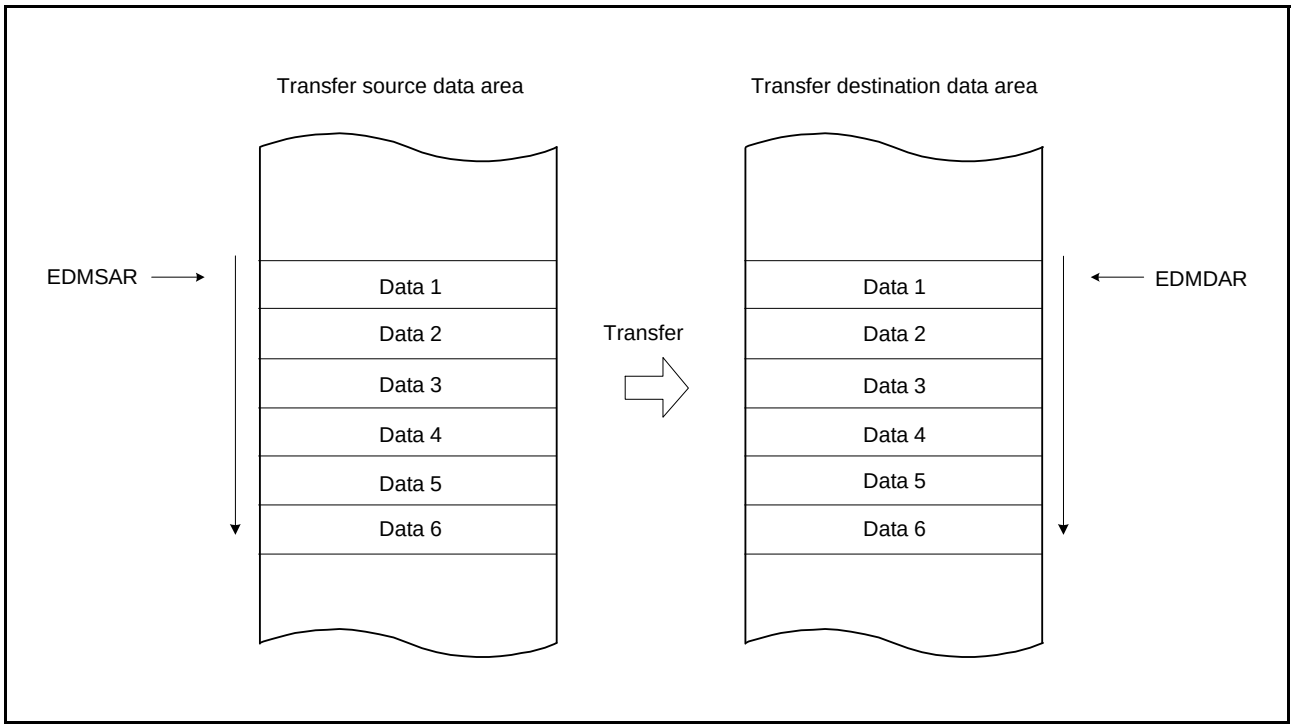


Figure 19.2 Operation in Normal Transfer Mode

(2) Repeat Transfer Mode

In repeat transfer mode, one data is transferred by one transfer request.
A maximum of 1K data can be set as a total repeat transfer size using EDMCRA of the EXDMACn.
A maximum of 1K can be set as the number of repeat transfer operations using EDMCRB of the EXDMACn; therefore, a maximum of 1K data × 1K = 1M can be set as a total data transfer size.
Either the transfer source or transfer destination can be specified as a repeat area. When transfer of the repeat size data is completed, the address of the specified repeat area (EDMSAR or EDMDAR of the EXDMACn) returns to the transfer start address. When data of the specified repeat size has all been transferred in repeat transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DTE bit in EDMCNT of EXDMACn in the repeat size end interrupt handling.
A transfer end interrupt request can be generated after completion of the specified number of repeat transfer operations.
Table 19.5 summarizes the register update operation in repeat transfer mode, and Figure 19.3 shows the operation in repeat transfer mode.

Table 19.5 Register Update Operation in Repeat Transfer Mode

		Update Operation after Completion of a Transfer by One Transfer Request	
Register	**Function**	**When EXDMACn.EDMCRAL is not 1**	**When EXDMACn.EDMCRAL is 1 (Transfer of the Last Data in Repeat Size)**
EXDMACn.EDMSAR	Transfer source address	Increment/decrement/fixed/ offset addition*1	• EXDMACn.EDMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition*1 • EXDMACn.EDMTMD.DTS[1:0] = 01b Initial value of EXDMACn.EDMSAR • EXDMACn.EDMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
EXDMACn.EDMDAR	Transfer destination address	Increment/decrement/fixed/ offset addition*1	• EXDMACn.EDMTMD.DTS[1:0] = 00b Initial value of EXDMACn.EDMDAR • EXDMACn.EDMTMD.DTS[1:0] = 01b Increment/decrement/fixed/offset addition*1 • EXDMACn.EDMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
EXDMACn.EDMCRAH	Repeat size	Not updated	Not updated
EXDMACn.EDMCRAL	Transfer counter	Decremented by one	EXDMACn.EDMCRAH
EXDMACn.EDMCRB	Repeat-count counter	Not updated	Decremented by one

Note 1. Offset addition can be specified only for EXDMAC0.

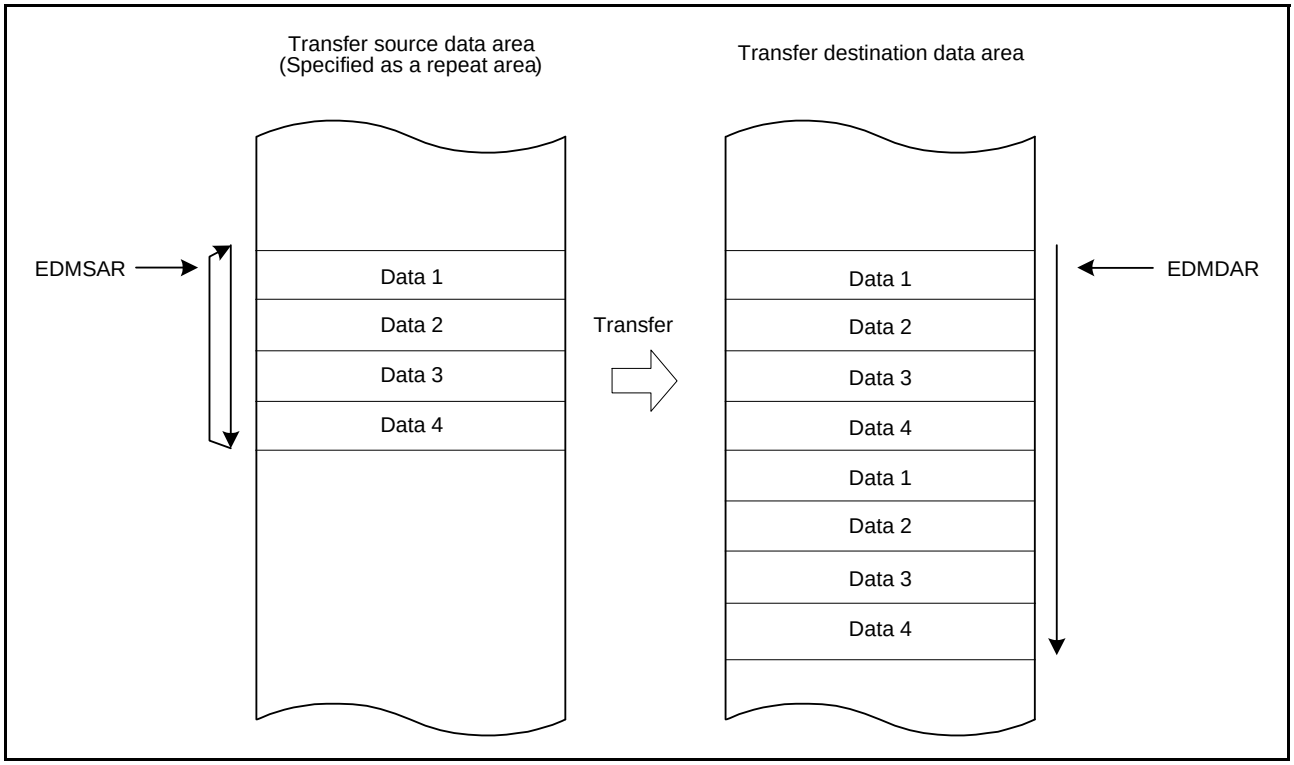


Figure 19.3 Operation in Repeat Transfer Mode

(3) Block Transfer Mode

In block transfer mode, a single block data is transferred by one transfer request.

A maximum of 1K data can be set as a total block transfer size using EDMCRA of the EXDMACn.

A maximum of 1K can be set as the number of block transfer operations using EDMCRB of the EXDMACn; therefore, a maximum of 1K data × 1K blocks = 1M can be set as a total data transfer size.

Either the transfer source or transfer destination can be specified as a block area. When transfer of a single block data is completed, the address of the specified block area (EDMSAR or EDMDAR of the EXDMACn) returns to the transfer start address. When a single block data has all been transferred in block transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DTE bit in EDMCNT of EXDMACn in the repeat size end interrupt handling.

Transfer end interrupt request can be generated after completion of the specified number of block transfer operations.

Table 19.6 summarizes the register update operation in block transfer mode, and Figure 19.4 shows the operation in block transfer mode.

Table 19.6 Register Update Operation in Block Transfer Mode

Register	Function	Update Operation after Completion of Single-Block Transfer by One Transfer Request
EXDMACn.EDMSAR	Transfer source address	• EXDMACn.EDMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition*1 • EXDMACn.EDMTMD.DTS[1:0] = 01b Initial value of EXDMACn.EDMSAR • EXDMACn.EDMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
EXDMACn.EDMDAR	Transfer destination address	• EXDMACn.EDMTMD.DTS[1:0] = 00b Initial value of EXDMACn.EDMDAR • EXDMACn.EDMTMD.DTS[1:0] = 01b Increment/decrement/fixed/offset addition*1 • EXDMACn.EDMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
EXDMACn.EDMCRAH	Block size	Not updated
EXDMACn.EDMCRAL	Block size counter	EXDMACn.EDMCRAH
EXDMACn.EDMCRB	Transfer-block counter	Decremented by one

Note 1. Offset addition can be specified only for EXDMAC0.

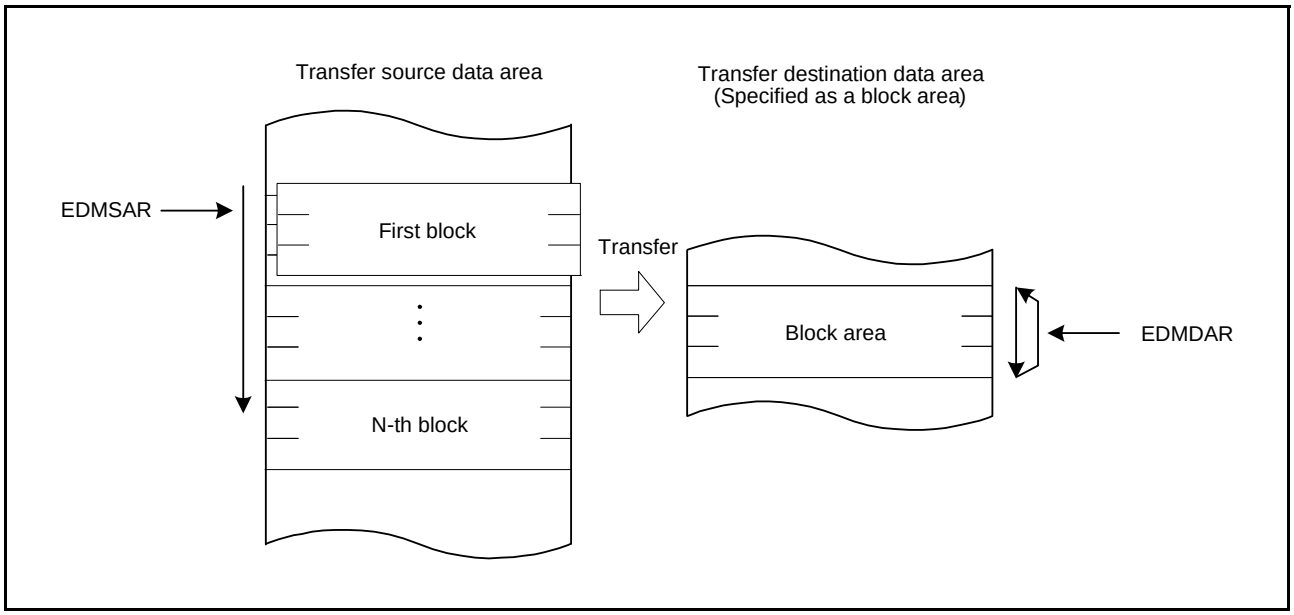


Figure 19.4 Operation in Block Transfer Mode

(4) Cluster Transfer Mode

In cluster transfer mode, a single cluster data is transferred by one transfer request. A maximum of 8 data can be set as a total cluster transfer size using EDMCRA of the EXDMACn.
A maximum of 1K can be set as the number of cluster transfer operations using EDMCRB of the EXDMACn; therefore, a maximum of 8 data × 1K = 8K can be set as a total data transfer size.
The cluster transfer mode can be selected from among cluster transfer dual address mode, cluster transfer read address mode, and cluster transfer write address mode.

- Cluster transfer dual address mode
  (EXDMACn.EDMTMD.MD[1:0] = 11b, EXDMACn.EDMAMD.AMS = 0)
  A single cluster data is transferred by one transfer request from the transfer source address to the cluster buffers. A single cluster data is then transferred from the cluster buffers to the transfer destination address.
- Cluster transfer read address mode
  (EXDMACn.EDMTMD.MD[1:0] = 11b, EXDMACn.EDMAMD.AMS = 1, EXDMACn.EDMAMD.DIR = 0)
  A single cluster data is transferred by one transfer request from the transfer source address to the cluster buffers.
- Cluster transfer write address mode
  (EXDMACn.EDMTMD.MD[1:0] = 11b, EXDMACn.EDMAMD.AMS = 1, EXDMACn.EDMAMD.DIR = 1)
  A single cluster data is transferred by one transfer request from the cluster buffers to the transfer destination address.

In cluster-transfer mode, DMA transfer stops on completion of the transfer of each cluster of data, and a repeat-size-completed interrupt request can be generated. DMA transfer can be restarted by writing 1 to the EXDMACn.EDMCNT.DTE bit during processing of the repeat-size-completed interrupt.
A repeat-size-completed interrupt request can also be generated on completion of transfer of clusters the specified number of times.

Table 19.7 summarizes the register update operation in cluster transfer mode, and Figure 19.5 shows the operation in cluster transfer mode.

Table 19.7 Register Update Operation in Cluster Transfer Mode (Dual Address Mode)

Register	Function	Update Operation after Completion of Single-Cluster Transfer by One Transfer Request
EXDMACn.EDMSAR	Transfer source address	• EXDMACn.EDMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition*1 • EXDMACn.EDMTMD.DTS[1:0] = 01b Initial value of EXDMACn.EDMSAR • EXDMACn.EDMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
EXDMACn.EDMDAR	Transfer destination address	• EXDMACn.EDMTMD.DTS[1:0] = 00b Initial value of EXDMACn.EDMDAR • EXDMACn.EDMTMD.DTS[1:0] = 01b Increment/decrement/fixed/offset addition*1 • EXDMACn.EDMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1
EXDMACn.EDMCRAH	Cluster size	Not updated
EXDMACn.EDMCRAL	Cluster size counter	EXDMACn.EDMCRAH
EXDMACn.EDMCRB	Transfer cluster count counter	Decremented by one

Note 1. Offset addition can be specified only for EXDMAC0.
In read address mode, the transfer destination address EXDMACn.EDMDAR is fixed (invalid).
In write address mode, the transfer source address EXDMACn.EDMSAR is fixed (invalid).

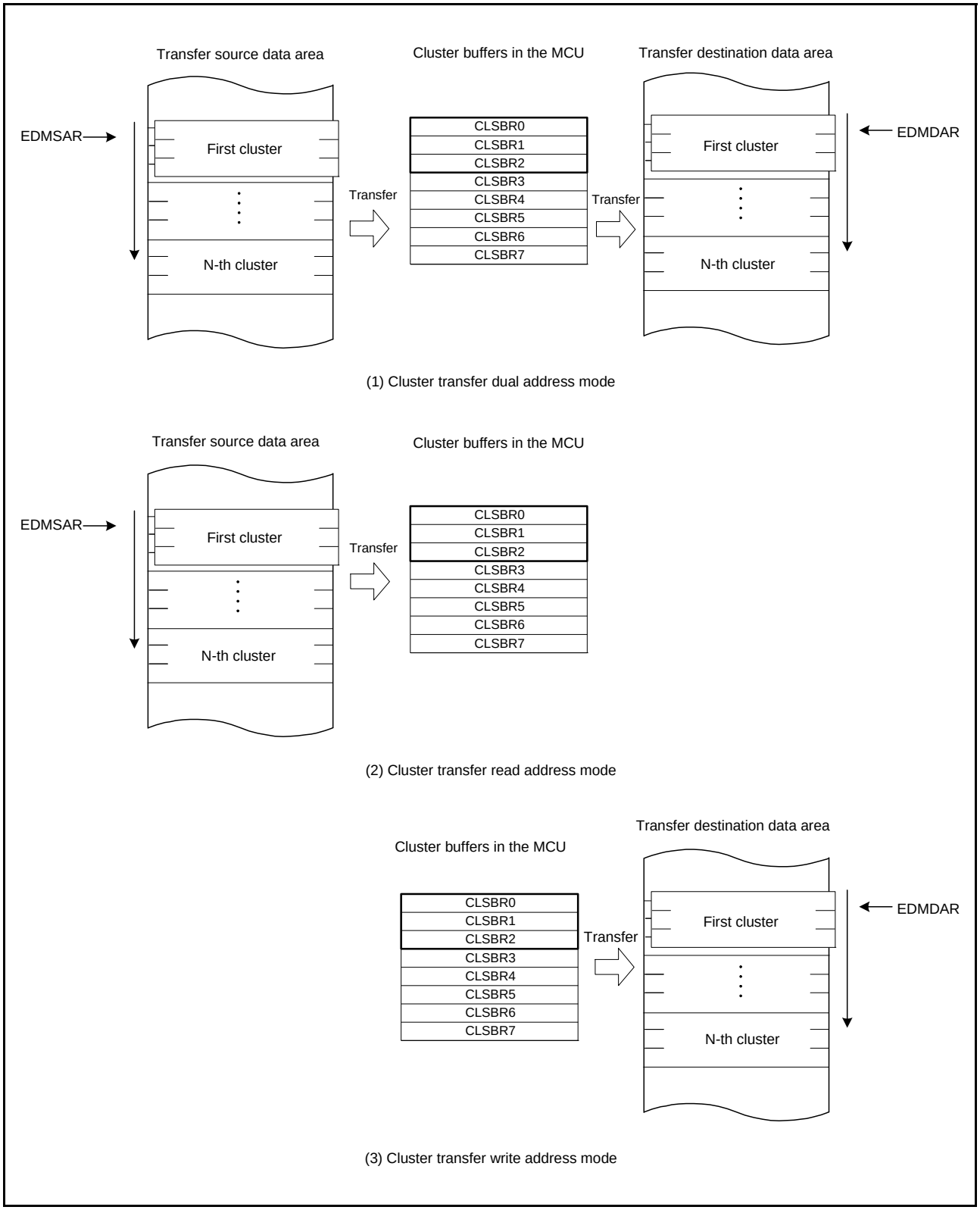


Figure 19.5 Operation in Cluster Transfer Mode

19.3.2 Extended Repeat Area Function

The EXDMAC supports a function to specify the extended repeat areas on the transfer source and destination addresses. With the extended repeat areas set, the address registers repeatedly indicate the addresses of the specified extended repeat areas.

The extended repeat areas can be specified separately to the transfer source address register (EDMSAR) and transfer destination address register (EDMDAR) of EXDMACn.

The extended repeat area on the source address is specified by the SARA[4:0] bits in EDMAMD of EXDMACn. The extended repeat area on the destination address is specified by the DARA[4:0] bits in EDMAMD of EXDMACn. The size can be specified separately for the source and destination sides. However, do not specify a repeat area or block area that is also an extended repeat area.

When the address register value reaches the end address of the extended repeat area and the extended repeat area overflows, DMA transfer is stopped and an interrupt by an extended repeat area overflow can be requested. When an overflow occurs in the extended repeat area on the transfer source while the SARIE bit in EDMINT of EXDMACn is set to 1, the ESIF flag in EDMSTS of EXDMACn is set to 1 and the DTE bit in EDMINT of EXDMACn is set to 0 to stop DMA transfer. At this time, if the ESIE bit in EDMINT of EXDMACn is set to 1, an interrupt by an extended repeat area overflow is requested.

When the DARIE bit in EDMINT of EXDMACn is set to 1, an overflow on the extended repeat area set in EDMDAR occurs, meaning that the destination side is a target. DMA transfer can be resumed by writing 1 to the DTE bit in EDMCNT of EXDMACn in the extended repeat area overflow interrupt handling.

Figure 19.6 shows an example of the extended repeat area operation.

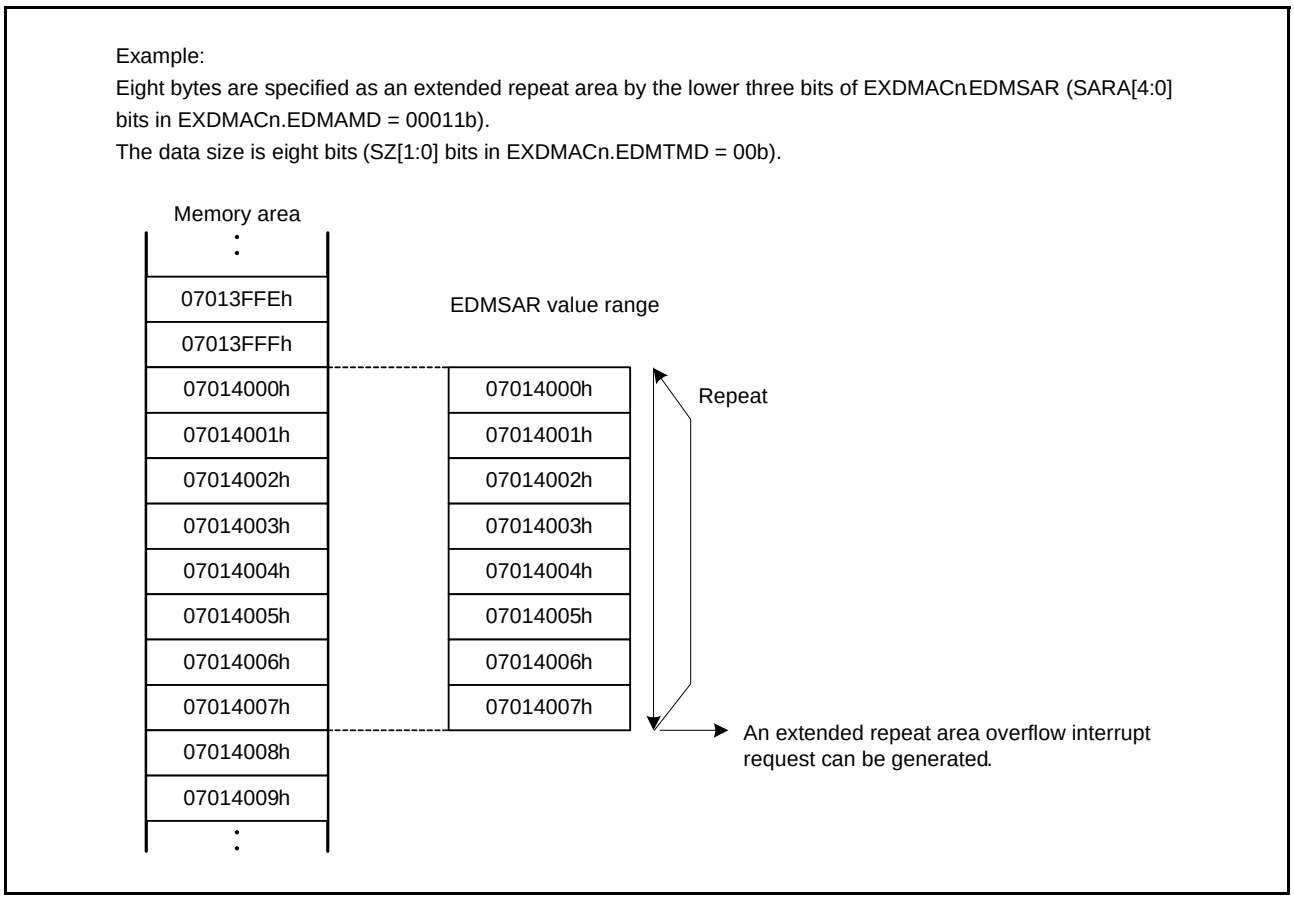


Figure 19.6 Example of Extended Repeat Area Operation

When an interrupt by an extended repeat area overflow is used in block transfer mode or cluster transfer mode, the following should be taken into consideration.
When a transfer is stopped by an interrupt by an extended repeat area overflow, the address register must be set so that the block size (or cluster size) is a power of 2 or the block size (or cluster size) boundary is aligned with the extended repeat area boundary. When an overflow on the extended repeat area occurs during a transfer of one block (or one cluster), the interrupt by the overflow is suspended until transfer of the block (or the cluster) is completed, and the transfer overruns.

Figure 19.7 shows an example when the extended repeat area function is used in block transfer mode.

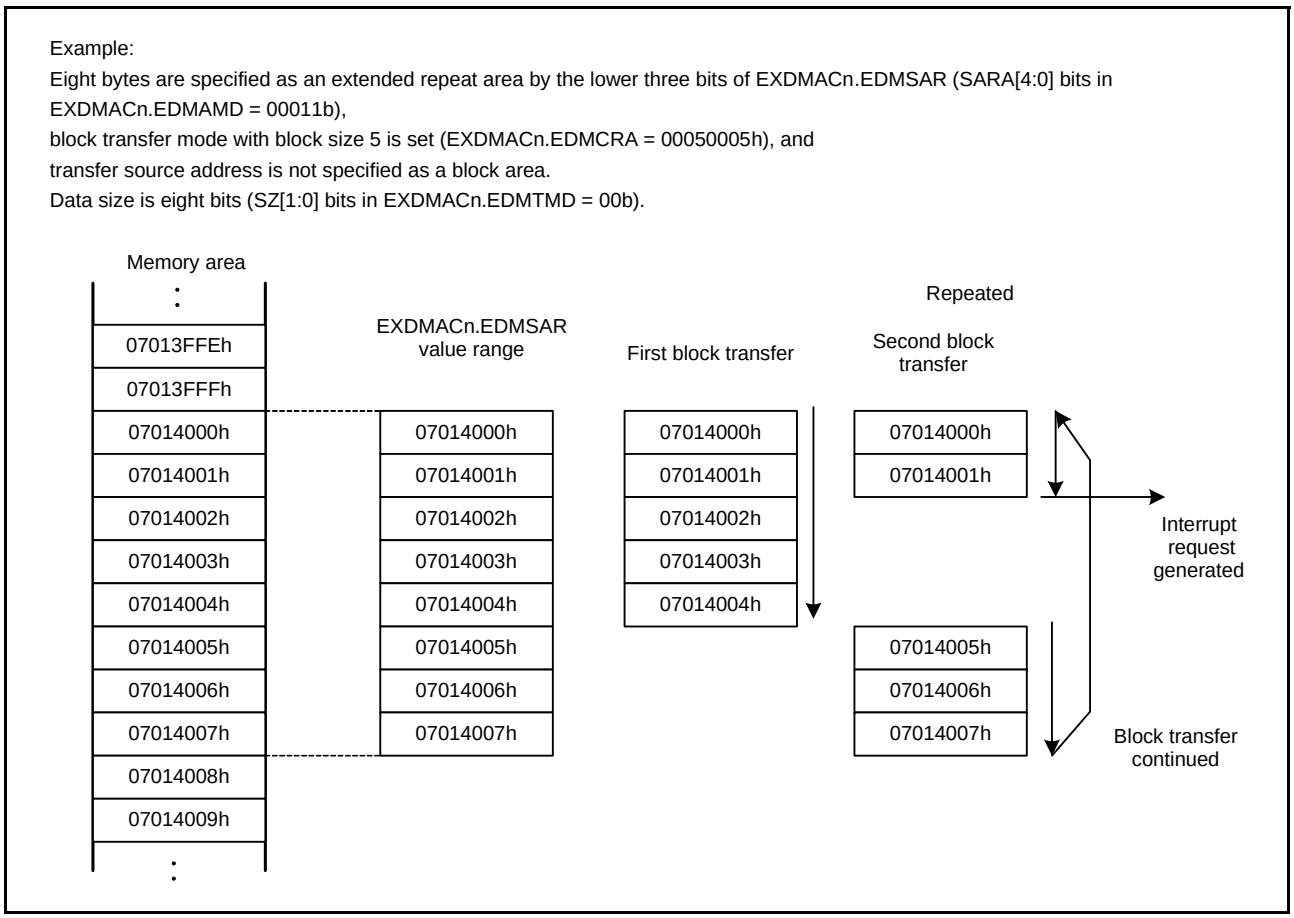


Figure 19.7 Example of Extended Repeat Area Function in Block Transfer Mode

19.3.3 Address Update Function using Offset

The source and destination addresses can be updated by fixing, increment, decrement, or offset addition. When the offset addition is selected, the offset specified by the EXDMA offset register (EDMOFR of EXDMAC0) is added to the address every time the EXDMAC performs one data transfer. This function realizes a data transfer where addresses are allocated to separated areas.

Offset subtraction can also be realized by setting a negative value in EDMOFR of EXDMAC0. In this case, the negative value must be 2's complement.

Address update function using offset can be specified only for the EXDMAC0 channel.

Table 19.8 lists the address update method in each address update mode.

Table 19.8 Address Update Method in Each Address Update Mode

Address Update Mode	Settings of EXDMACn.EDMAMD.SM[1:0] and EXDMACn.EDMAMD.DM[1:0] for Address Update Modes	Address Update Method (for Different SZ[1:0] Settings in EDMTMD of EXDMACn)		
		SZ[1:0] = 00b	SZ[1:0] = 01b	SZ[1:0] = 10b
Address fixed	00b	Fixed		
Offset addition	01b	+EXDMAC0.EDMOFR*1		
Increment	10b	+1	+2	+4
Decrement	11b	–1	–2	–4

Note 1. When setting a negative value in the EXDMA offset register, the value must be 2’s complement. The 2's complement is obtained by the following formula.
2’s complement of a negative offset value = ~ (offset) + 1 (~: bit inversion)

(1) Basic Transfer Using Offset Addition

Figure 19.8 shows an example of address updating using offset addition.

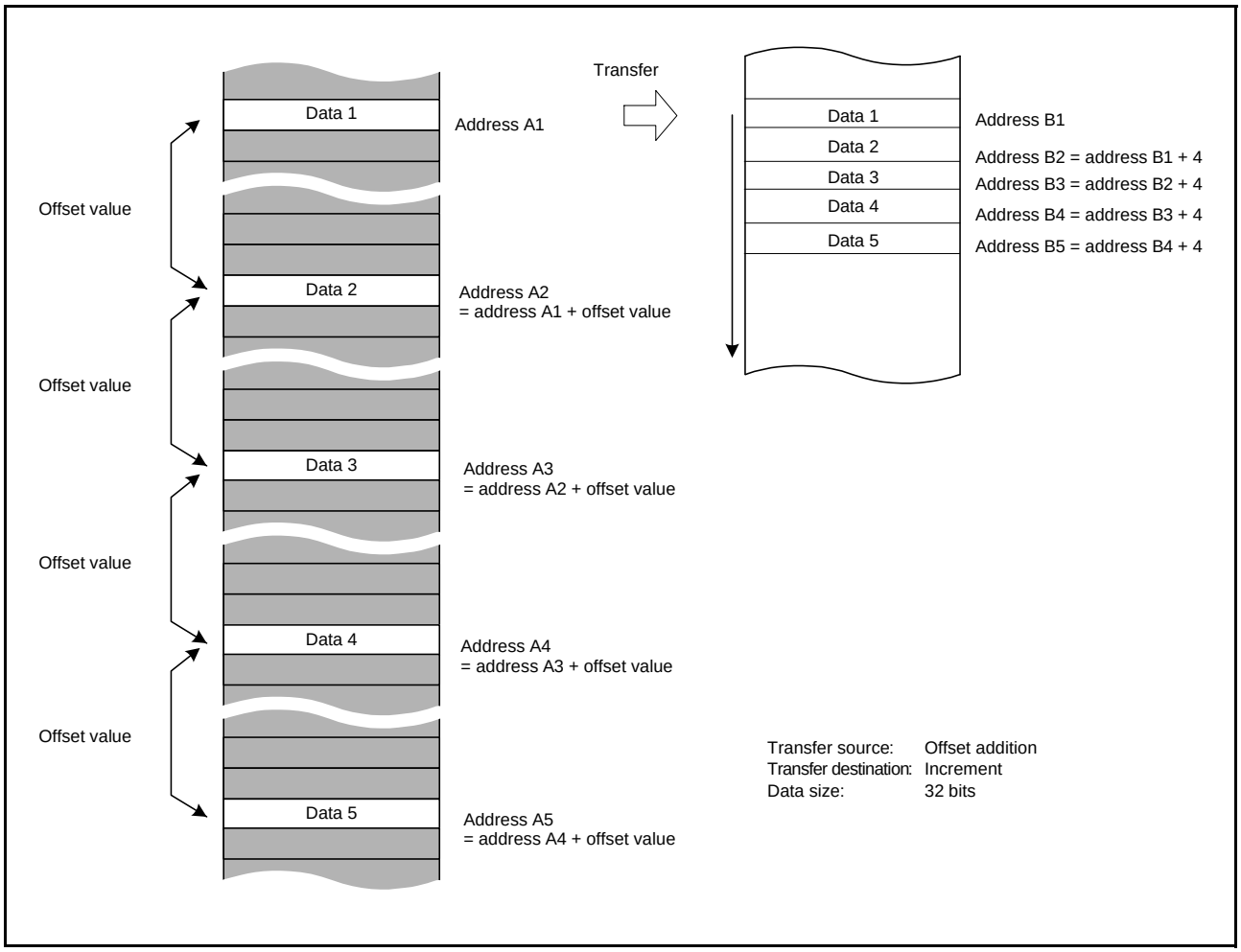


Figure 19.8 Example of Address Updating by Offset Addition

In Figure 19.8, the transfer data is 32 bits long, and offset addition and increment are set as the transfer source address update mode and transfer destination address update mode, respectively. The second and subsequent data is each read from the transfer source address obtained by adding the offset value to the previous address. The data read from the addresses at the specified intervals is written to the continuous locations on the destination.

(2) Example of XY Conversion Using Offset Addition

Figure 19.9 shows the XY conversion using offset addition in repeat transfer mode.
The settings are as follows.

- EXDMAC0.EDMAMD register: Source address update mode (offset addition)
- EXDMAC0.EDMAMD register: Source address update mode (incremented)
- EXDMAC0.EDMTMD register: Transfer data size select (32-bit transfer)
- EXDMAC0.EDMTMD register: Transfer mode select (repeat transfer)
- EXDMAC0.EDMTMD register: Repeat area select (the source is specified as the repeat area)
- EXDMAC0.EDMOFR register: Address offset setting (10h)
- EXDMAC0.EDMCRA register: The number of repeat transfer setting (4h)
- EXDMAC0.EDMINT register: Repeat size end interrupt enable

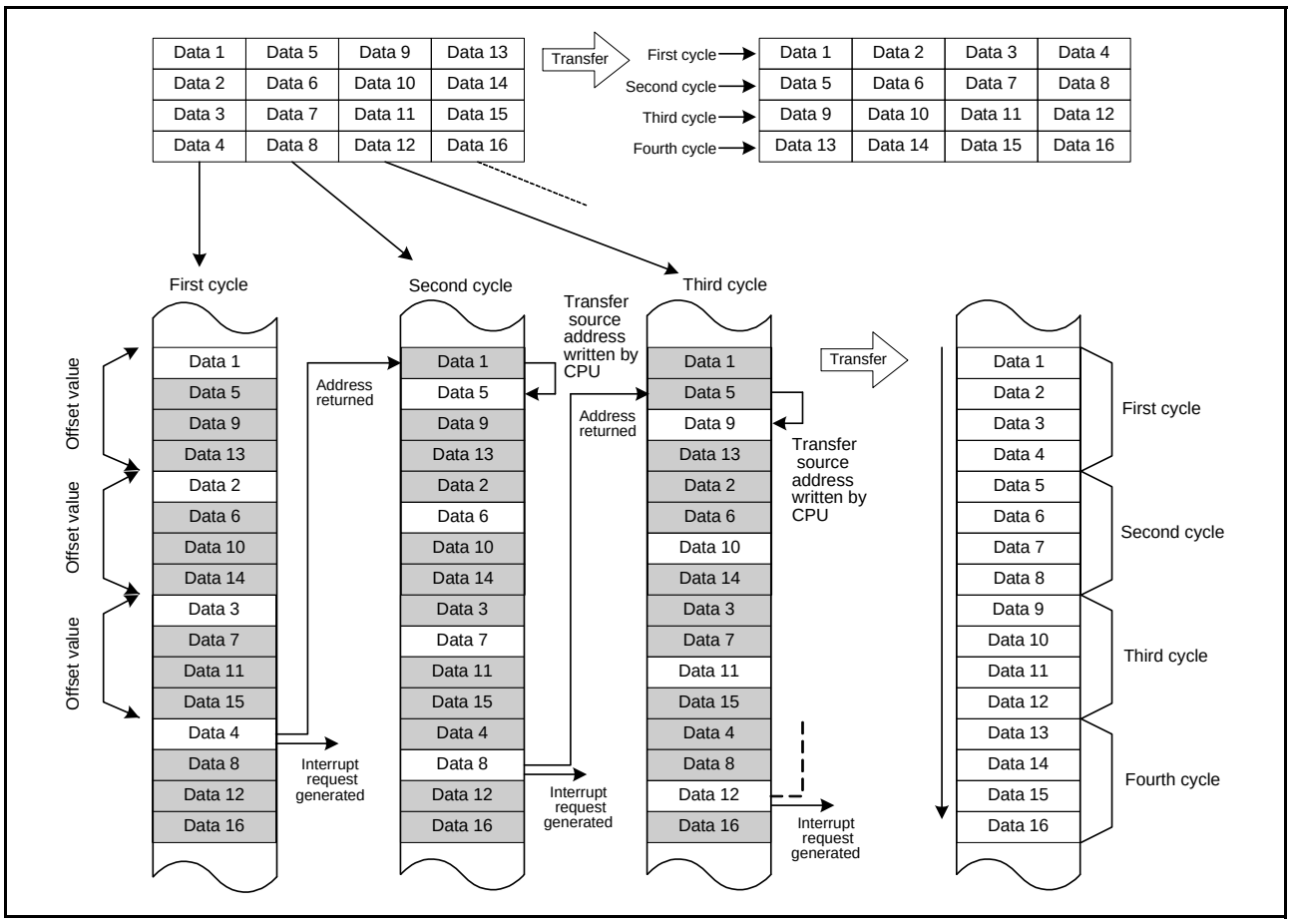


Figure 19.9 XY Conversion Operation Using Offset Addition in Repeat Transfer Mode

When a transfer starts, the offset value is added to the transfer source address every time data is transferred. The transfer data is written to the destination continuous addresses. When data 4 is transferred, which means that the repeat size of transfers is completed, EXDMAC returns the transfer source address to the transfer start address (address of data 1 on the transfer source) and a repeat size end interrupt is requested. While this interrupt stops the transfer temporarily, perform the following.

- EXDMAC0.EDMSAR: Rewrite the DMA transfer source address to the address of data 5 (with the above example, the data 1 address + 4).
- EXDMAC0. EDMCNT: Set the DTE bit to 1.

The DMA transfer is resumed from the state when the DMA transfer is stopped. After that, the operations described above are repeated until the transfer source data is transposed to the destination area (XY conversion)

Figure 19.10 shows a flowchart of the XY conversion.

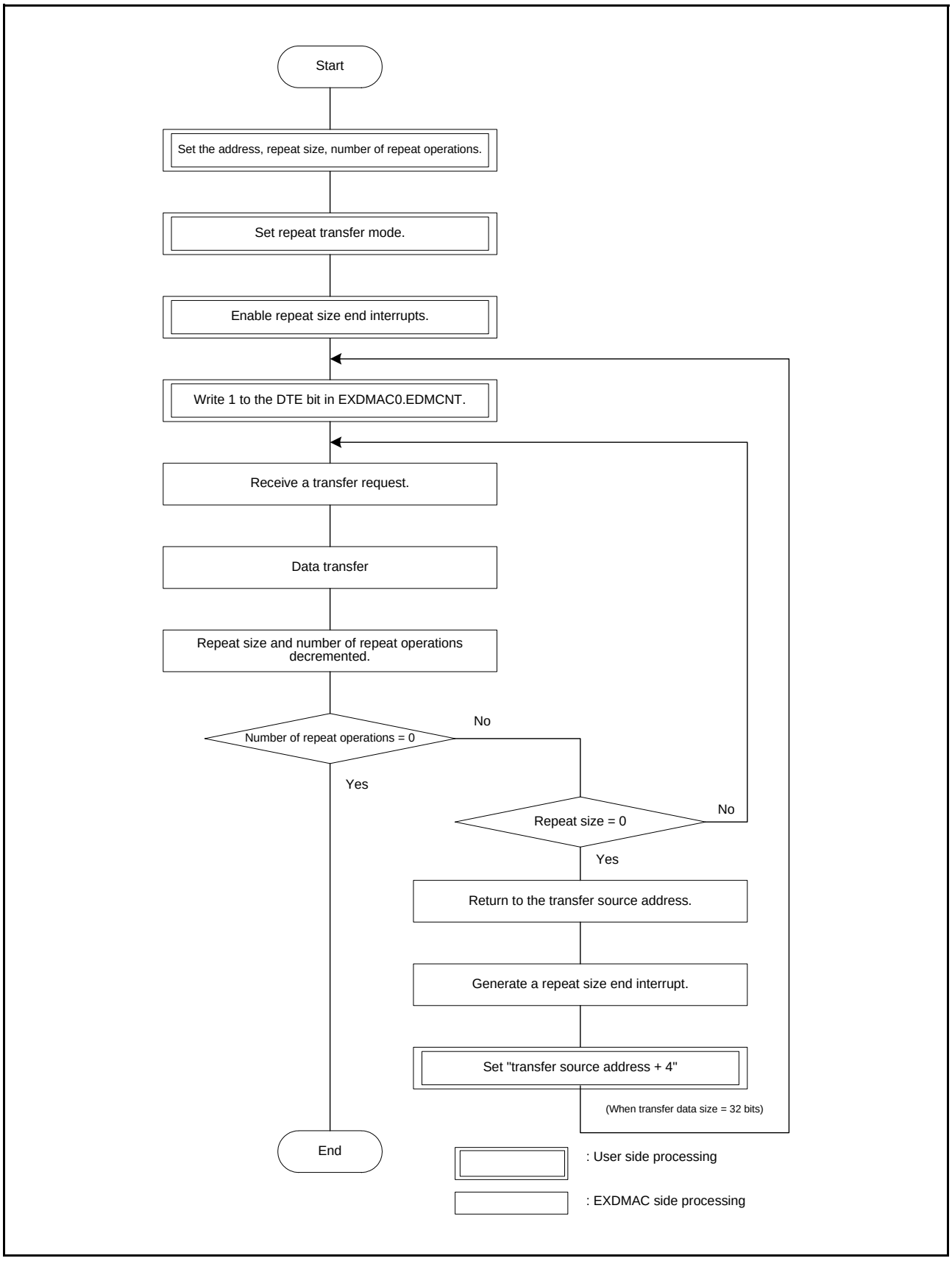


Figure 19.10 XY Conversion Flowchart Using Offset Addition in Repeat Transfer Mode

19.3.4 Address Modes

The EXDMAC provides dual and single address modes (dual, read, and write address modes in cluster transfer), either of which can be selectable. Table 19.9 lists the relationship between transfer modes and address modes.

Table 19.9 Relationship between Transfer Modes and Address Modes

Transfer Mode	Address Mode	Single Address Direction	EXDMAC Operation
Normal transfer mode (EDMTMD.MD[1:0] = 00b)	Dual address mode (EDMAMD.AMS = 0)	—	Reads and then writes data.
	Single address mode (EDMAMD.AMS = 1)	Transfer source (EDMAMD.DIR = 0)	Only reads data; outputs EDACKn to the device to be written to. This MCU receives no read data.
		Transfer destination (EDMAMD.DIR = 1)	Only writes data; outputs EDACKn to the device to be read from. This MCU outputs no write data.
Repeat transfer mode (EDMTMD.MD[1:0] = 01b)	Dual address mode (EDMAMD.AMS = 0)	—	Reads and then writes data.
	Single address mode (EDMAMD.AMS = 1)	Transfer source (EDMAMD.DIR = 0)	Only reads data; outputs EDACKn to the device to be written to. This MCU receives no read data.
		Transfer destination (EDMAMD.DIR = 1)	Only writes data; outputs EDACKn to the device to be read from. This MCU outputs no write data.
Block transfer mode (EDMTMD.MD[1:0] = 10b)	Dual address mode (EDMAMD.AMS = 0)	—	Reads and writes data alternately for every data transfer specified by EDMTMD.SZ[1:0] (transfer data size)
	Single address mode (EDMAMD.AMS = 1)	Transfer source (EDMAMD.DIR = 0)	Only reads data; outputs EDACKn to the device to be written to. This MCU receives no read data.
		Transfer destination (EDMAMD.DIR = 1)	Only writes data; outputs EDACKn to the device to be read from. This MCU outputs no write data.
Cluster transfer mode (EDMTMD.MD[1:0] = 11b)	Dual address mode (EDMAMD.AMS = 0)	—	Reads data of cluster size and then writes data of cluster size.
	Read address mode (EDMAMD.AMS = 1)	Transfer source (EDMAMD.DIR = 0)	Only reads data of cluster size. Transfers data to the cluster buffers.
	Write address mode (EDMAMD.AMS = 1)	Transfer destination (EDMAMD.DIR = 1)	Only writes data of cluster size. Transfers data from the cluster buffers.

19.4 Transfer Operation

Descriptions of examples of operations in transfer by the EXDMAC are given in the following passages.
Operations of the EXDMAC are synchronized by the external bus clock (BCLK). The examples that follow are for cases where the external bus clock (BCLK) and the signal output on the BCLK pin are at the same frequency unless there is a particular reason for doing otherwise.

Note: An idle cycle is inserted between two adjacent accesses depending on the external bus recovery cycle setting. One idle cycle, however, is inserted between a read access and a write access even if the recovery cycle is set to 0. Refer to section 16, Buses, for details on the recovery cycle.

19.4.1 Normal/Repeat Transfer Operation

(1) Dual Address Mode

Figure 19.11 shows the bus cycle example in normal-transfer dual address mode. In the example, a 16-bit data (SZ[1:0] = 01b in EDMTMD of EXDMACn) is transferred from a device with 16-bit 2-cycle access to another device with 16-bit 2-cycle access, which is started at the falling edge of EDREQn.
The bus cycles in repeat-transfer dual address mode are the same as those in normal-transfer dual address mode.

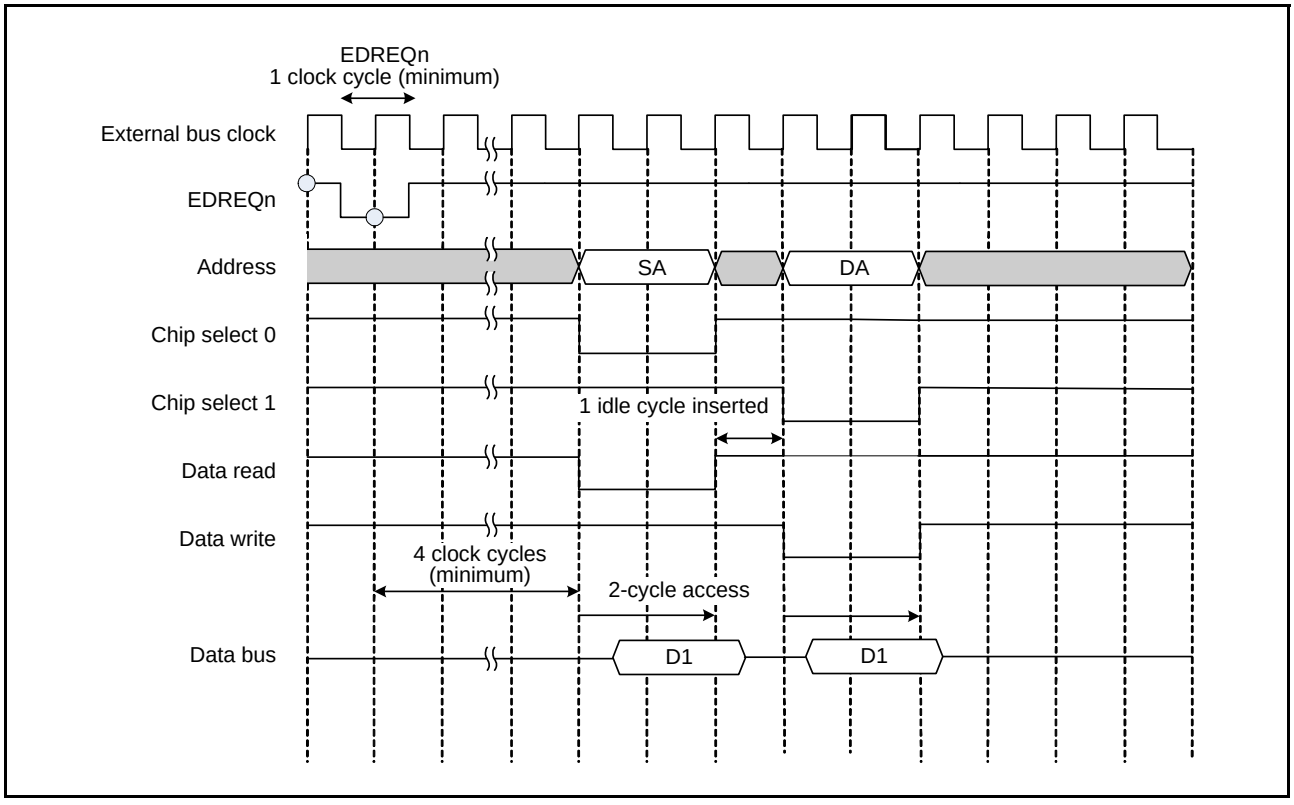


Figure 19.11 Bus Cycle Example in Normal-Transfer Dual Address Mode

(2) Single Address Mode

In single address mode, data is read from the transfer source address and directly transferred to the transfer destination device without being taken in the MCU. Here, EDACKn is output to one of the external transfer-destination and transfer-source devices, and the address is simultaneously output to the other transfer device for access.
With the DIR bit in EDMAMD of EXDMACn set to 0, the transfer source address is output to the external bus and EDACKn is output to the transfer destination. With the DIR bit in EDMAMD of EXDMACn set to 1, the transfer destination address is output to the external bus and the EDACKn is output to the transfer source. Figure 19.12 shows the data flow in single address mode.

Figure 19.13 shows the bus cycle example in normal-transfer single address mode. In the example, one data is transferred in 2-cycle access when the DIR bit in EDMAMD of EXDMACn is set to 1 (transfer destination address is output) and when set to 0 (transfer source address is output).

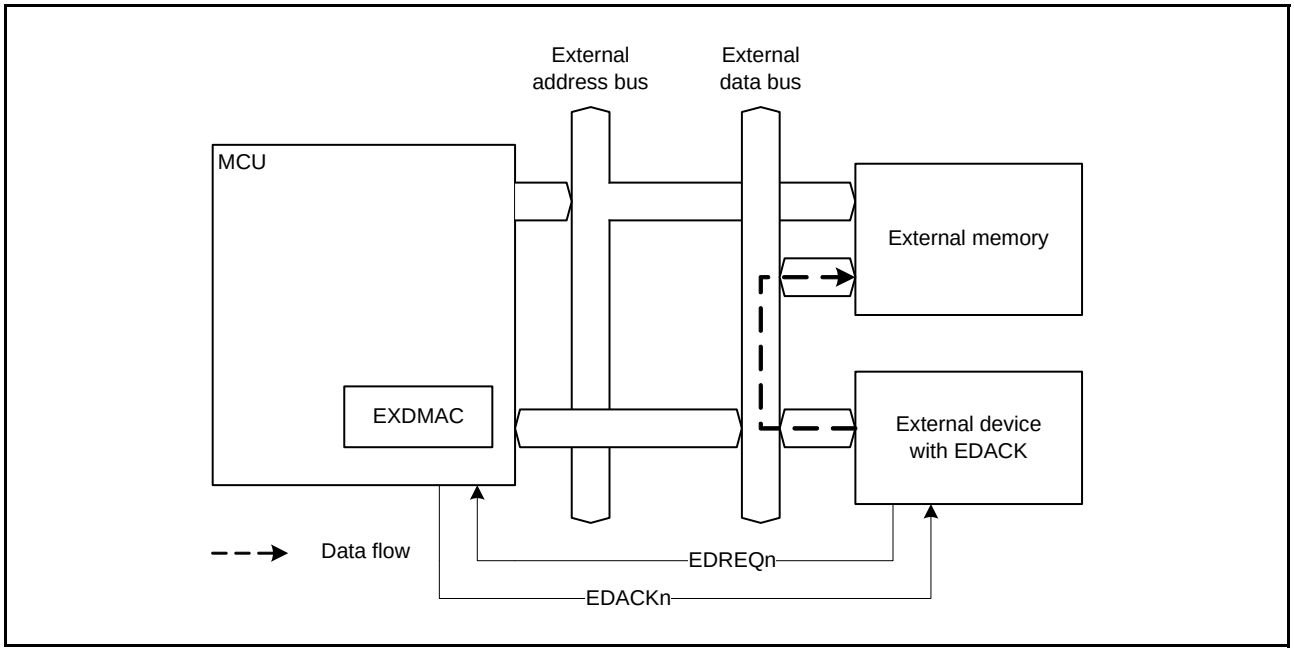


Figure 19.12 Data Flow in Single Address Mode (when EDMAMD.DIR = 1)

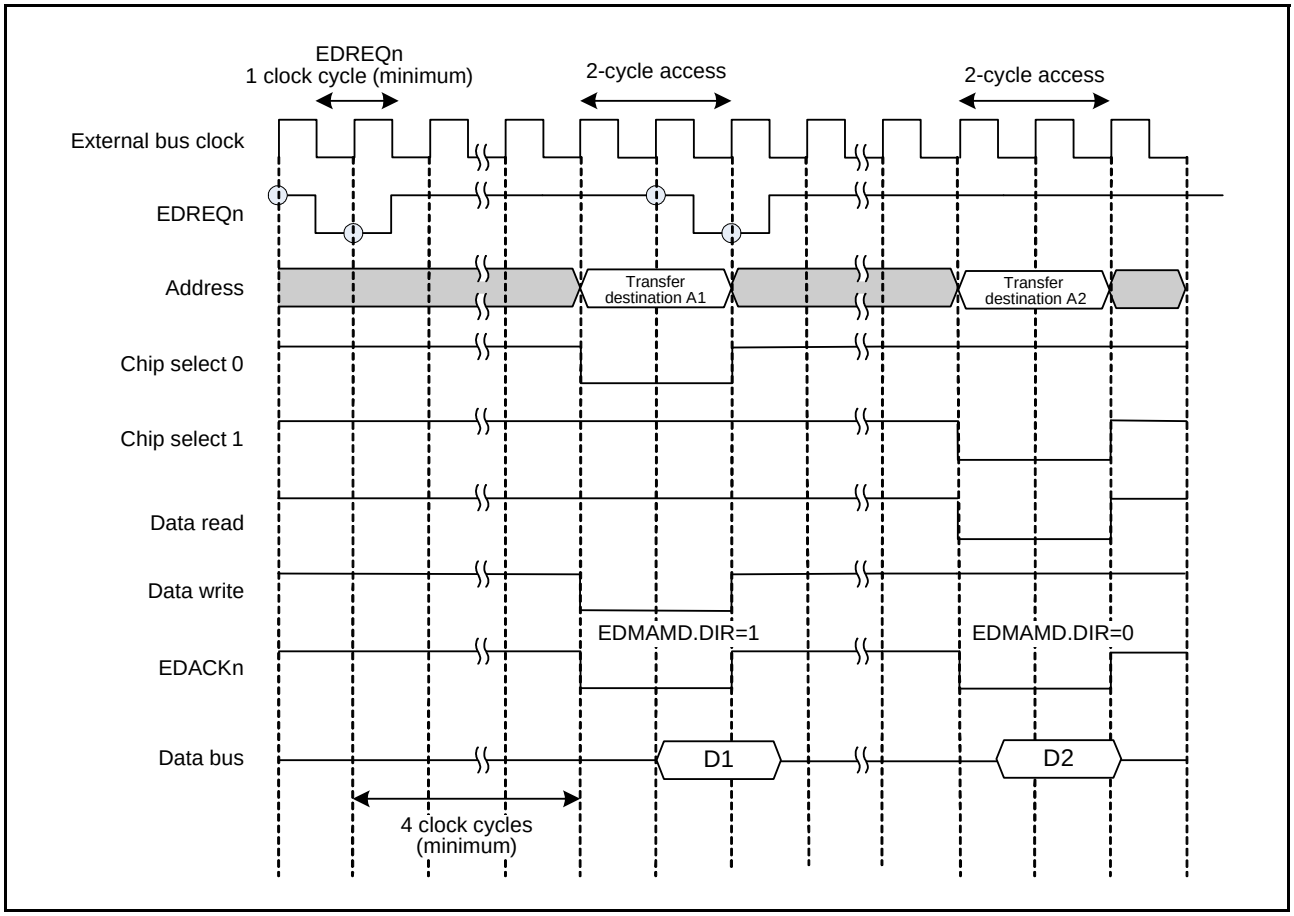


Figure 19.13 Bus Cycle Example in Normal-Transfer Single Address Mode

19.4.2 Block Transfer Operation

(1) Dual Address Mode

Figure 19.14 shows the bus cycle example in block-transfer dual address mode. In the example, a 16-bit data (SZ[1:0] = 01 in EDMTMD of EXDMACn) is transferred from a device with 16-bit 2-cycle access to another device with 16-bit 2-cycle access when the block size is 3.

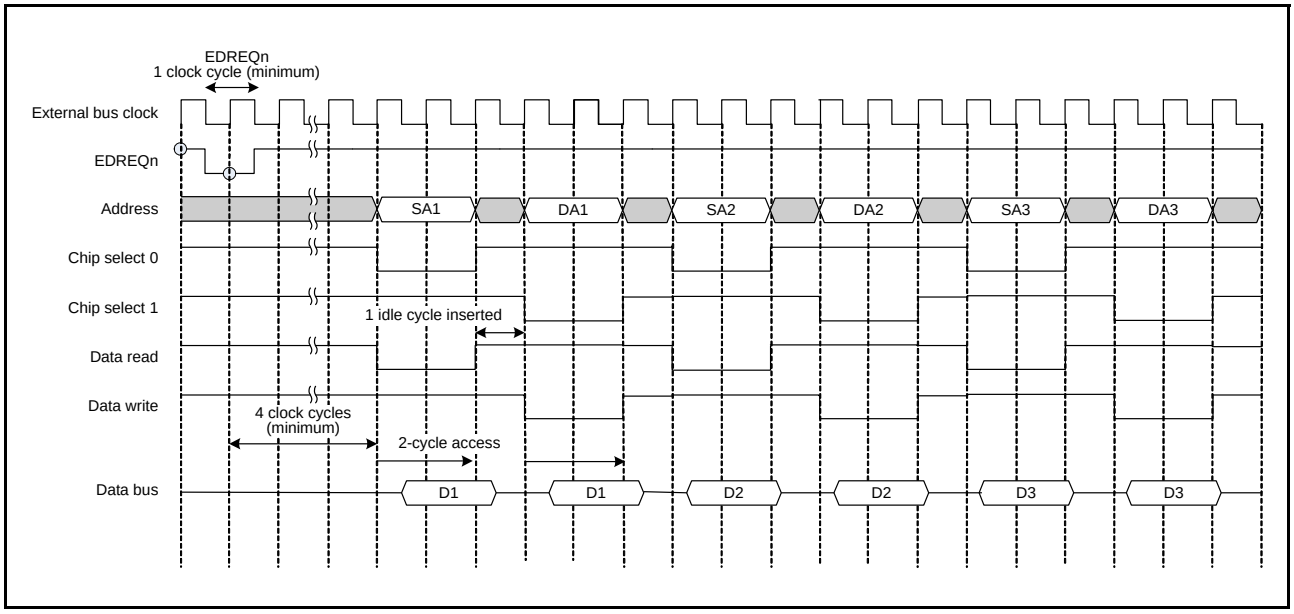


Figure 19.14 Bus Cycle Example in Block-Transfer Dual Address Mode

(2) Single Address Mode

Figure 19.15 shows the bus cycle example in block-transfer single address mode. In the example, a 16-bit data (SZ[1:0] = 01 in EDMTMD of EXDMACn) is transferred in three bus clock cycles from a device with EDACKn with 16-bit access to another device with 16-bit access when the block size is 3.

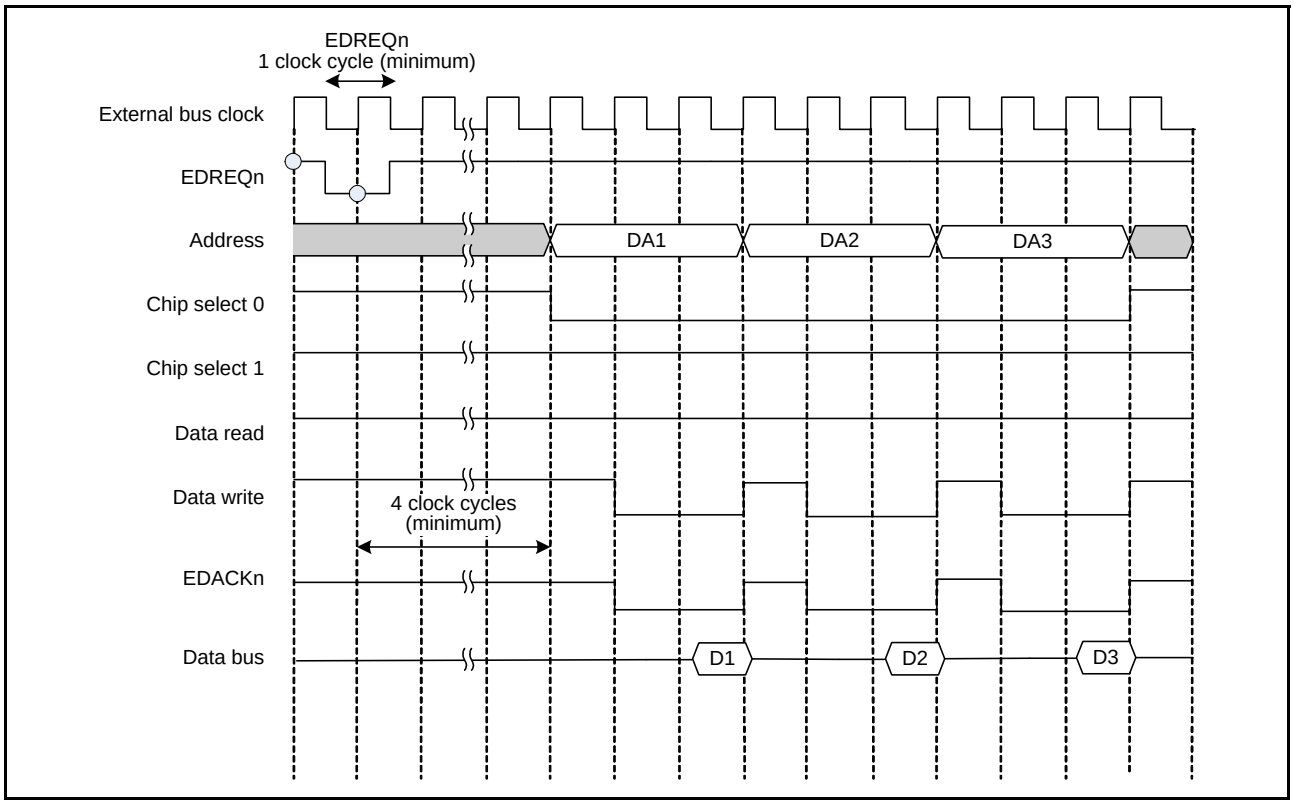


Figure 19.15 Data Flow in Block Transfer Single Address Mode

19.4.3 Cluster Transfer Operation

(1) Dual Address Mode

In cluster-transfer dual address mode, cluster-size data is transferred from the external transfer source device to the external transfer destination device via the cluster buffers. Figure 19.16 shows the data flow in cluster-transfer dual address mode and Figure 19.17 shows the bus cycle example, in which one cluster is transferred in two clock cycles when the cluster size is three.

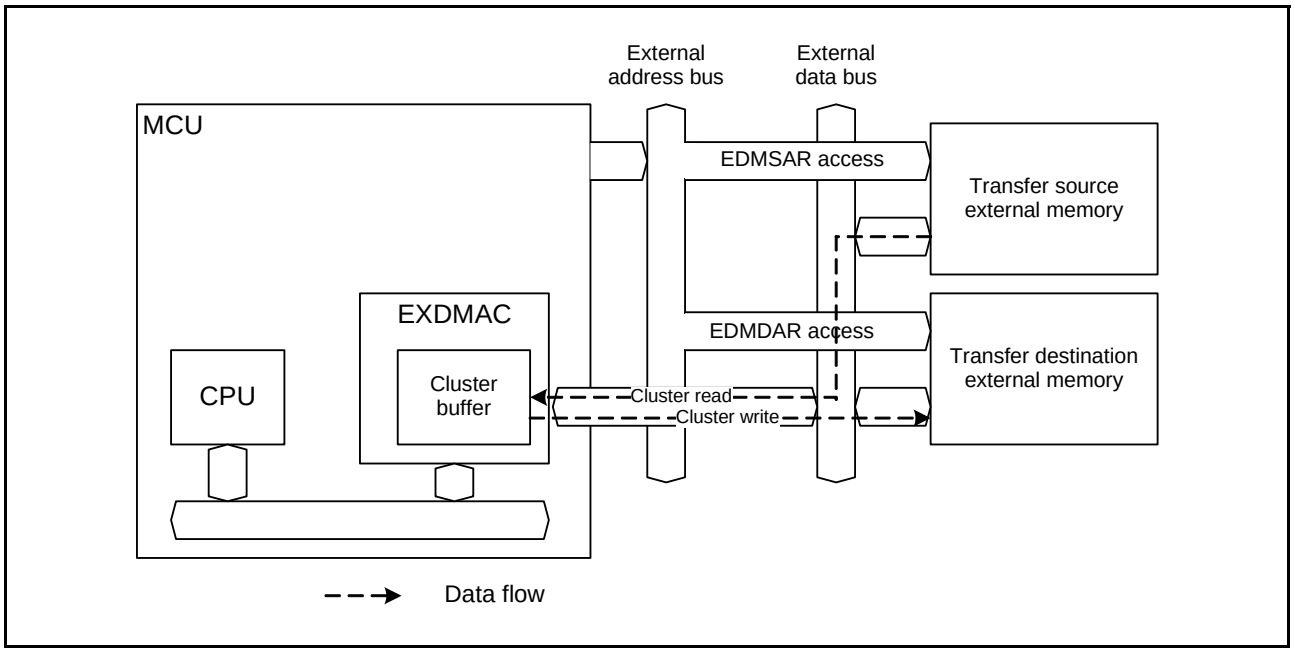


Figure 19.16 Data Flow in Cluster-Transfer Dual Address Mode

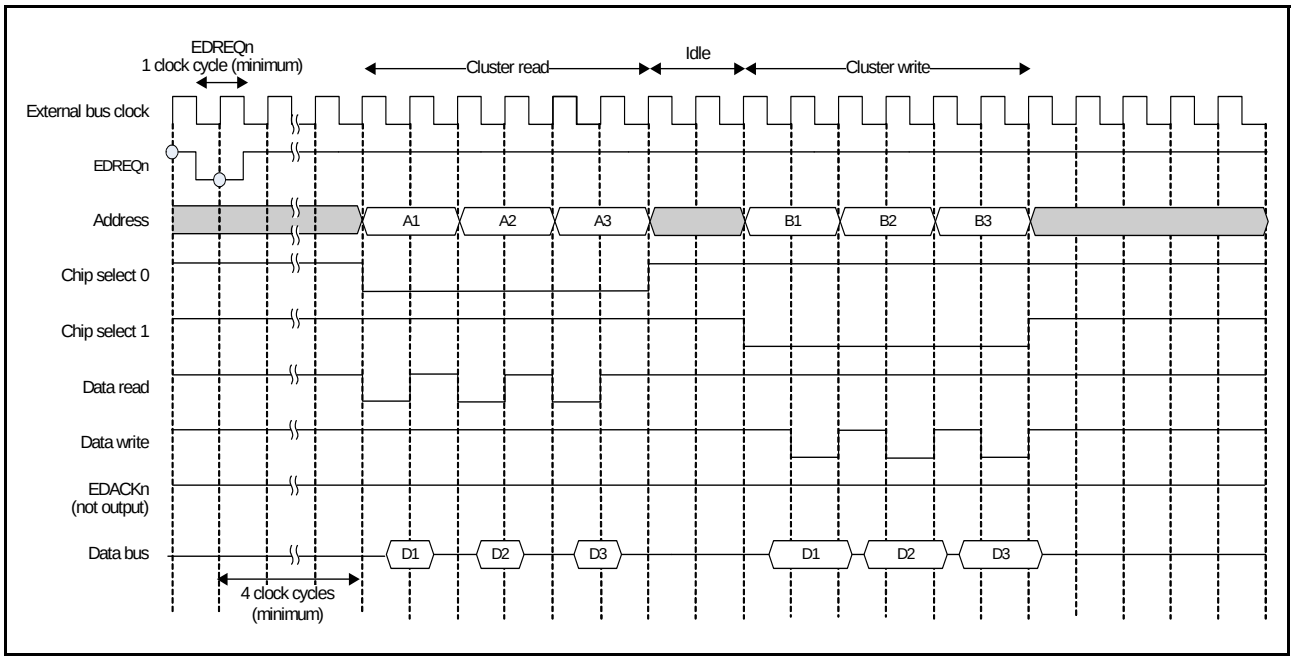


Figure 19.17 Bus Cycle Example in Cluster-Transfer Dual Address Mode

(2) Read Address Mode

In cluster-transfer read address mode, cluster-size data is transferred from the external transfer source device to the cluster buffers. The data transferred in the cluster buffers can be read by the CPU. Figure 19.18 shows the data flow in cluster-transfer read address mode and Figure 19.19 shows the bus cycle example, in which one cluster is transferred in two clock cycles when the cluster size is six.

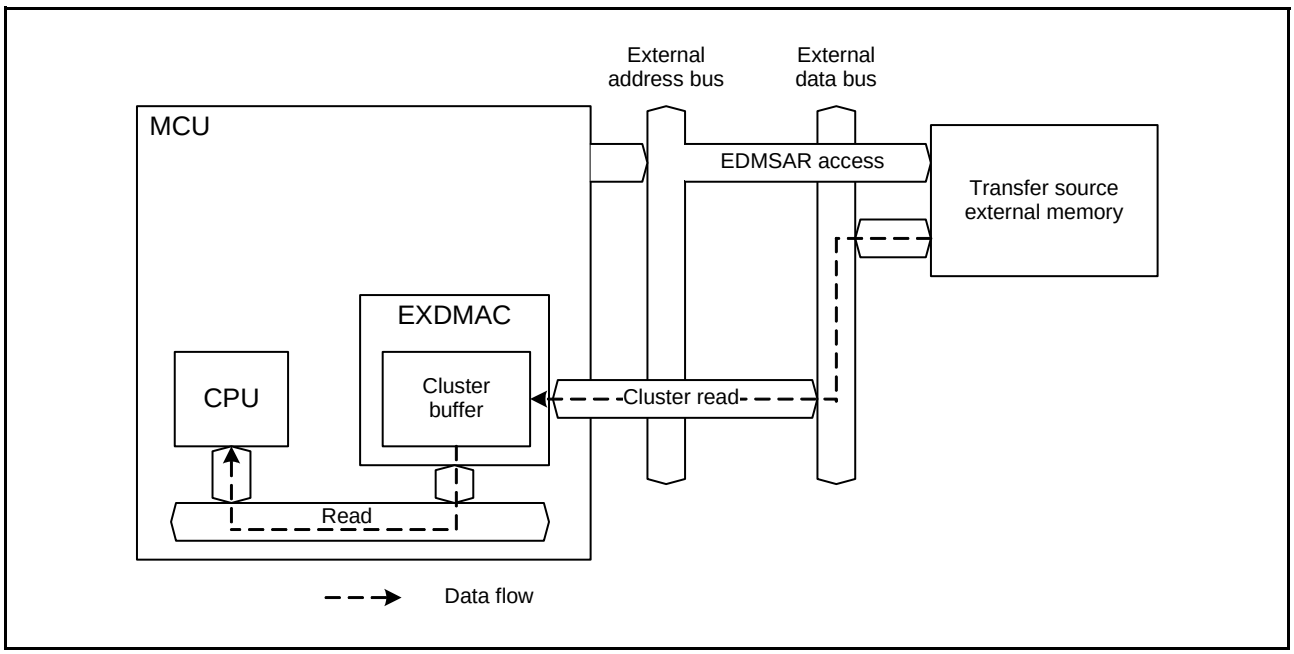


Figure 19.18 Data Flow in Cluster-Transfer Read Address Mode

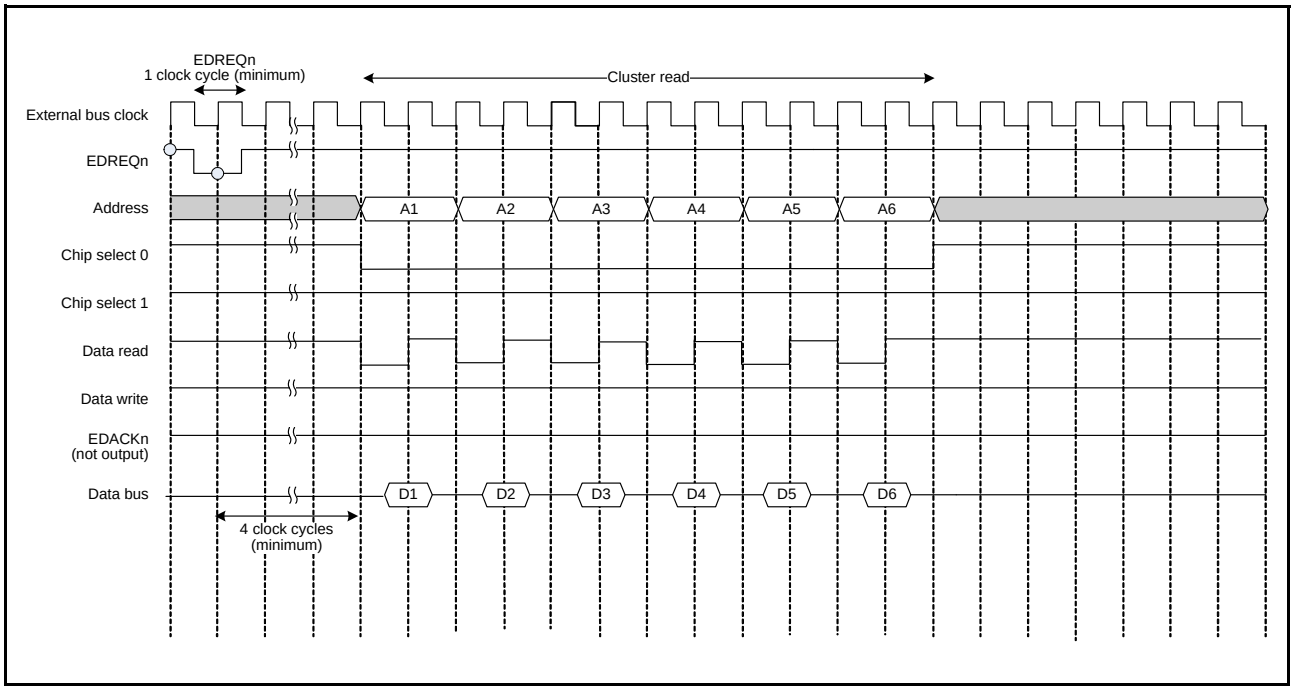


Figure 19.19 Bus Cycle Example in Cluster-Transfer Read Address Mode

(3) Write Address Mode

In cluster-transfer write address mode, the data is written to the cluster buffers by the internal bus master such as the CPU, DMAC, and DTC and then transferred to the external transfer destination device. Figure 19.20 shows the data flow in cluster-transfer write address mode and Figure 19.21 shows the bus cycle example, in which one cluster is transferred in two clock cycles when the cluster size is six.

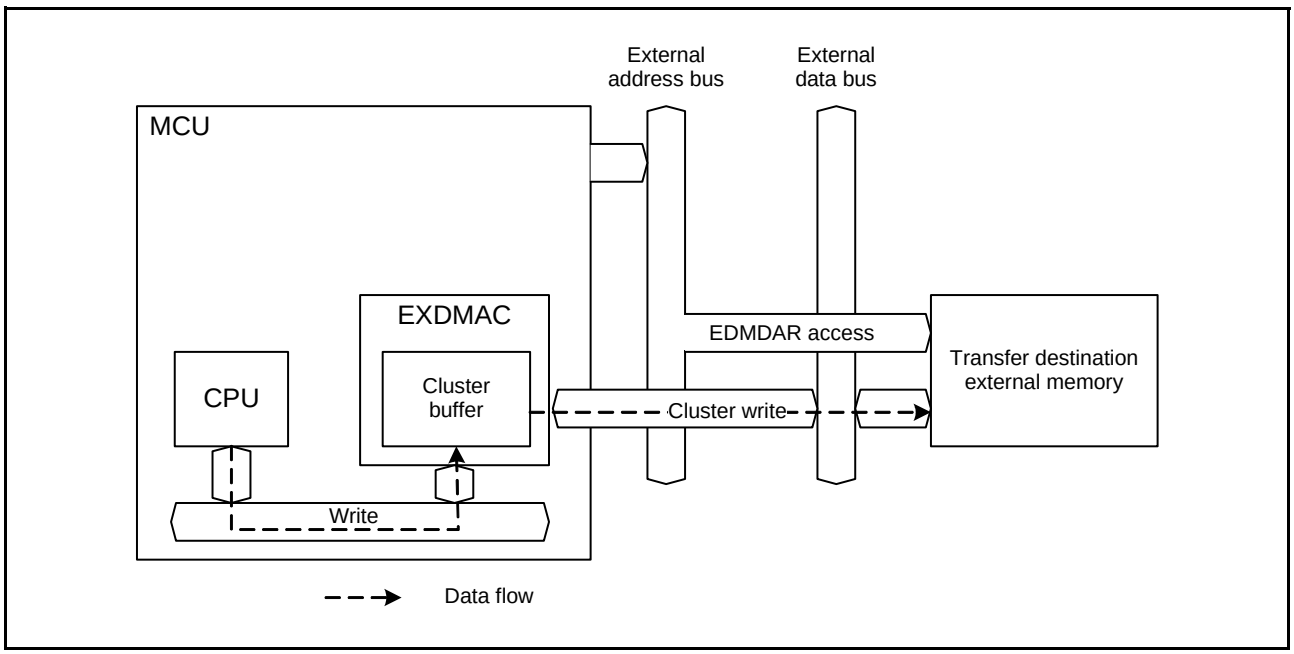


Figure 19.20 Data Flow in Cluster-Transfer Write Address Mode

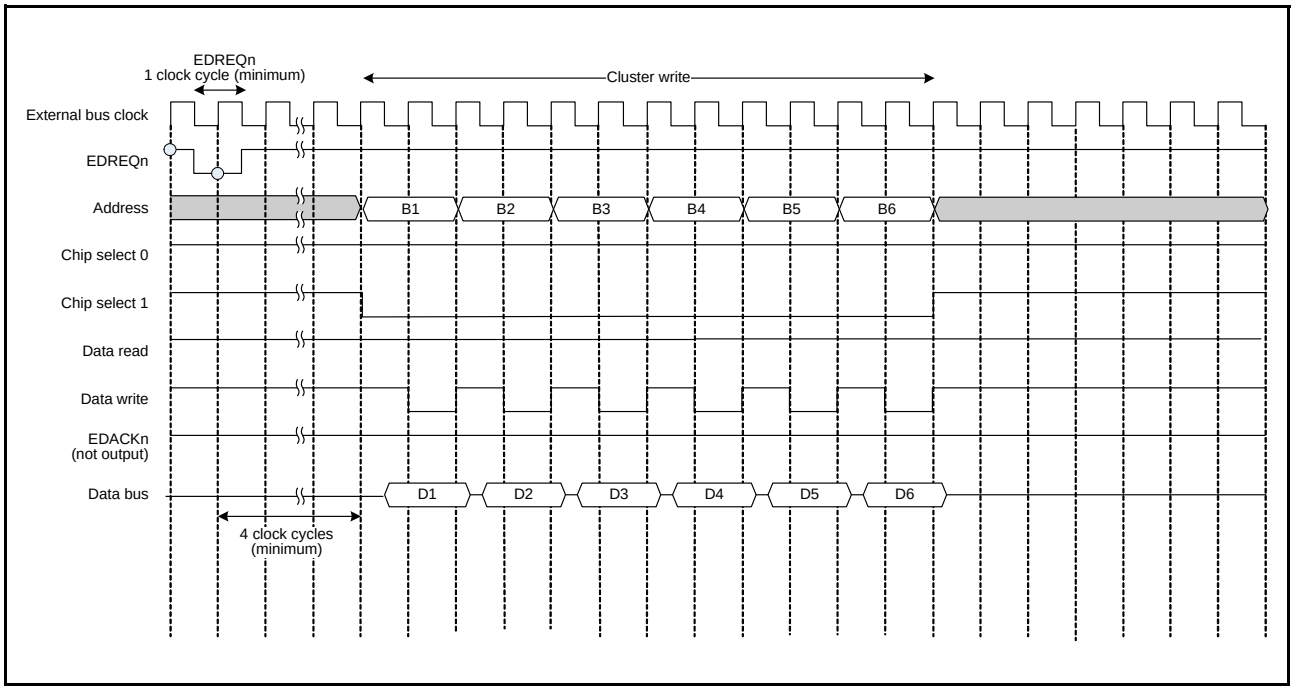


Figure 19.21 Bus Cycle Example in Cluster-Transfer Write Address Mode

19.5 Request Sources and Procedures

19.5.1 Request Sources

The EXDMAC can be triggered by software, by an external DMA transfer request pin (EDREQn pins), or by the DMA transfer requests from the peripheral modules (the software configurable interrupt B source select register 144 (SLIBR144) or the software configurable interrupt A source select register 208 (SLIAR208) should be set for channel 0, and the software configurable interrupt B source select register 145 (SLIBR145) or the software configurable interrupt A source select register 209 (SLIAR209) should be set for channel 1). Setting the DCTG[1:0] bits in EDMTMD of EXDMACn selects the request source.

(1) Trigger by Software

Setting the EXDMACn.EDMREQ.DCTG[1:0] bits to 00b enables the trigger by software.
To start DMA transfer by software, follow the procedure below.

1. Check that the EXDMACn.EDMREQ.SWREQ bit is 0 (DMA transfer is not requested).
2. Set the EXDMACn.EDMTMD.DCTG[1:0] bits to 00b (activation by software).
3. Set the EXDMACn.EDMCNT.DTE bit to 1 (enables DMA transfer).
4. Set the EXDMACn.EDMREQ.CLRS bit (DMA software start bit auto clear select) and also set the EXDMACn.EDMREQ.SWREQ bit to 1 (DMA transfer is requested).

When the EXDMACn.DMREQ.CLRS bit is 0 (SWREQ bit is cleared after DMA transfer is started by software), the EXDMACn.EDMREQ.SWREQ bit is set to 0 after data transfer is started in response to a DMA transfer request. When the CLRS bit is 1 (SWREQ bit is not cleared after DMA transfer is started by software), the SWREQ bit is not set to 0. In this case, a DMA transfer request is issued again after completion of a transfer.

(2) Trigger by external DMA transfer request Pin (EDREQn)

Setting the EXDMACn.EDMTMD.DCTG[1:0] bits to 10b enables the trigger by the external DMA transfer request pins.
To set the trigger by the external DMA transfer request pin, follow the procedure below.

1. Set the detection mode by the EXDMACn.EDMRMD.DREQS[1:0] bits.
2. Set the EXDMACn.EDMTMD.DCTG[1:0] bits to 10b (external DMA transfer request pins).
3. Set the EXDMACn.EDMERF.EREQ flag to 1 to clear the EREQ flag.
4. Set the EXDMACn.EDMCNT.DTE bit to 1 (enables DMA transfer).

When the falling edge or rising edge is selected by using the EXDMACn.EDMRMD.DREQS[1:0] bits, if the external DMA transfer request pin detects an edge, the EXDMACn.EDMERF.EREQ flag is set to 1. The EXDMACn.EDMERF.EREQ flag is set to 0 when the DMA transfer is started by the external request. Moreover, this flag is set to 0 by writing 1 to it.
When the low level detection is set by the EXDMACn.EDMRMD.DREQS[1:0] bits, if the external DMA transfer request pin is low level, the EXDMACn.EDMERF.EREQ flag is set to 1 (DMA transfer is requested). If the external DMA transfer request pin is high level, the EXDMACn.EDMERF.EREQ flag is 0 (DMA transfer is not requested). In case of the low level detection, when the DMA transfer is started by the external request or 1 is written to the flag, the EXDMACn.EDMERF.EREQ flag is not set to 0.
When the EDMAST.DMST bit and the EXDMACn.EDMCNT.DTE bit are set to 1 (enables DMA transfer) while the EXDMACn.EDMERF.EREQ flag is 1 (DMA transfer is requested), the DMA transfer is started.
The value of the EXDMACn.EDMERF.EREQ flag is retained regardless of the settings in the EDMAST.DMST and EXDMACn.EDMCNT.DTE bits.
Figure 19.22 and Figure 19.23 show the external DMA transfer request timings in the falling edge detection mode and low level detection mode, respectively.

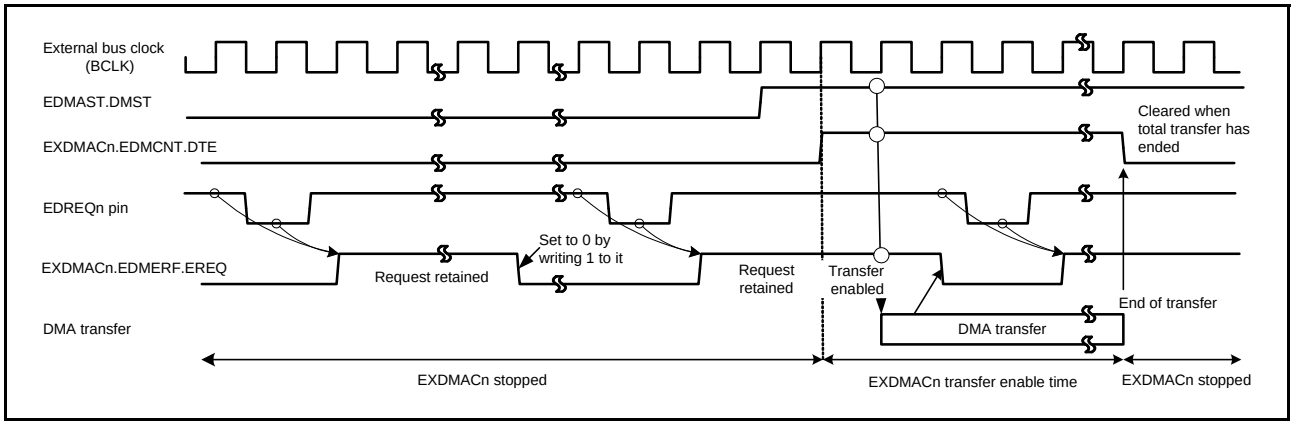


Figure 19.22 External DMA Transfer Request Timing in Falling-Edge Detection Mode

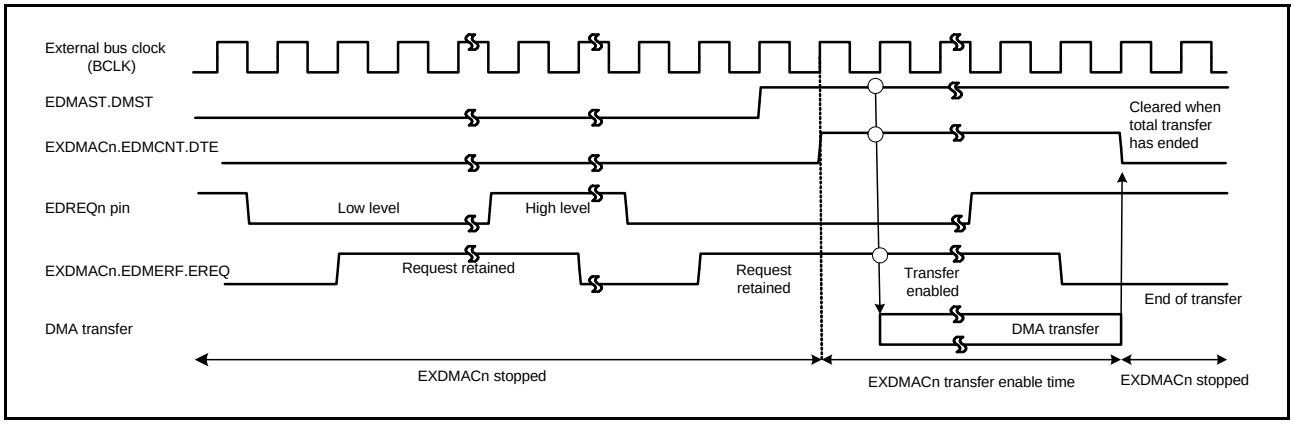


Figure 19.23 External DMA Transfer Request Timing in Low-Level Detection Mode

(3) Trigger by DMA Transfer Requests from Peripheral Modules

Setting the DCTG[1:0] bits in EDMTMD of EXDMACn to 11b enables the trigger by DMA transfer requests from the peripheral modules (an interrupt specified by the software configurable interrupt B source select register (ICU.SLIBR144 or ICU.SLIBR145) or the software configurable interrupt A source select register (ICU.SLIAR208 or ICU.SLIAR209)).

To start DMA transfer by DMA transfer requests from the peripheral modules follow the procedure below.

1. Set the software configurable interrupt B source select register 144 (SLIBR144) or the software configurable interrupt A source select register 208 (SLIAR208) for channel 0, and set the software configurable interrupt B source select register 145 (SLIBR145) or the software configurable interrupt A source select register 209 (SLIAR209) for channel 1 (For the setting procedure, refer to section 15.4.5, Software Configurable Interrupts, in section 15, Interrupt Controller (ICUE)).
2. Set the EXDMACn.EDMTMD.DCTG[1:0]bits to 11b (DMA transfer requests from the peripheral modules).
3. Set the EXDMACn.EDMPRF.PREQ flag to 1, and then clear it to 0.
4. Set the EXDMACn.EDMCNT.DTE bit to 1 (DMA transfer is enabled).

None of the values of the interrupt request enable bits (IERm.IENj) affect EXDMACn requests initiated by peripheral modules.

When a DMA transfer request is input from the peripheral modules, the EXDMACn.EDMPRF.PREQ flag is set to 1 (DMA transfer is requested). The EXDMACn.EDMPRF.PREQ flag is set to 0 (DMA transfer is not requested) when the DMA transfer is started by the peripheral module request.

This flag is set to 0 by writing 1 to it.
When the EDMAST.DMST bit is set to 1 (EXDMAC module start) and the EXDMACn.EDMCNT.DTE bit is set to 1 (DMA transfer is enabled) while the EXDMACn.EDMPRF.PREQ flag is 1, the DMA transfer is started.
The value of the EXDMACn.EDMPRF.PREQ flag is retained regardless of the settings of the EDMAST.DMST and EXDMACn.EDMCNT.DTE bits.
The EXDMACn.EDMPRF.PREQ flag is set to 1 regardless of the state of the module stop bit when the corresponding EXDMACn is started by an internal peripheral module.

19.5.2 Activating the EXDMAC

Figure 19.24 shows the register setting procedure.

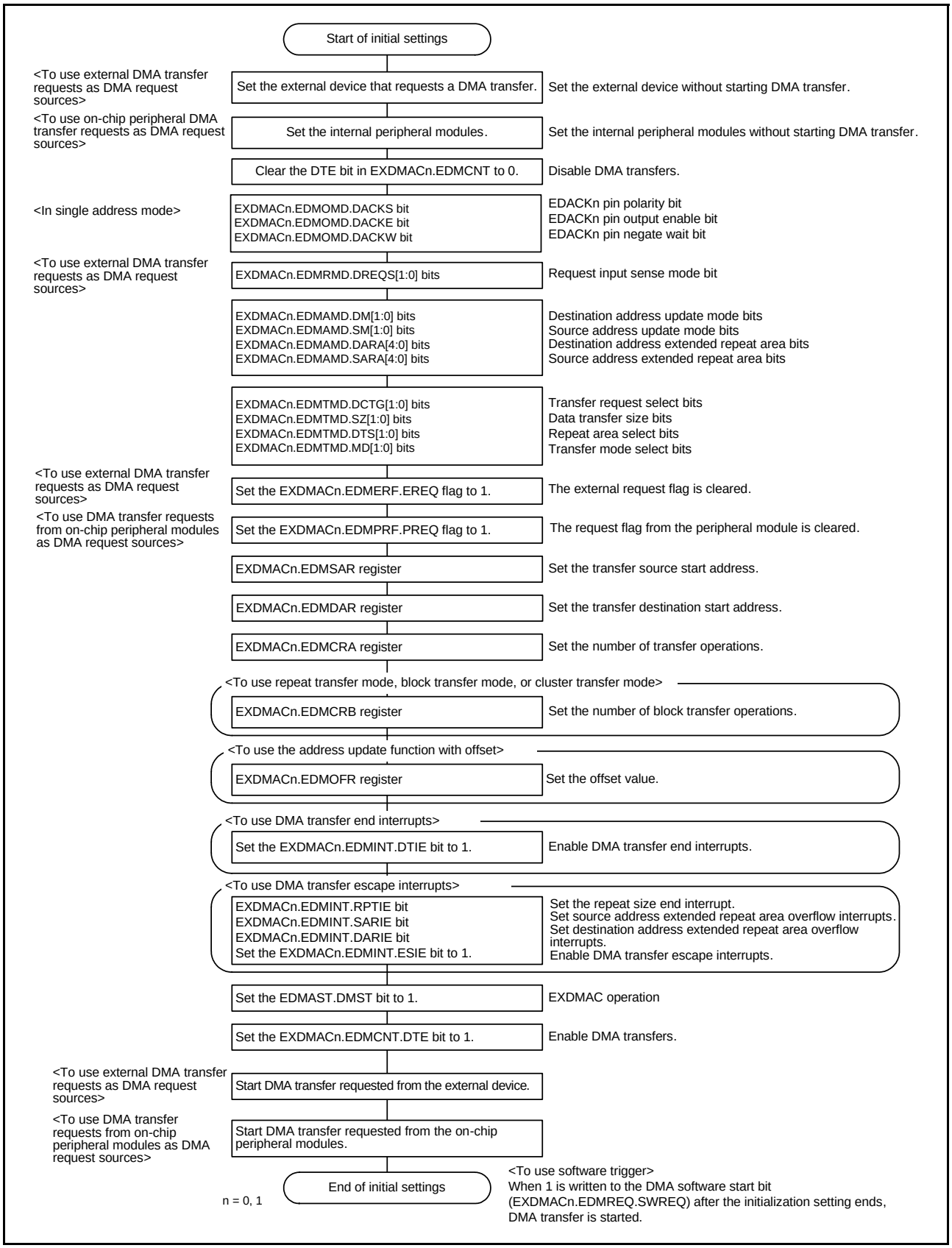


Figure 19.24 Register Setting Procedure

19.5.3 Starting DMA Transfer

Setting the DTE bit in EDMCNT of EXDMACn to 1 (DMA transfer enabled) and setting the DMST bit in EDMAST to 1 (EXDMAC start) enable DMA transfer of channel n (n = 0, 1).
When DMA transfer requests are generated, channel arbitration is performed where a DMA transfer request of higher-priority channel is accepted and DMA transfer of the channel starts. When a DMA transfer request is accepted and DMA transfer starts, the ACT flag in EDMSTS of EXDMACn is set to 1 (DMA transfer is in progress).

19.5.4 Registers during DMA Transfer

The EXDMAC registers are updated by a DMA transfer. The value to be updated differs according to the other settings and the transfer state. The registers to be updated are EDMSAR, EDMDAR, EDMCRA, EDMCRB, EDMCNT, and EDMSTS of EXDMACn.

(1) EXDMA Source Address Register (EXDMACn.EDMSAR)

When data has been transferred in response to one transfer request, the contents of EDMSAR are updated to the address to be accessed by the next transfer request.
For details on register update operation in each transfer mode, refer to Table 19.4 to Table 19.7.

(2) EXDMA Destination Address Register (EXDMACn.EDMDAR)

When data has been transferred in response to one transfer request, the contents of EDMDAR are updated to the address to be accessed by the next transfer request.
For details on register update operation in each transfer mode, refer to Table 19.4 to Table 19.7.

(3) EXDMA Transfer Count Register (EXDMACn.EDMCRA)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.
For details on register update operation in each transfer mode, refer to Table 19.4 to Table 19.7.

(4) EXDMA Block Transfer Count Register (EXDMACn.EDMCRB)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.
For details on register update operation in each transfer mode, refer to Table 19.4 to Table 19.7.

(5) DMA Transfer Enable Bit (EXDMACn.EDMCNT.DTE)

The EXDMACn.EDMCNT.DTE bit provides a way to control enabling or prohibition of transfer by writing to the relevant registers.
The EXDMACn.EDMCNT.DTE bit is set to 0 when any of the following conditions is generated by the DMA transfer.

- When the specified total volume of data transfer is completed
- When DMA transfer is stopped by the repeat size end interrupt
- When DMA transfer is stopped by the extended repeat area overflow interrupt

Writing to the registers of an EXDMAC channel is prohibited if the corresponding EXDMACn.EDMCNT.DTE bit is 1 (except to the EXDMACn.EDMCNT register itself).
Change the settings of the registers as required after writing 0 to the EXDMACn.EDMCNT.DTE bit.

(6) EXDMA Active Flag (EXDMACn.EDMSTS.ACT)

The ACT flag in EDMSTS of EXDMACn indicates whether the EXDMACn is in the idle or active state.
This flag is set to 1 when the EXDMACn starts data transfer, and is set to 0 when data transfer in response to one transfer request is completed.
Even when DMA transfer is stopped by writing 0 to the DTE bit in EDMCNT of EXDMACn, this flag remains 1 until DMA transfer is completed.

(7) Transfer End Interrupt Flag (EXDMACn.EDMSTS.DTIF)

The DTIF flag in EDMSTS of EXDMACn is set to 1 after transfer of the total transfer size of data is completed.
When both this flag and the DTIE bit in EDMINT of EXDMACn are set to 1, a transfer end interrupt is requested.
This flag is set to 1 when the DMA transfer bus cycle is completed and the ACT flag in EDMSTS of EXDMACn is set to 0 indicating the DMA transfer end.
This flag is automatically set to 0 when the DTE bit in EDMCNT of EXDMACn is set to 1 during the interrupt handling.

(8) Transfer Escape End Interrupt Flag (EXDMACn.EDMSTS.ESIF)

The ESIF flag in EDMSTS of EXDMACn is set to 1 when a repeat size end interrupt or extended repeat area overflow interrupt is requested. When this flag and the ESIE bit in EDMINT of EXDMACn are set to 1, a transfer escape end interrupt is requested.
This flag is set to 1 when the bus cycle of the DMA transfer having caused the interrupt request is completed and the ACT flag in EDMSTS of EXDMACn is set to 0 indicating the DMA transfer end.
This flag is automatically set to 0 when the DTE bit in EDMCNT of EXDMACn is set to 1 during an interrupt handling.
Before sending an interrupt request from the EXDMACn to the CPU or the DTC, the interrupt control register must be set.
For details, refer to section 15, Interrupt Controller (ICUE).

19.5.5 Channel Priority

When multiple DMA transfer requests are present, the EXDMAC determines the priority of channels that have DMA transfer requests. The channel priority is fixed as channel 0 > channel 1.
When a DMA transfer request is generated during data transfer, channel arbitration is started after the final data has been transferred, and DMA transfer of the higher-priority channel starts.

19.6 Ending DMA Transfer

The operation for ending DMA transfer depends on the transfer end conditions. When DMA transfer ends, the DTE bit in EDMCNT and the ACT flag in EDMSTS of EXDMACn are changed from 1 to 0, indicating that DMA transfer has ended.

19.6.1 Transfer End by Completion of Specified Total Number of Transfer Operations

(1) In Normal Transfer Mode (EXDMACn.EDMTMD.MD[1:0] = 00b)

When the value of the EXDMACn.EDMCRAL register changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in EDMCNT of EXDMACn is set to 0 and the DTIF flag in EDMSTS of EXDMACn is set to 1 at the same time. If the DTIE bit in EDMINT of EXDMACn is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

(2) In Repeat Transfer Mode (EXDMACn.EDMTMD.MD[1:0] = 01b)

When the value of DMCRB of EXDMACn changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in EDMCNT of EXDMACn is set to 0 and the DTIF flag in EDMSTS of EXDMACn is set to 1 at the same time. If the DTIE bit in EDMINT of EXDMACn is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

(3) In Block Transfer Mode (EXDMACn.EDMTMD.MD[1:0] = 10b)

When the value of EDMCRB of EXDMACn changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in EDMCNT of EXDMACn is set to 0 and the DTIF flag in EDMSTS of EXDMACn is set to 1 at the same time. If the DTIE bit in EDMINT of EXDMACn is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

(4) In Cluster Transfer Mode (EXDMACn.EDMTMD.MD[1:0] = 11b)

When the value of EDMCRB of EXDMACn changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in EDMCNT of EXDMACn is set to 0 and the DTIF flag in EDMSTS of EXDMACn is set to 1 at the same time. If the DTIE bit in EDMINT of EXDMACn is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

Before sending an interrupt request from the EXDMACn to the CPU or the DTC, the interrupt control register must be set. For details, refer to section 15, Interrupt Controller (ICUE).

19.6.2 Transfer End by Repeat Size End Interrupt

In repeat transfer mode, a repeat size end interrupt is requested when transfer of a 1-repeat size of data is completed while the RPTIE bit in EDMINT of EXDMACn is set to 1. When the interrupt is requested to complete DMA transfer, the DTE bit in EDMCNT of EXDMACn is set to 0 and the ESIF flag in EDMSTS of EXDMACn is set to 1. If the ESIE bit in EDMINT of EXDMACn is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC. Here, the transfer can be resumed by writing 1 to the DTE bit in EDMCNT of EXDMACn.
A repeat size end interrupt can be requested also in block transfer mode (or cluster transfer mode). In block transfer mode (or cluster transfer mode), the interrupt is requested in the same way as in repeat transfer mode when transfer of a 1-block size (or 1-cluster) data is completed.
Before sending an interrupt request from the EXDMACn to the CPU or the DTC, the interrupt control register must be set. For details, refer to section 15, Interrupt Controller (ICUE).

19.6.3 Transfer End by Interrupt on Extended Repeat Area Overflow

When an overflow on the extended repeat area occurs while the extended repeat area is specified and the SARIE or DARIE bit in EDMINT of EXDMACn is set to 1, an interrupt by an extended repeat area overflow is requested. When the interrupt is requested, the DMA transfer is terminated, the DTE bit in EDMCNT of EXDMACn is set to 0, and the ESIF flag in EDMSTS of EXDMACn is set to 1. If the ESIE bit in EDMINT of EXDMACn is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.
Even if an interrupt by an extended repeat area overflow is requested during a read cycle, the following write cycle is performed.
In block transfer mode (or cluster transfer mode), even if an interrupt by an extended repeat area overflow is requested during a 1-block (or 1-cluster) transfer, the remaining data in the block (or the cluster) is transferred; transfer is terminated after a block (or cluster) transfer.
Before sending an interrupt request from the EXDMACn to the CPU or the DTC, the interrupt control register must be set. For details, refer to section 15, Interrupt Controller (ICUE).

19.7 Interrupts

The EXDMAC can output one interrupt request to the CPU or the DTC for each channel. Table 19.10 lists the relation among the interrupt sources, the interrupt status flags, and the interrupt enable bits. Figure 19.25 shows the schematic logic diagram of interrupt outputs. The procedures for suspending or resuming DMA transfer by the EXDMAC interrupt are shown in Figure 19.26.

Table 19.10 Relation among Interrupt Sources, Interrupt Status Flags, and Interrupt Enable Bits

Interrupt Sources		Interrupt Enable Bits	Interrupt Status Flags	Request Output Enable Bits
Transfer end		—	EXDMACn.EDMSTS.DTIF	EXDMACn.EDMINT.DTIE
Escape transfer end	Repeat size end	EXDMACn.EDMINT.RPTIE	EXDMACn.EDMSTS.ESIF	EXDMACn.EDMINT.ESIE
	Source address extended repeat area overflow	EXDMACn.EDMINT.SARIE		
	Destination address extended repeat area overflow	EXDMACn.EDMINT.DARIE		

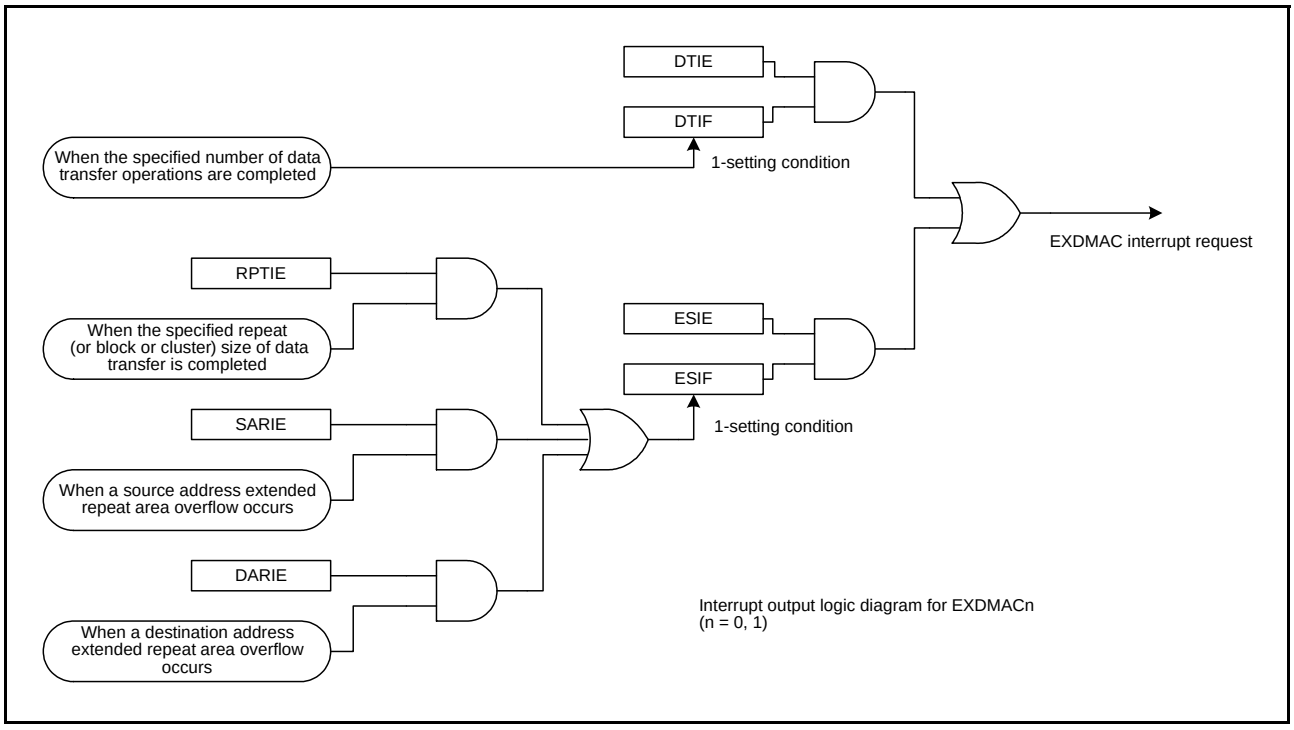


Figure 19.25 Schematic Logic Diagram of Interrupt Outputs

Specifically, the different procedures are used for canceling an interrupt to restart DMA transfer in the following two cases: (1) discontinuing or terminating DMA transfer and (2) continuing DMA transfer.

(1) When Discontinuing or Terminating DMA Transfer

Write 0 to the DTIF flag in EDMSTS of EXDMACn to clear a transfer end interrupt, and to the ESIF flag in EDMSTS of EXDMACn to clear a repeat size interrupt and an extended repeat area overflow interrupt. The EXDMACn remains in the stop state. When starting another DMA transfer after that, set the appropriate registers, and set the DTE bit in EDMCNT of EXDMACn to 1.

(2) When Continuing DMA Transfer

Write 1 to the DTE bit in EDMCNT of EXDMACn. The ESIF flag in EDMSTS of EXDMACn is automatically set to 0 (interrupt source cleared), and DMA transfer is resumed.

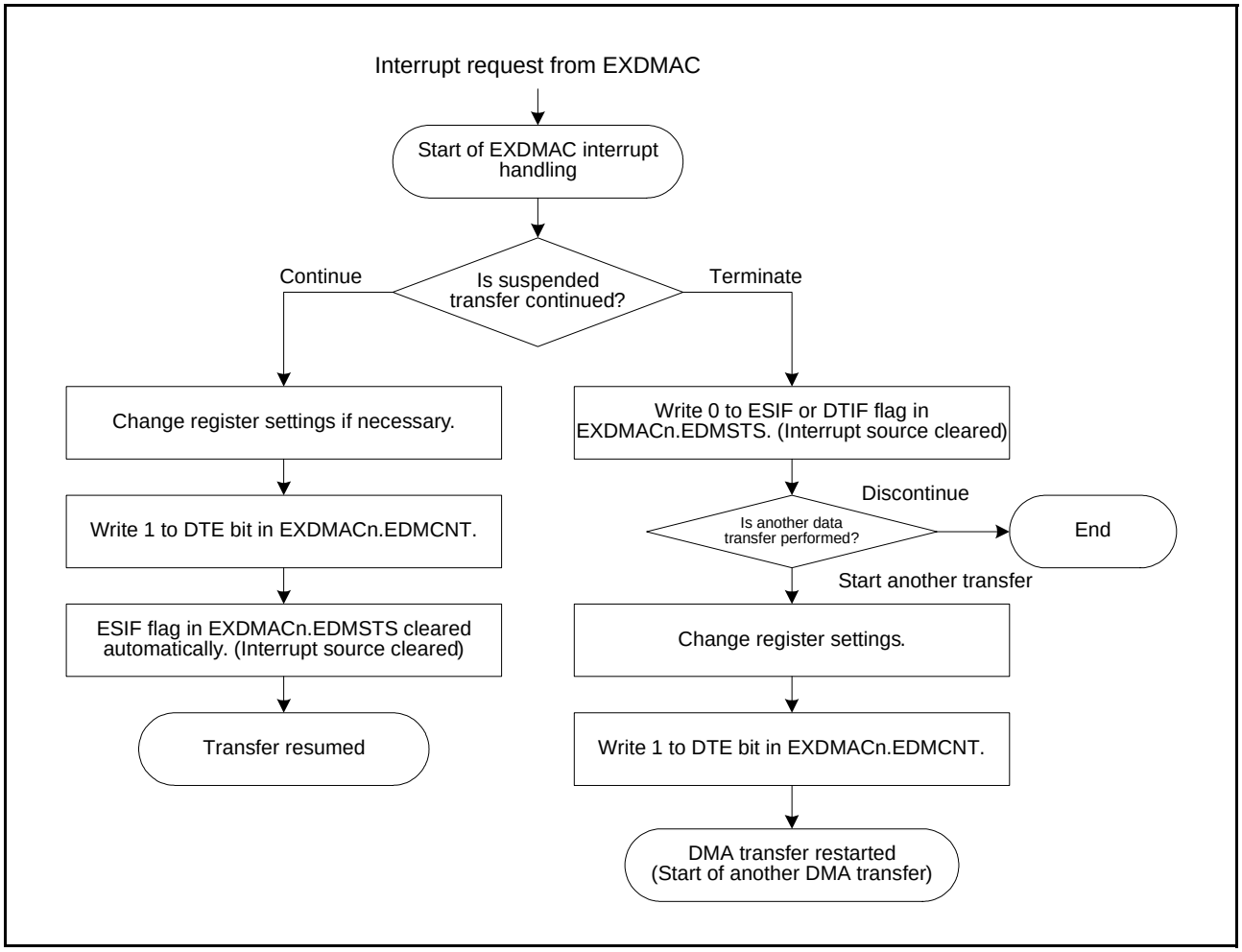


Figure 19.26 Procedures for Suspending or Resuming DMA Transfer by the EXDMAC Interrupt

19.8 Low-Power Consumption Function

To place the EXDMAC in the module-stop state, all-module clock stop mode, software-standby mode, or deep software-standby mode, set the EDMAST.DMST bit to 0 (EXDMAC stopped), and then perform the following processing.

(1) Module-Stop Function

Writing 1 to the MSTPCRA.MSTPA29 bit (transition to the module-stop state) enables the module-stop function of the EXDMAC. If DMA transfer is in progress at the time a 1 is written to the MSTPA29 bit, the transition to the module-stop state proceeds after DMAC transfer has ended.
Do not access the EXDMAC registers while the MSTPCRA.MSTPA29 bit is 1.
Writing 0 to the MSTPA29 bit releases the EXDMAC from the module-stop state.

(2) All-Module Clock Stop Mode

Follow the procedure in section 11.6.2.1, Transition to All-Module Clock Stop Mode, in section 11, Low Power Consumption. If DMA transfer is in progress at the time the WAIT instruction is executed, the EXDMAC can enter all-module clock stop mode after completion of the current DMA transfer.
After the EXDMAC returns from all-module clock stop mode, writing 0 to the MSTPA29 bit releases the EXDMAC from the module-stop state.

(3) Software Standby and Deep Software Standby Modes

Follow the procedure in section 11.6.3.1, Transition to Software Standby Mode or section 11.6.4.1, Transition to Deep Software Standby Mode, in section 11, Low Power Consumption.
If DMA transfer is in progress at the time the WAIT instruction is executed, the EXDMAC enters software standby mode or deep software stand by mode after completion of the current DMA transfer.

(4) Notes on Low-Power Consumption Function

For the timing of WAIT instruction execution and register settings, refer to section 11.7.6, Timing of WAIT Instructions, in section 11, Low Power Consumption.
To perform DMA transfer after returning from low-power consumption mode, set the EDMAST.DMST bit to 1 again.

19.9 EDACKn Operation in Single Address Mode

In single address mode, EDACKn is output to one of the external transfer-source or transfer-destination devices and the address is simultaneously output to the other transfer device for access.
When the external device receiving EDACKn transfers data to/from the CS area, the EDACKn negation timing can be adjusted by setting the DACKW bit in EDMOMD of EXDMACn. Specifically, the timing can be advanced by one BCLK cycle if the external device is a transfer destination and delayed by one BCLK cycle if the external device is a transfer source. When the external device receiving EDACKn transfers data to/from the SDRAM, the EDACKn negation timing cannot be adjusted by the DACKW bit EDMOMD of EXDMACn. If one of the parties for transfer by the EXDMAC (source or destination) is the SDRAM area, the EXDMACn.EDMOMD.DACKSEL bit can be set to 1 so that the EDACKn signal is only asserted for one-half of the SDCLK cycle in the latter half of the data-valid interval. The EXDMACn.EDMOMD.DACKSEL cannot be used to adjust the period for assertion of the EDACKn signal when a party for transfer is a CS area. For the CS area addresses and SDRAM area addresses, refer to section 4, Address Space.
The following sections show EDACKn operation examples in single address mode, in which data is transferred to/from the CS and SDRAM areas in normal and block transfer modes.

19.9.1 EDACKn Operation Example in Normal-Transfer (CS Area) Single Address Mode

Figure 19.27 shows the operation example in which data is transferred from the CS area to the device with EDACKn in normal transfer mode. Setting the DACKW bit in EDMOMD of EXDMACn to 1 allows EDACKn to be negated one BCLK cycle before the data read signal is negated.
Figure 19.28 shows the operation example in which data is transferred from the device with EDACKn to the CS area in normal transfer mode. Setting the DACKW bit to 1 allows EDACKn to be negated one BCLK cycle after the data write signal is negated.
For the data read signal, data write signal, and CS area access timing setting registers, refer to section 16, Buses.

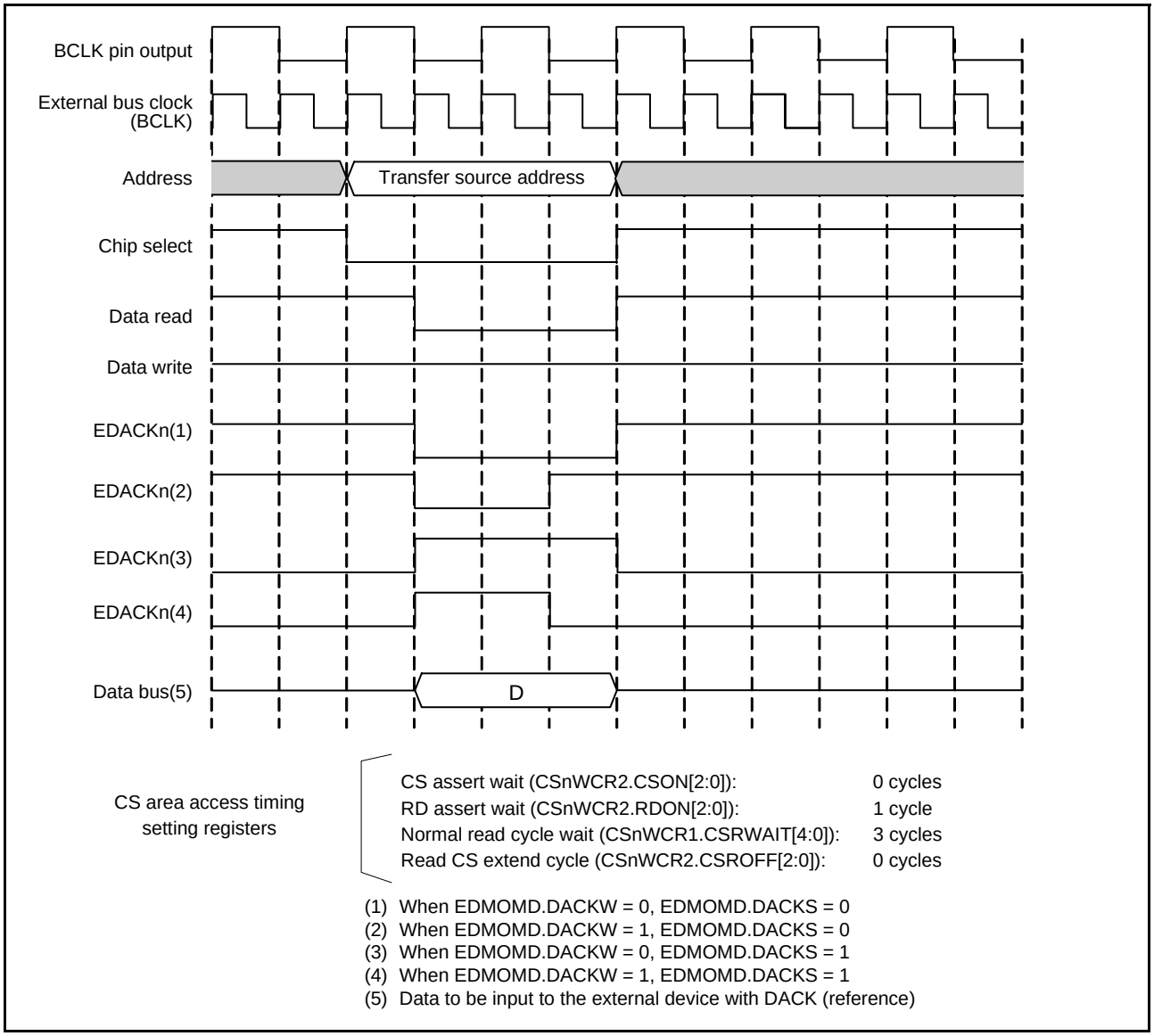


Figure 19.27 Operation Example in Normal-Transfer (CS Area Read) Single Address Mode

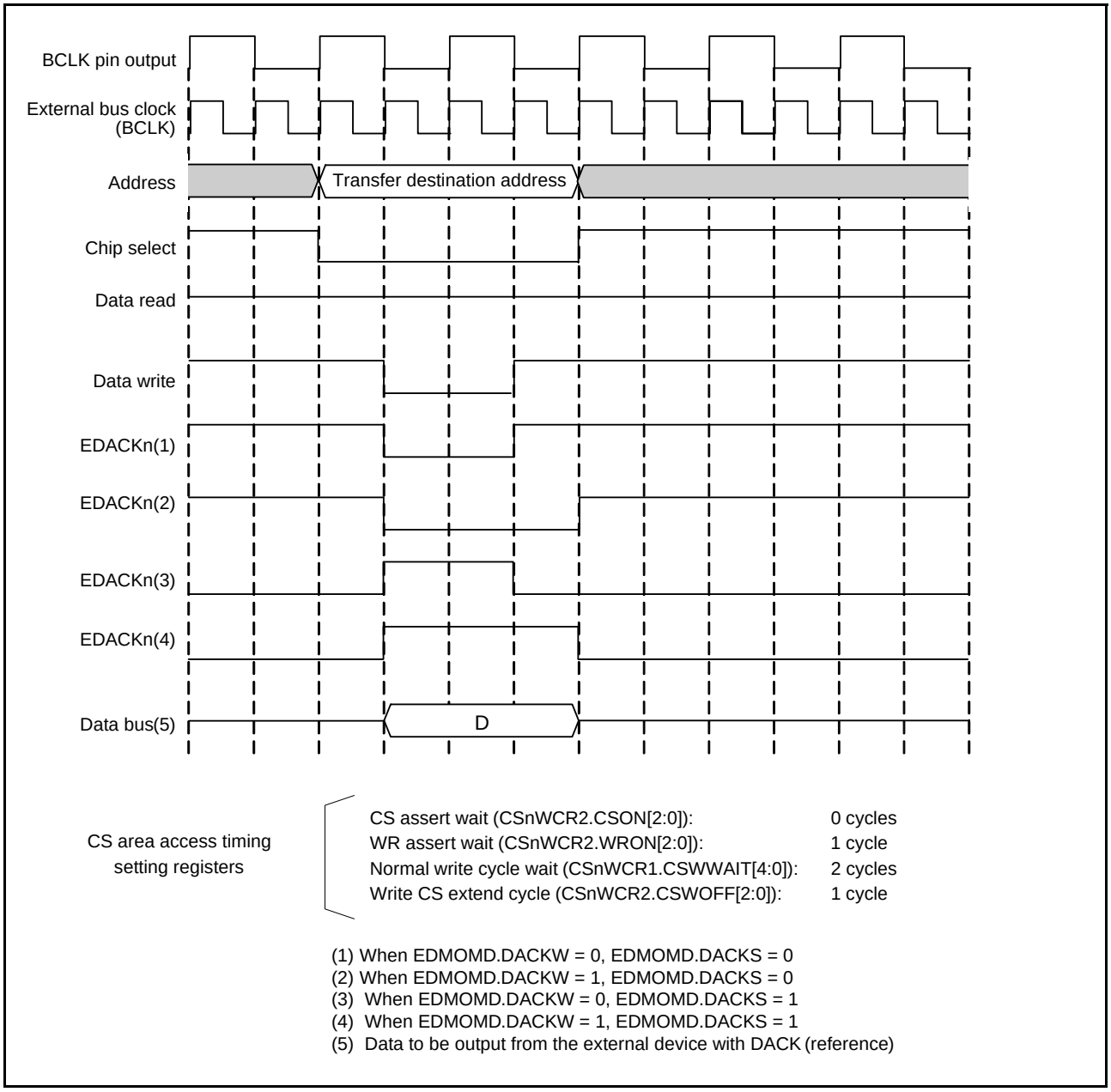


Figure 19.28 Operation Example in Normal-Transfer (CS Area Write) Single Address Mode

19.9.2 EDACKn Operation Example in Normal-Transfer (SDRAM Area) Single Address Mode

Figure 19.29 shows the operation example in which data is transferred from SDRAM to the device with EDACKn in normal transfer mode.

EDACKn is asserted while SDRAM is outputting data.

Figure 19.30 shows the operation example in which data is transferred from the device with EDACKn to SDRAM in normal transfer mode.

EDACKn is asserted while SDRAM is writing data.

For the SDRAM commands and SDRAM access timing setting registers, refer to section 16, Buses.

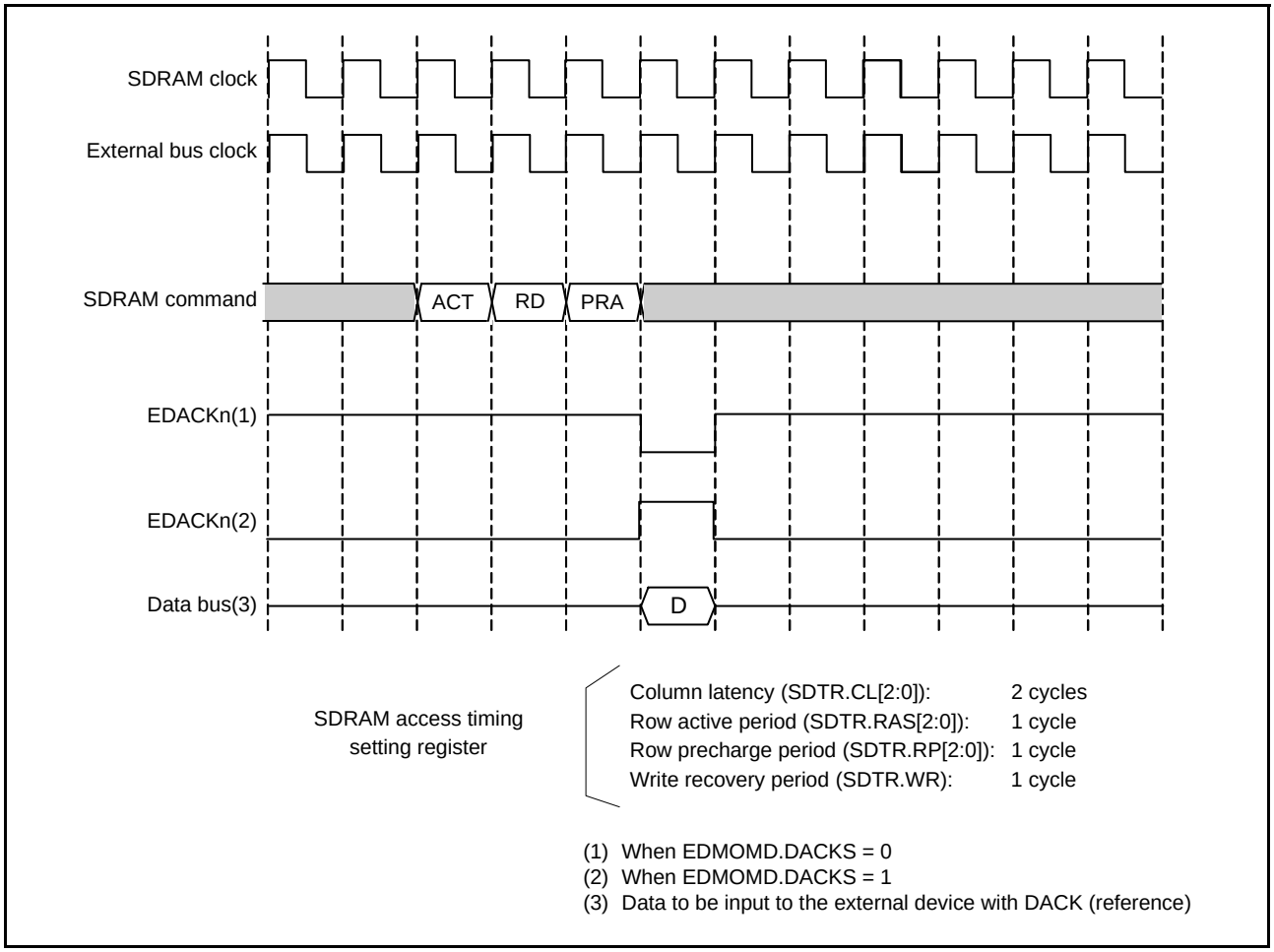


Figure 19.29 Operation Example in Normal-Transfer (SDRAM Area Read) Single Address Mode

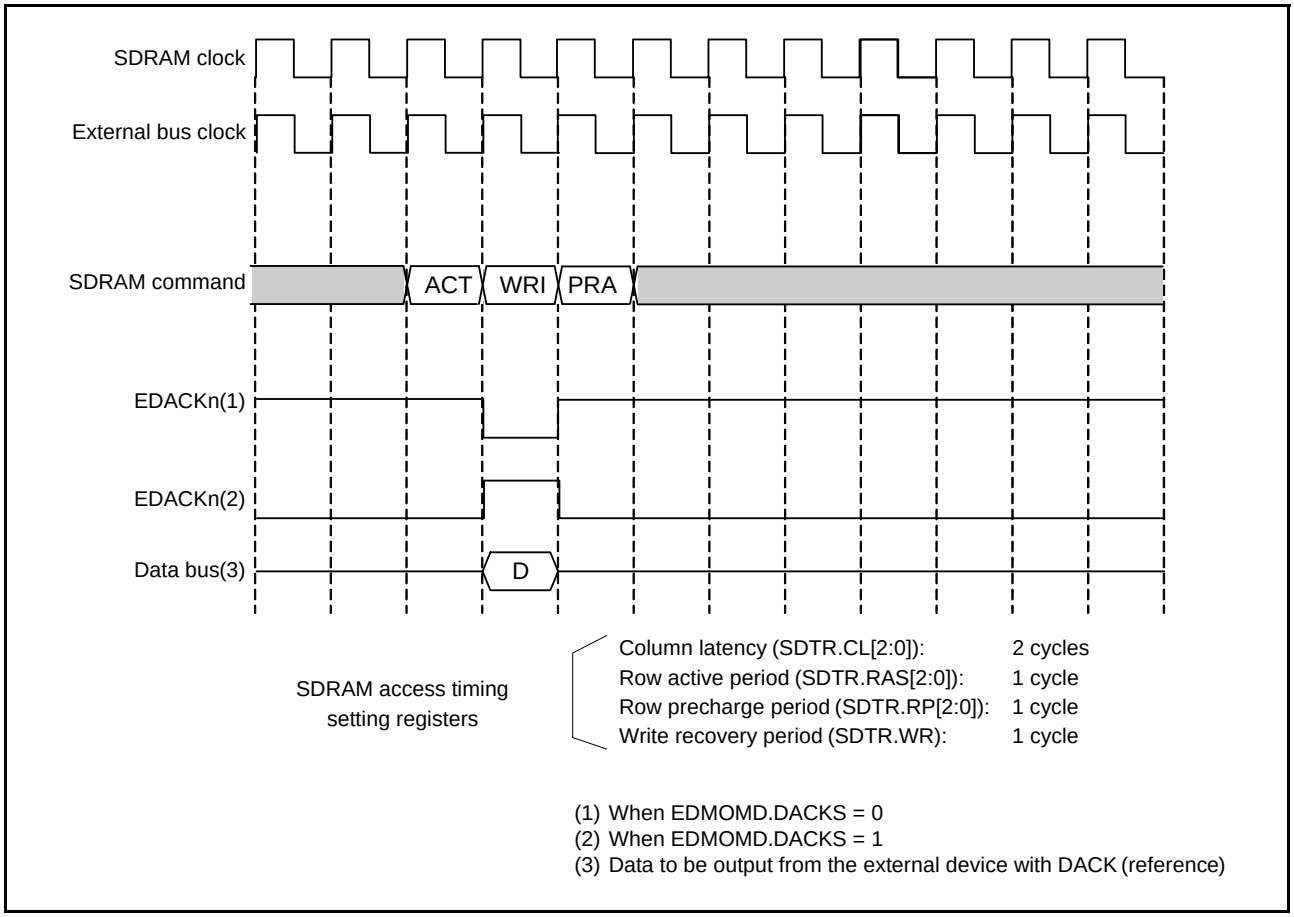


Figure 19.30 Operation Example in Normal-Transfer (SDRAM Area write) Single Address Mode

19.9.3 EDACKn Operation Example in Block-Transfer (CS Area) Single Address Mode

Figure 19.31 shows the operation example in which data is transferred from the CS area to the device with EDACKn in block transfer mode (block size = two). Setting the DACKW bit in EDMOMD of EXDMACn to 1 allows EDACKn to be negated one BCLK cycle before the data read signal is negated.

Figure 19.32 shows the operation example in which data is transferred from the device with EDACKn to the CS area in block transfer mode (block size = two). Setting the DACKW bit to 1 allows EDACKn to be negated one BCLK cycle after the data write signal is negated.

For the data read signal, data write signal, and CS area access timing setting registers, refer to section 16, Buses.

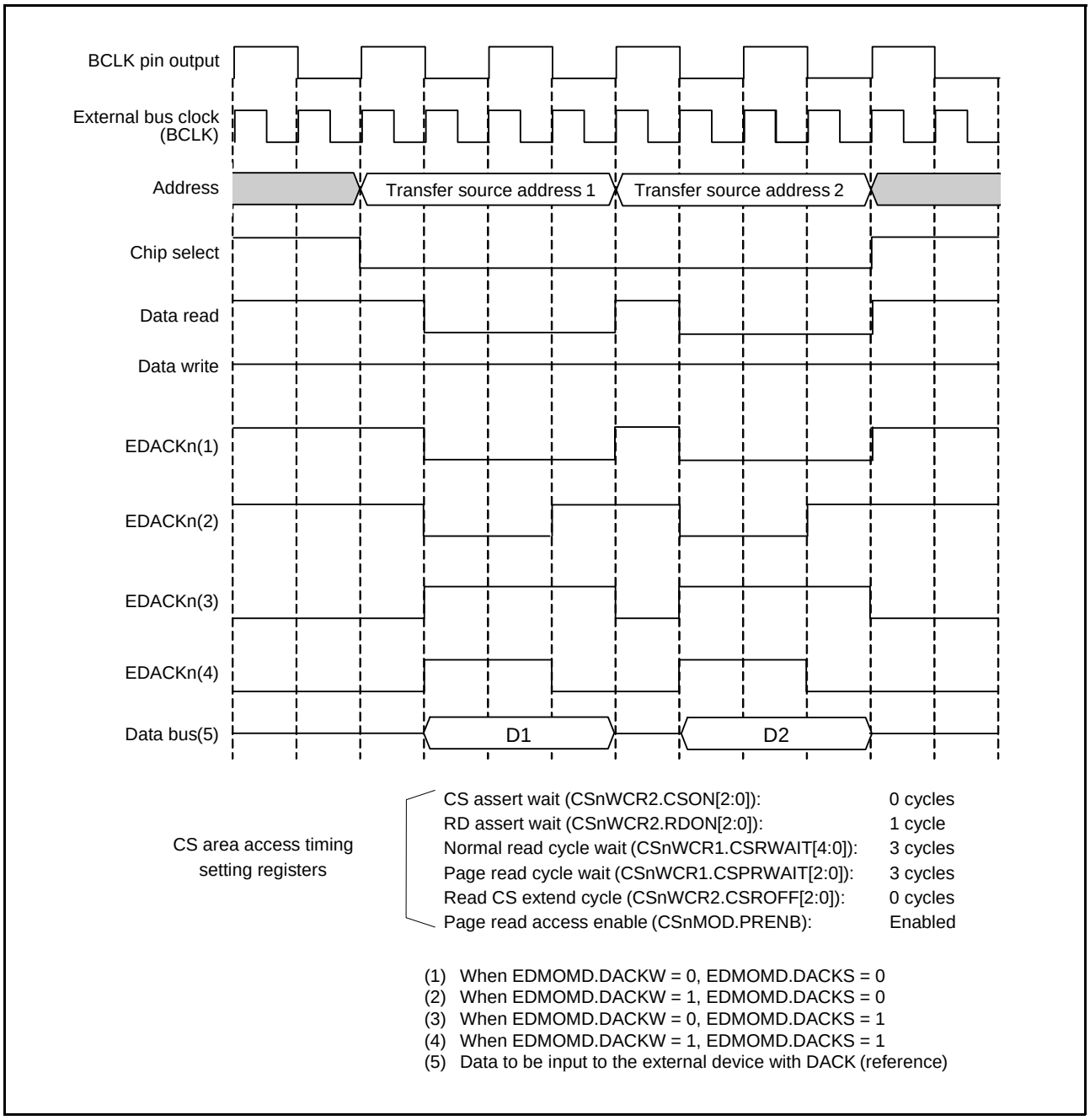


Figure 19.31 Operation Example in Block-Transfer (CS Area Read) Single Address Mode

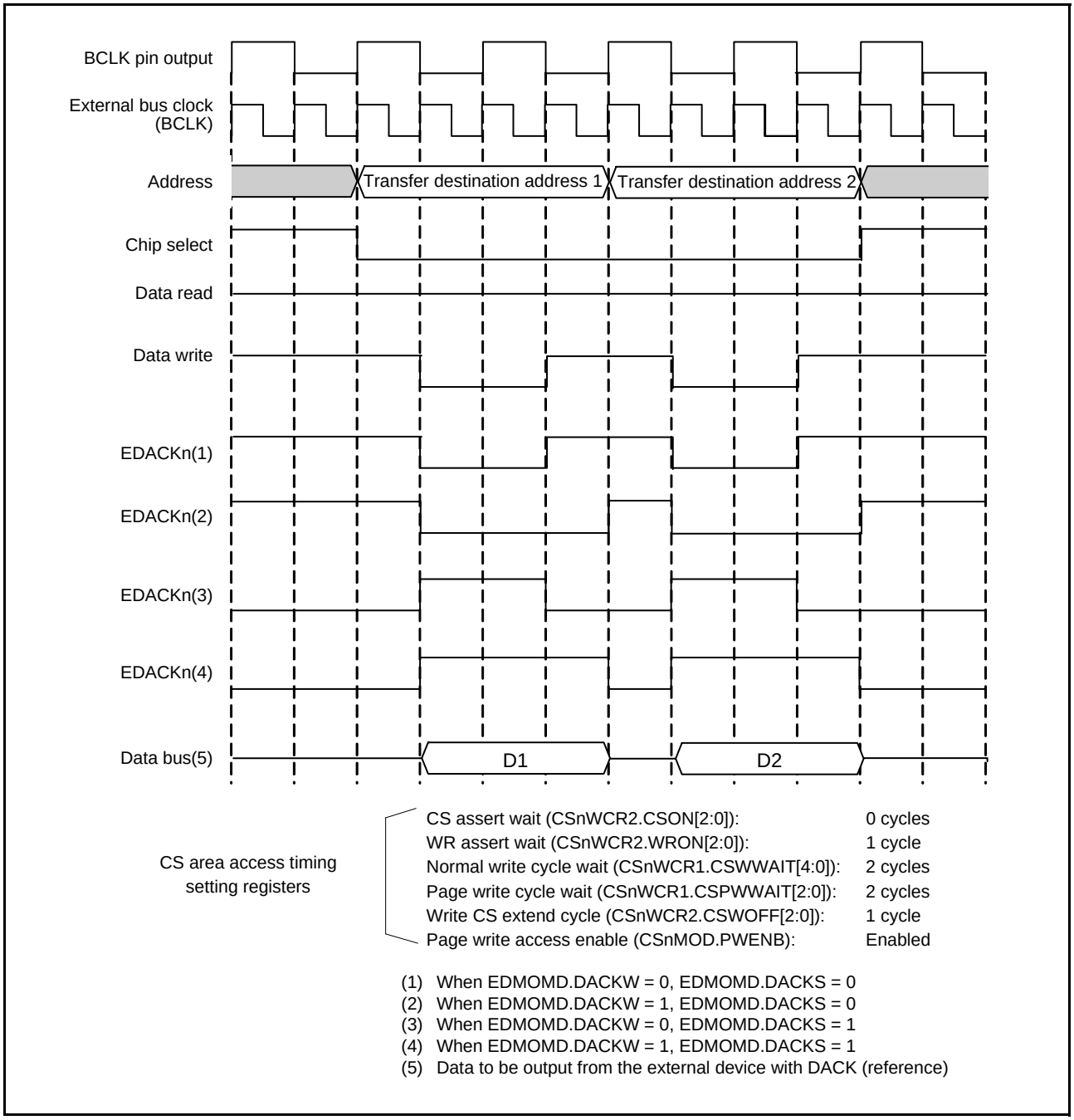


Figure 19.32 Operation Example in Block-Transfer (CS Area Write) Single Address Mode

19.9.4 EDACKn Operation Example in Block-Transfer (SDRAM Area) Single Address Mode

Figure 19.33 shows the operation example in which data is transferred from SDRAM to the device with EDACKn in block transfer mode (block size = four) when the SDRAM continuous access enable bit is enabled (SDAMOD.BE = 1) and EXDMACn.EDMOMD.DACKSEL set to 0.

EDACKn is asserted while SDRAM is outputting data.

Figure 19.34 shows the operation example in which data is transferred from the device with EDACKn to SDRAM in block transfer mode (block size = four) when the SDRAM continuous access enable bit is enabled (SDAMOD.BE = 1) and EXDMACn.EDMOMD.DACKSEL set to 0.

EDACKn is asserted while SDRAM is writing data.
Figure 19.35 shows the operation example in which data is transferred from SDRAM to the device with EDACKn in block transfer mode (block size = four) when the SDRAM continuous access enable bit is enabled (SDAMOD.BE = 1) and EXDMACn.EDMOMD.DACKSEL set to 1.
EDACKn is asserted while SDRAM is outputting data.
Figure 19.36 shows the operation example in which data is transferred from the device with EDACKn to SDRAM in block transfer mode (block size = four) when the SDRAM continuous access enable bit is enabled (SDAMOD.BE = 1) and EXDMACn.EDMOMD.DACKSEL set to 1.
EDACKn is asserted during SDRAM is writing data.
Figure 19.37 shows the operation example in which data is transferred from SDRAM to the device with EDACKn in block transfer mode (block size = two) when the SDRAM continuous access enable bit is disabled (SDAMOD.BE = 0) and EXDMACn.EDMOMD.DACKSEL set to 0.
Figure 19.38 shows the operation example in which data is transferred from the device with EDACKn to SDRAM in block transfer mode (block size = two) when the SDRAM continuous access enable bit is disabled (SDAMOD.BE = 0) and EXDMACn.EDMOMD.DACKSEL set to 0.
EDACKn is asserted while SDRAM is writing data.
For the SDRAM commands and SDRAM access timing setting registers, refer to section 16, Buses.

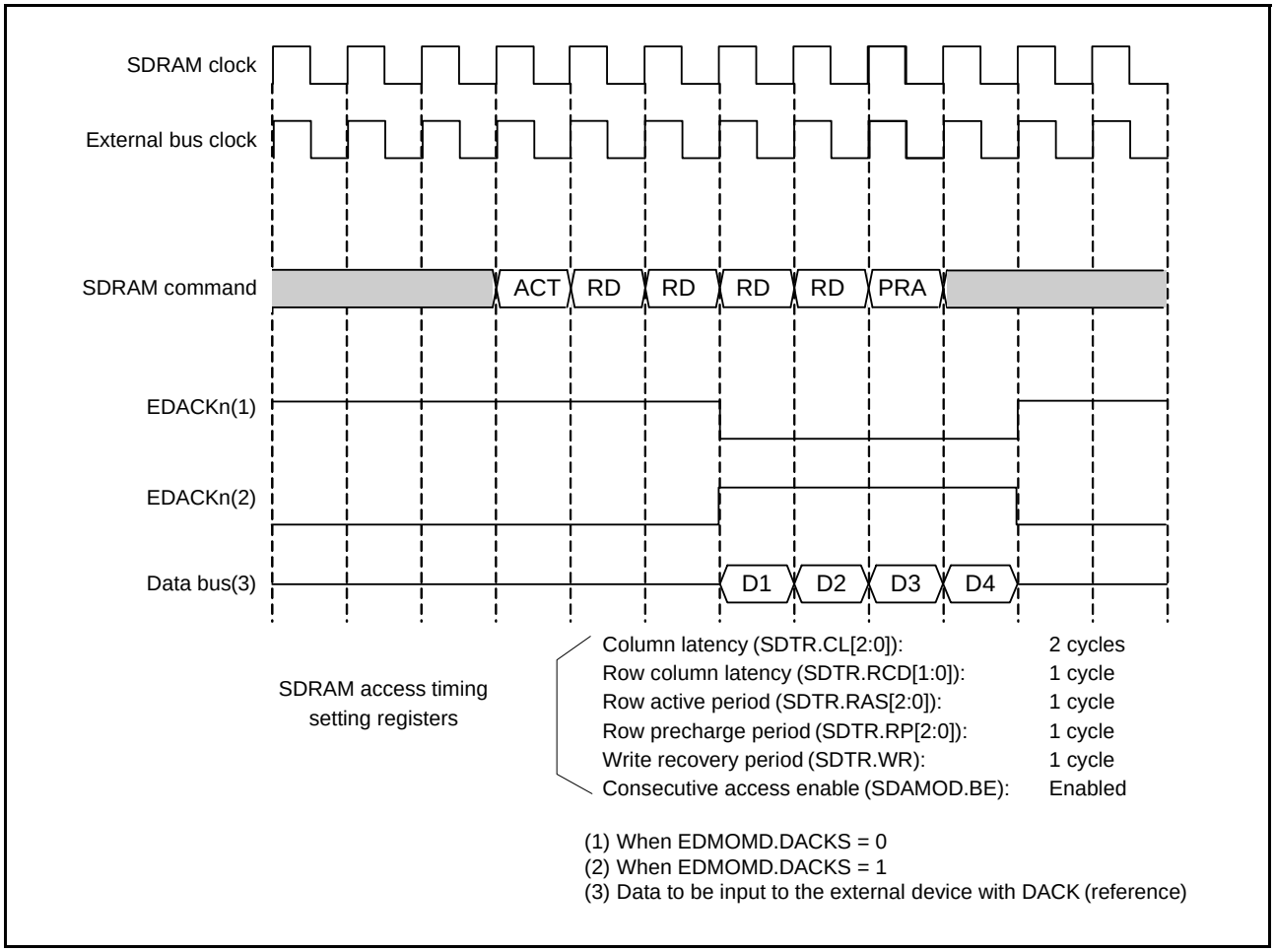


Figure 19.33 Operation Example in Block-Transfer (SDRAM Area Read: Consecutive Access Enabled, EXDMACn.EDMOMD.DACKSEL = 0) Single Address Mode

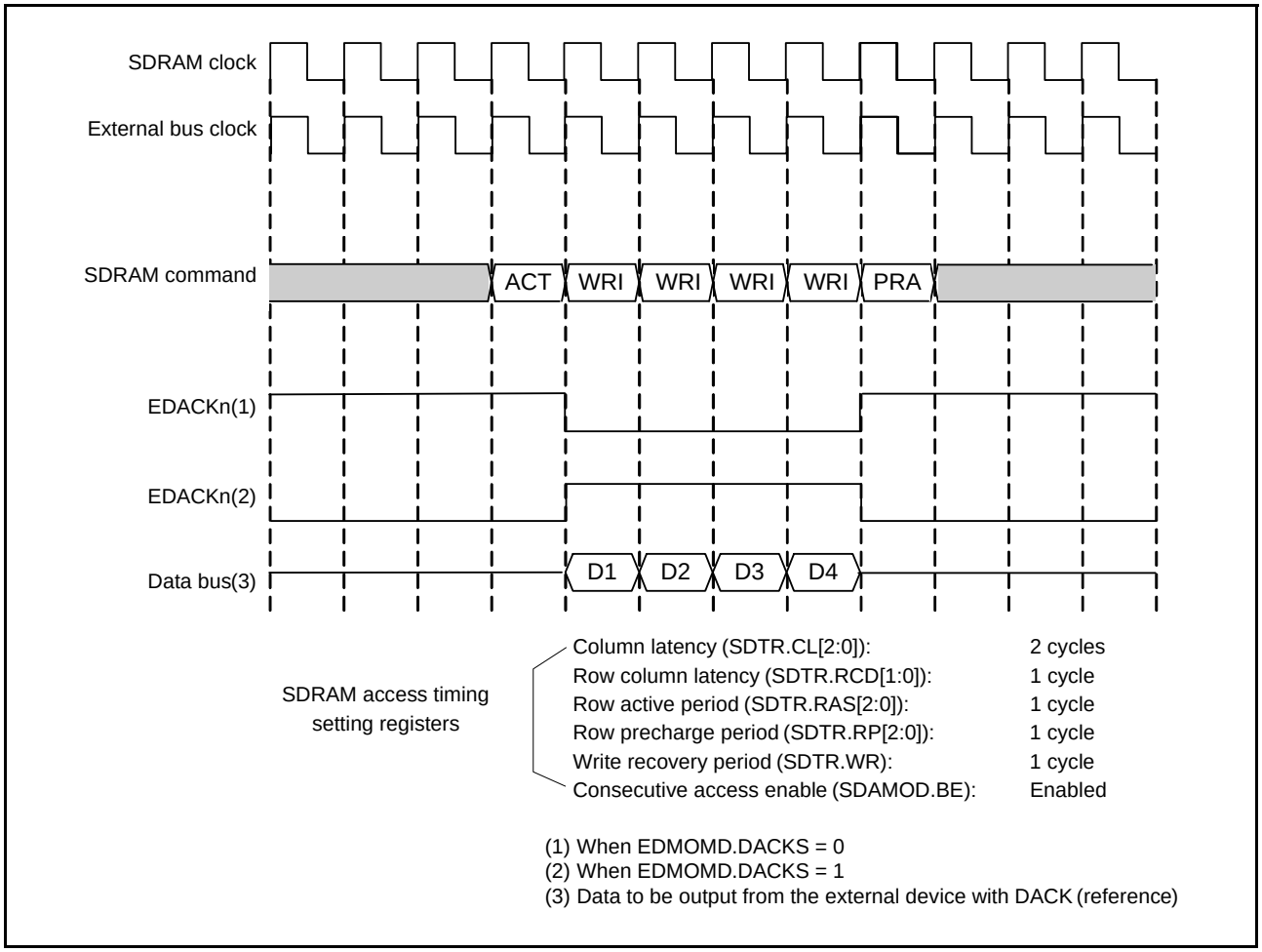


Figure 19.34 Operation Example in Block-Transfer (SDRAM Area Write: Consecutive Access Enabled, EXDMACn.EDMOMD.DACKSEL = 0) Single Address Mode

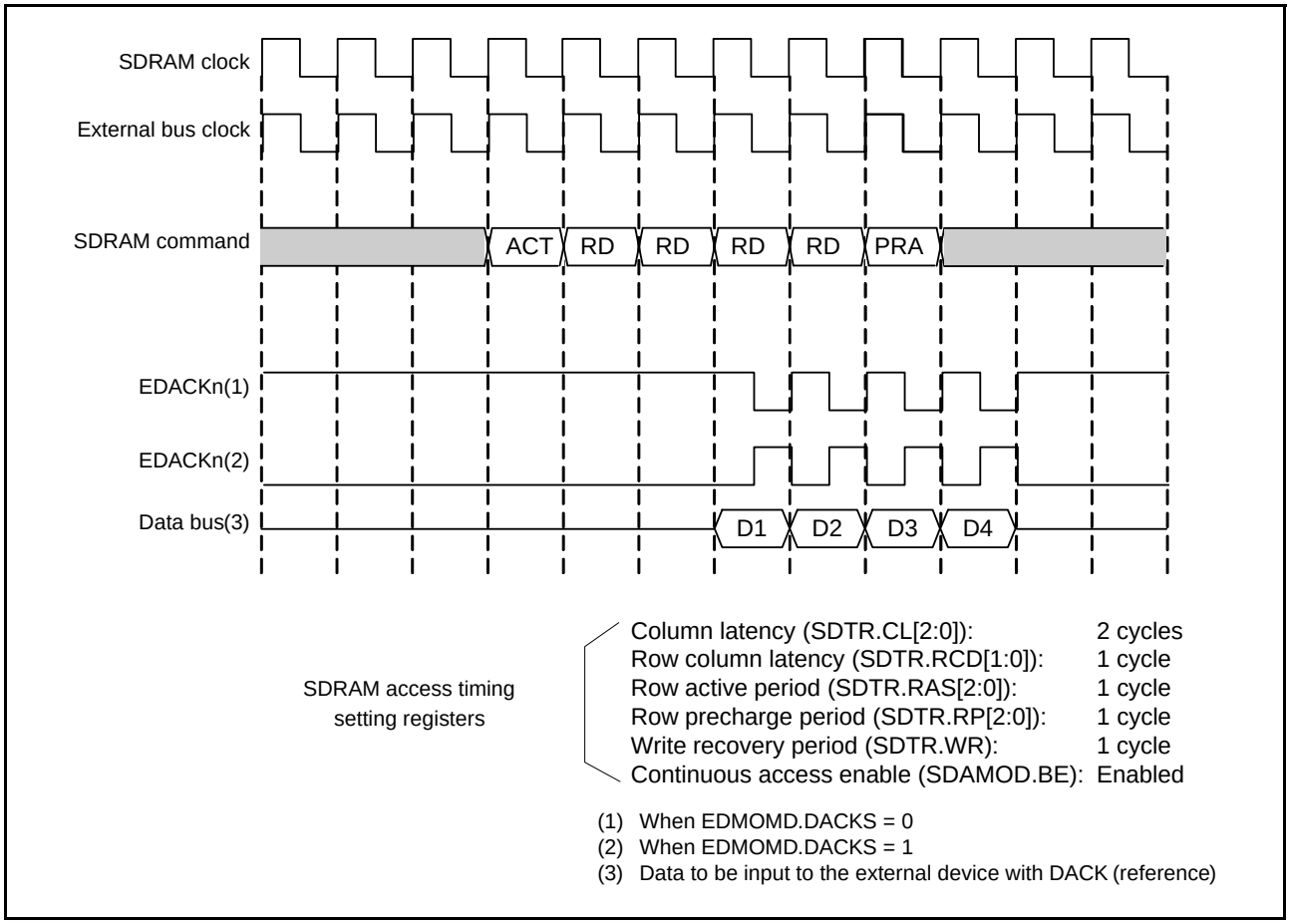


Figure 19.35 Operation Example in Block-Transfer (SDRAM Area Read: Continuous Access Enabled, EXDMACn.EDMOMD.DACKSEL = 1) Single Address Mode

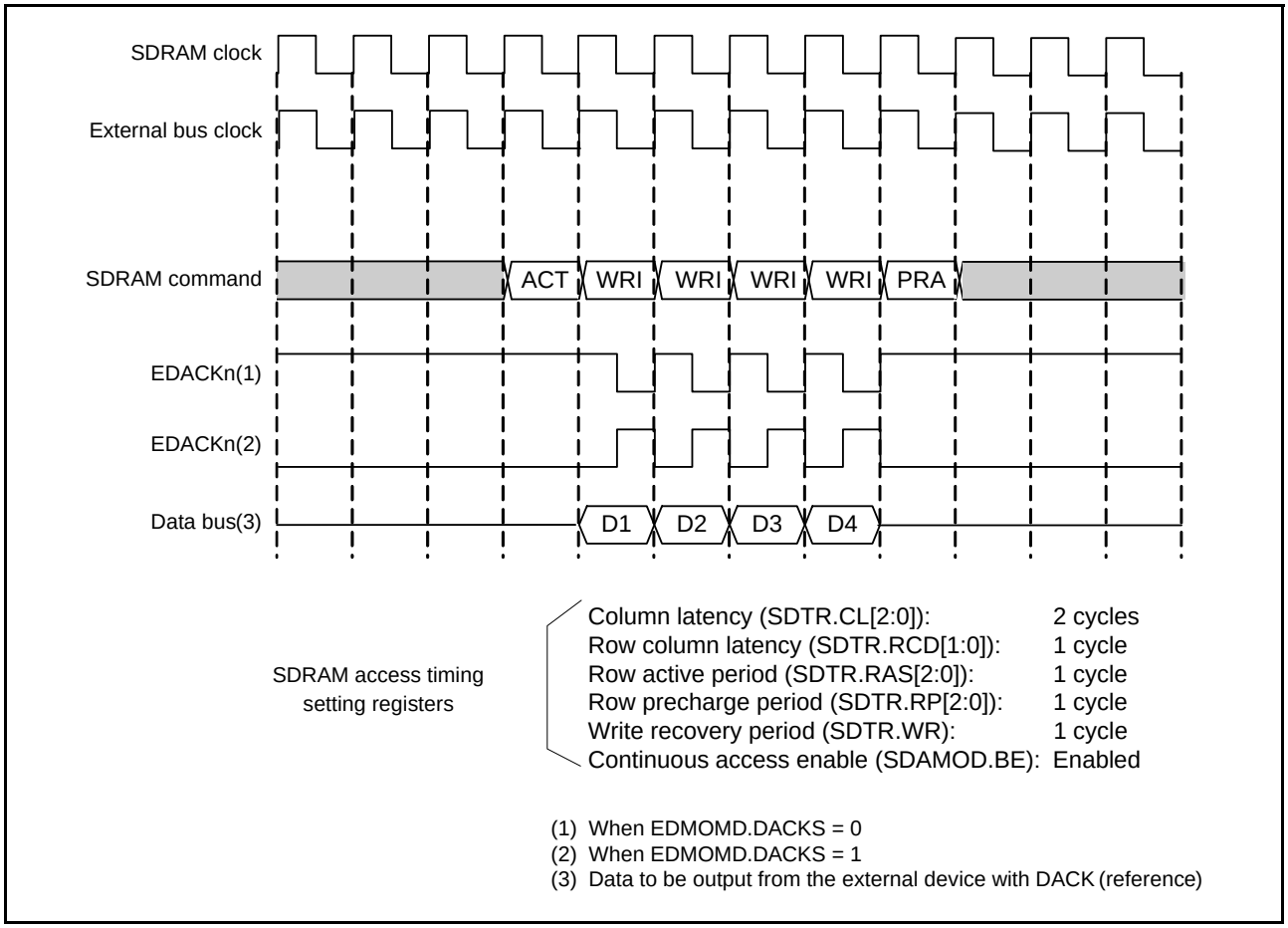


Figure 19.36 Operation Example in Block-Transfer (SDRAM Area Write: Continuous Access Enabled, EXDMACn.EDMOMD.DACKSEL = 1) Single Address Mode

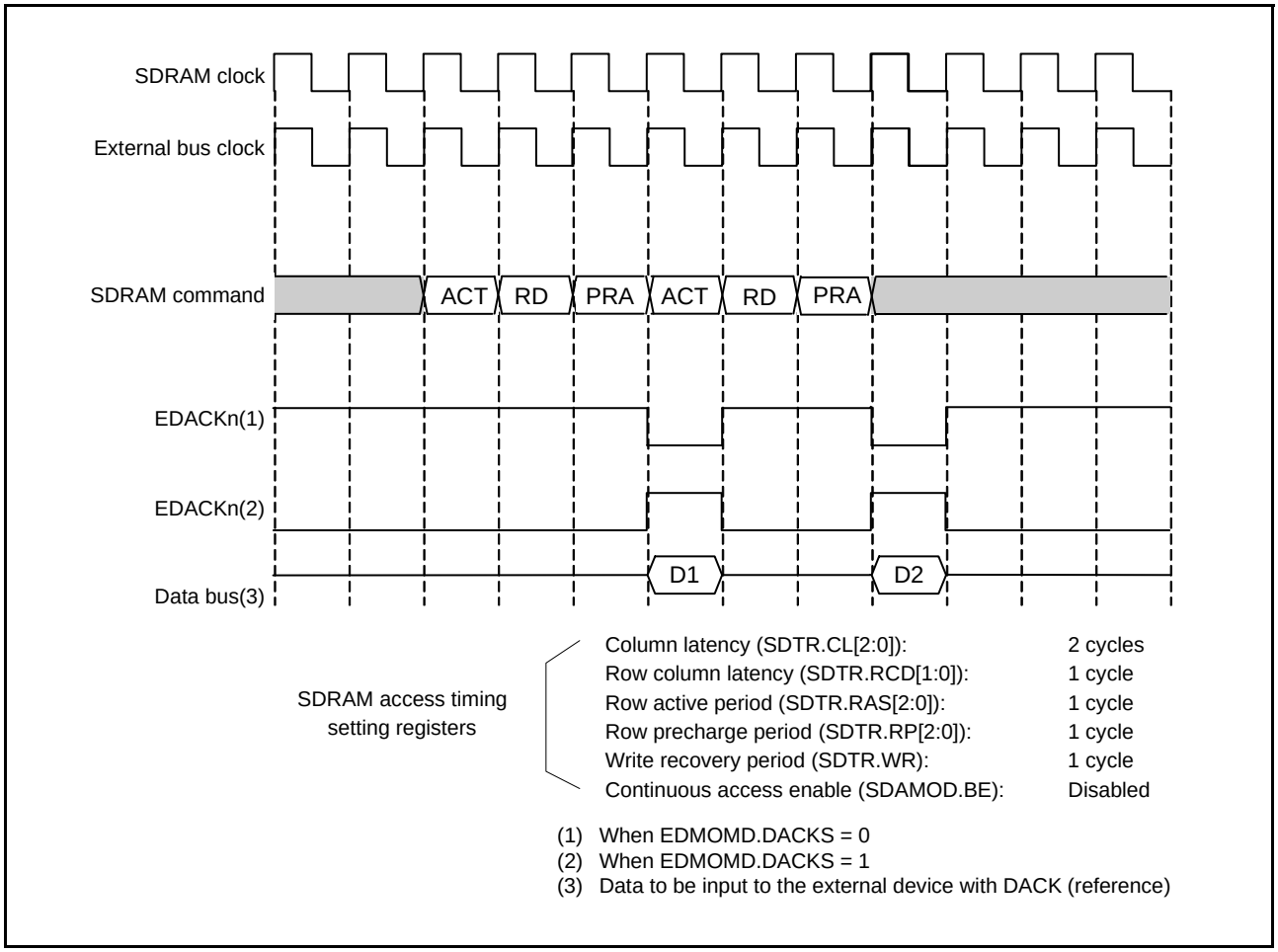


Figure 19.37 Operation Example in Block-Transfer (SDRAM Area Read: Continuous Access Disabled, EXDMACn.EDMOMD.DACKSEL = 0) Single Address Mode

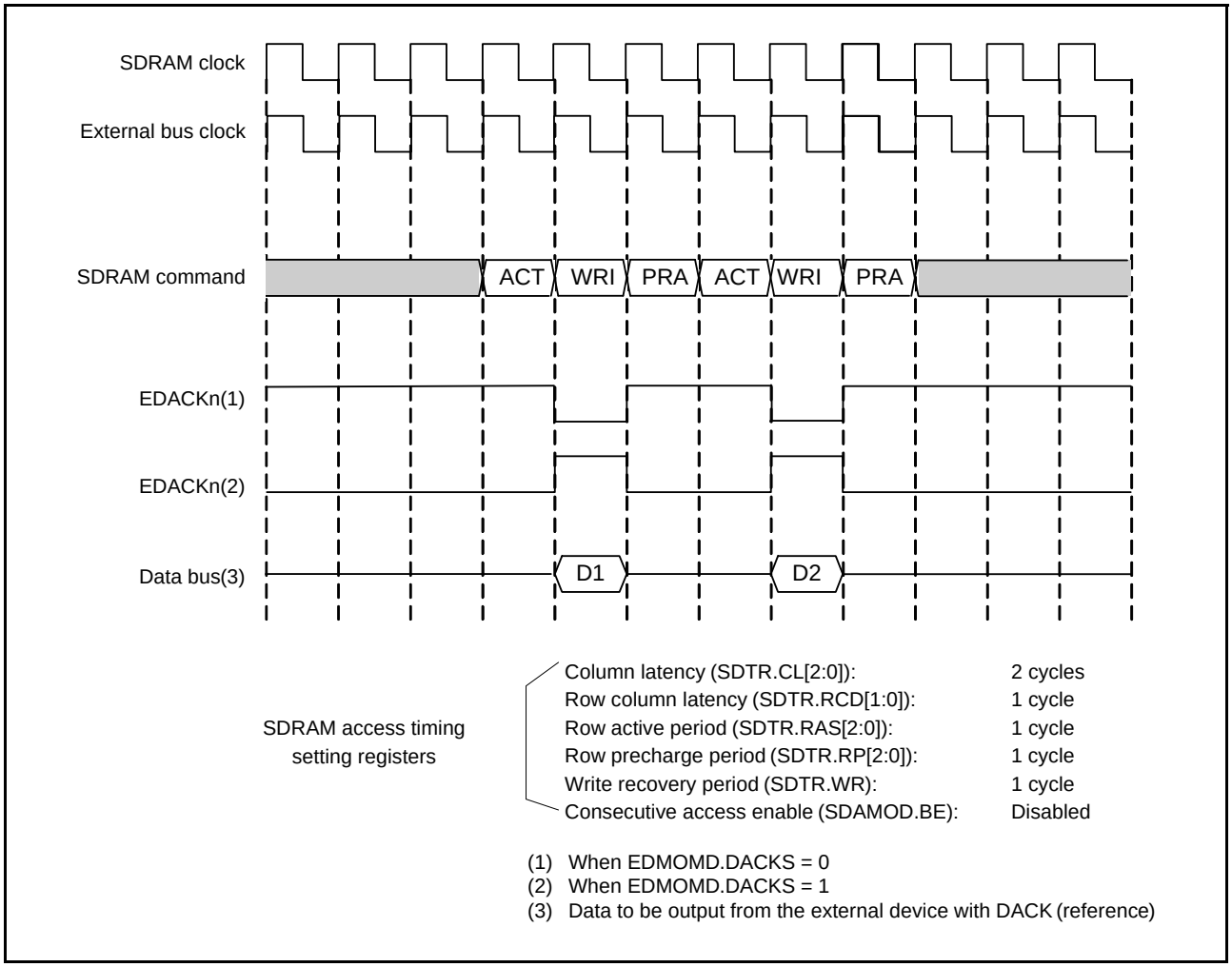


Figure 19.38 Operation Example in Block-Transfer (SDRAM Area Write: Consecutive Access Disabled) Single Address Mode

19.10 Usage Notes

19.10.1 Cluster Buffers

The EXDMAC provides eight 32-bit cluster buffers (CLSBR0 to CLSBR7), in which data is stored in the different manner depending on the transfer size setting (SZ[1:0] bits in EDMTMD of EXDMACn).
Figure 19.39 shows how data is stored in cluster buffers.

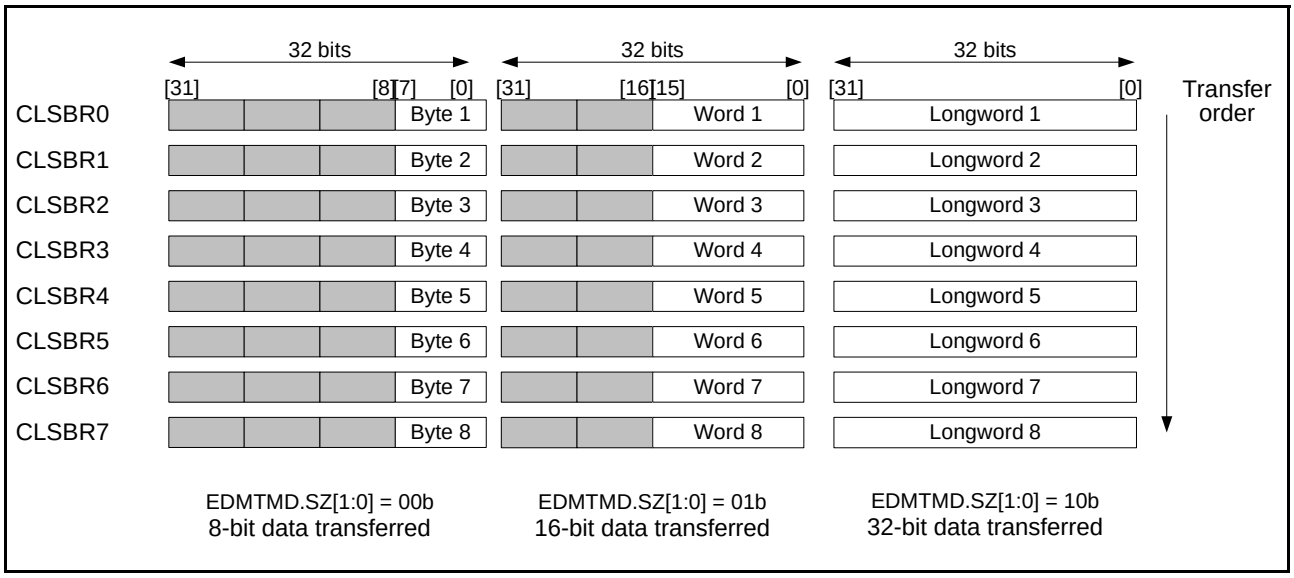


Figure 19.39 Data Storage in Cluster Buffers

19.10.2 Access to the Registers during DMA Transfer

Do not write to the EDMSAR, EDMDAR, EDMCRA, EDMCRB, EDMTMD, EDMOMD, EDMINT, EDMAMD, EDMOFR, and EDMRMD registers of EXDMACn while the EDMSTS.ACT flag of the same channel is set to 1 (DMA operating state) or the EDMCNT.DTE bit of the same channel is set to 1 (DMA transfer enabled).

19.10.3 DMA Transfer to Reserved Areas

DMA transfer to the reserved areas is prohibited. If such an access is made, transfer results are not guaranteed. For details on the reserved areas, refer to section 4, Address Space.

20. Data Transfer Controller (DTCb)

This MCU incorporates a data transfer controller (DTC).
The DTC is triggered by an interrupt request to perform data transfers.
In addition to the conventional methods of DTC transfer (normal, repeat, block, and chain), DTCb supports sequential transfer, in which it handles a series of transfers made up of a combination of the other methods. In sequential transfer, the data that is initially transferred selects one from possible 256 sequences for execution. The DTCb can divide one sequence into several transfers depending on how the parts of the sequence are combined.

20.1 Overview

Table 20.1 lists the specifications of the DTC, and Figure 20.1 shows a block diagram of the DTC.

Table 20.1 DTC Specifications

Item	Description
Number of transfer channels	• The same number as all interrupt sources that can start the DTC transfer.
Transfer modes	• Normal transfer mode A single transfer request leads to a single data transfer. • Repeat transfer mode A single transfer request leads to a single data transfer. The transfer address is returned to the transfer start address after the number of data transfers corresponding to “repeat size”. The maximum number of repeat transfers is 256, and the maximum data transfer size is 256 × 32 bits, 1024 bytes. • Block transfer mode A single transfer request leads to the transfer of a single block. The maximum block size is 256 × 32 bits = 1024 bytes.
Chain transfer	• Multiple types of data transfers can sequentially be executed in response to a single request. • Either “performed only when the transfer counter becomes 0” or “every time” can be selected.
Sequence transfer	A series of complicated transfers can be registered as a sequence. Any sequence can be selected by the transfer data and executed. • Only one trigger source can be set at a time. • Up to 256 sequences for a single trigger source • The data that is initially transferred in response to a transfer request determines a sequence • The whole sequence can be executed on a single request, or be suspended in the middle of the sequence and resumed on the next transfer request (division of sequence).
Transfer space	• In short-address mode: 16 Mbytes (Areas from 0000 0000h to 007F FFFFh and FF80 0000h to FFFF FFFFh except reserved areas) • In full-address mode: 4 Gbytes (Area from 0000 0000h to FFFF FFFFh except reserved areas)
Data transfer units	• Single data: 1 byte (8 bits), 1 word (16 bits), 1 longword (32 bits) • Single block size: 1 to 256 data
CPU interrupt source	• An interrupt request can be generated to the CPU on a request source for a data transfer. • An interrupt request can be generated to the CPU after a single data transfer. • An interrupt request can be generated to the CPU after data transfer of specified volume.
Event link function	An event link request is generated after one data transfer (for block, after one block transfer).
Read skip	Reading of the transfer information can be skipped when the same transfer is repeated.
Write-back skip	Write-back of the transferred data that is not updated can be skipped when the address of the transfer source or destination is fixed.
Write-back disable	Allows disabling the write-back of transfer information.
Displacement addition	The displacement value can be added to the transfer source address (for each transfer information)
Low power consumption function	Module stop state can be set.

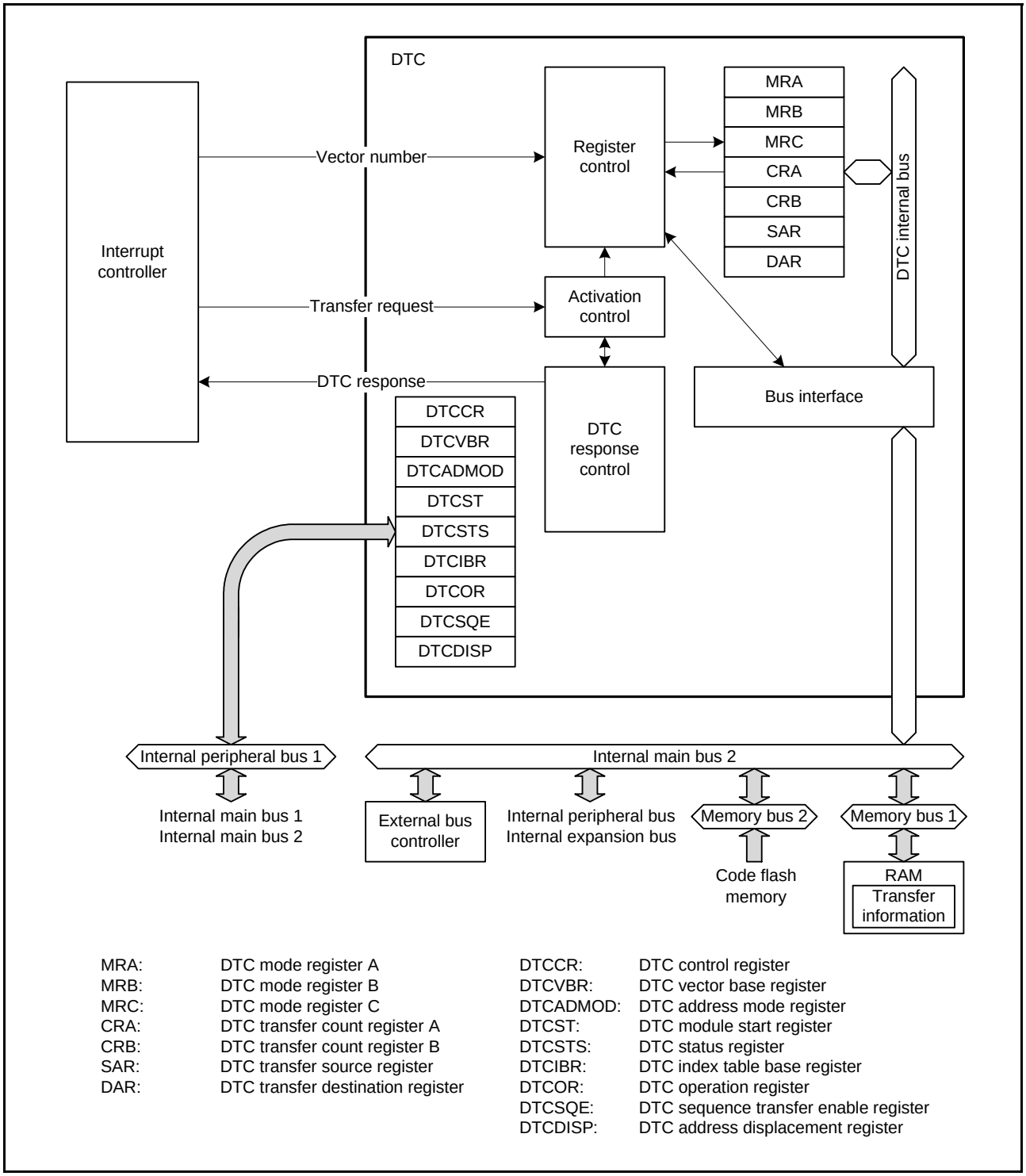


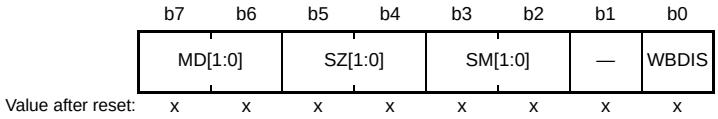
Figure 20.1 DTC Block Diagram

20.2 Register Descriptions

Registers MRA, MRB, MRC, SAR, DAR, CRA, and CRB are DTC internal registers, which cannot be directly accessed from the CPU. Values to be set in these DTC internal registers are placed in the RAM area as transfer information. When accepting a transfer request, the DTC reads the transfer information from the RAM area and sets it in the internal registers. After the data transfer ends, the values of the updated internal register are written back to the RAM area as transfer information.

20.2.1 DTC Mode Register A (MRA)

Address(es): (inaccessible directly from the CPU)



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	WBDIS	Write-back Disable	0: Writes back the transfer information on completion of the data transfer 1: Does not write back the transfer information on completion of the data transfer	—
b1	—	Reserved	Set this bit to 0.	—
b3, b2	SM[1:0]	Transfer Source Address Addressing Mode	b3 b2 0 0: The address in the SAR register is fixed. (write-back to SAR is skipped.) 0 1: The address in the SAR register is fixed. (write-back to SAR is skipped.) 1 0: The SAR value is incremented after a data transfer. (+1 when the SZ[1:0] bits are 00b, +2 when 01b, +4 when 10b) 1 1: The SAR value is decremented after a data transfer. (–1 when the SZ[1:0] bits are 00b, –2 when 01b, –4 when 10b)	—
b5, b4	SZ[1:0]	DTC Data Transfer Size	b5 b4 0 0: Byte (8-bit) transfer 0 1: Word (16-bit) transfer 1 0: Longword (32-bit) transfer 1 1: Setting prohibited	—
b7, b6	MD[1:0]	DTC Transfer Mode Select	b7 b6 0 0: Normal transfer mode 0 1: Repeat transfer mode 1 0: Block transfer mode 1 1: Setting prohibited	—

MRA register is used to select the DTC operating mode and cannot be accessed directly from the CPU.

WBDIS Bit (Write-back Disable)

The WBDIS bit selects whether to write back the transfer information.

When the bit is 0, updated transfer information is written back.

When the bit is 1, updated transfer information is not written back even with the setting of that address is incremented after a transfer, and the same data transfer is executed every time for each transfer request. The transfer information can be stored in ROM because the transfer information is not written back.

While the WBDIS bit is 1, operation for each transfer mode is as follows:

(1) Normal transfer and repeat transfer modes
1-byte, 1-word, or 1-longword of data is transferred on a single transfer request. The transfer address and transfer count are not updated so that the same transfer is repeated on each transfer request. When the transfer count is 1, the ICU.DTCERn.DTCE bit is not set to 0, and data transfer continues in response to the next transfer request.

(2) Block transfer mode
1-block of data is transferred on a single transfer request. The transfer address and transfer count are not updated so that the same block transfer is repeated on each transfer request. When the block transfer count is 1, the ICU.DTCERn.DTCE bit is not set to 0, and data transfer continues in response to the next transfer request.

When setting the DISPE bit to 1, set the MRA.WBDIS bit to 1 (does not write back the transfer information). If the value of the WBDIS bit in any transfer information is 1, set the DTCCR.RRS bit to 0 (so that reading of the transfer information is not skipped).

20.2.2 DTC Mode Register B (MRB)

Address(es): (inaccessible directly from the CPU)

	b7	b6	b5	b4	b3	b2	b1	b0
	CHNE	CHNS	DISEL	DTS	DM[1:0]		INDX	SQEND
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	SQEND	Sequence Transfer End	0: Continue the sequence transfer 1: End the sequence transfer	—
b1	INDX	Index Table Reference	0: Does not refer to the index table 1: Refers the index table based on the transferred data*1	—
b3, b2	DM[1:0]	Transfer Destination Address Addressing Mode	b3 b2 0 0: The address in the DAR register is fixed. (Write-back to DAR is skipped.) 0 1: The address in the DAR register is fixed. (Write-back to DAR is skipped.) 1 0: The DAR value is incremented after data transfer. (+1 when the MRA.SZ[1:0] bits are 00b, +2 when 01b, +4 when 10b) 1 1: The DAR value is decremented after data transfer. (–1 when the MRA.SZ[1:0] bits are 00b, –2 when 01b, –4 when 10b)	—
b4	DTS	DTC Transfer Mode Select	0: Transfer destination side is repeat area or block area. 1: Transfer source side is repeat area or block area.	—
b5	DISEL	DTC Interrupt Select	0: An interrupt request to the CPU is generated on completion of the specified number of data transfers. 1: An interrupt request to the CPU is generated for each data transfer.	—
b6	CHNS	DTC Chain Transfer Select	0: Chain transfer is performed on completion of each transfer. 1: Chain transfer is performed only when the transfer counter is changed from 1 to 0 or 1 to CRAH.	—
b7	CHNE	DTC Chain Transfer Enable	0: Chain transfer is disabled. 1: Chain transfer is enabled.	—

Note 1. Set the MRA.MD[1:0] bits to 00b (normal transfer mode) when setting the INDX bit to 1.

MRB register is used to select the DTC operating mode and cannot be accessed directly from the CPU.

SQEND Bit (Sequence Transfer End)

The SQEND bit selects whether to continue or end sequence transfer. Refer to Table 20.2 for details.
This bit can only be set to 1 for transfer information referred to by the DTC index table. Set this bit to 0 for transfer information referred to by the DTC vector table.

INDX Bit (Index Table Reference)

When the value of the INDX bit in transfer information that is read is 1, a sequence transfer proceeds. Refer to Table 20.2 for details.
Set this bit to 0 for transfer information which is not associated with sequence transfer or is not intended to start sequence transfer. Do not allow transfer requests to be generated by the sources different from that specified in the DTCSQE register but having the INDX bit set to 1.

Table 20.2 Values of Bits CHNE, SQEND, and INDX in the Sequence Transfer and DTC Operation

CHNE Bit	SQEND Bit	INDX Bit	Operation	Usage
0	0	1	Start sequence transfer	Use this setting for the transfer information that is first read in response to a transfer request from the source specified in the DTCSQE register.
1	0	0	Continue sequence transfer	Use this setting for the first or intermediate transfer information in a sequence.
0	0	0	Suspend sequence transfer	Use this setting for the first or intermediate transfer information in a sequence.
0	1	0	End sequence transfer	Use this setting with the last transfer information in a sequence.
0	1	1	End current sequence transfer and start new sequence transfer	Use this setting with the last transfer information in a sequence.

Note: Do not set the values other than listed above.

DTS Bit (DTC Transfer Mode Select)

The DTS bit specifies the side (transfer source or destination) to be a repeat area or block area in repeat transfer mode or block transfer mode.

CHNS Bit (DTC Chain Transfer Select)

The CHNS bit selects the chain transfer condition.

When the CHNE bit is 0, setting of the CHNS bit is ignored. For details on the conditions to select the chain transfer, refer to Table 20.4, Chain Transfer Conditions.

When the next transfer is chain transfer, completion of the specified number of transfers is not determined, the interrupt status flag for the request source is not cleared, and an interrupt request to the CPU is not generated.

CHNE Bit (DTC Chain Transfer Enable)

The CHNE bit enables or disables chain transfer.

The chain transfer condition is selected by the CHNS bit.

For details of chain transfer, refer to section 20.4.6, Chain Transfer.

Refer to Table 20.2 for the setting value to be used in the sequence transfer.

20.2.3 DTC Mode Register C (MRC)

Address(es): (inaccessible directly from the CPU)

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

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	DISPE	Displacement Addition	0: The displacement value is not added to the transfer source address. 1: The displacement value is added to the transfer source address.	—
b7 to b1	—	Reserved	Set these bits to 0.	—

The MRC register is used to select DTC operating mode and cannot be accessed directly from the CPU.
This register can only be used in full-address mode, but not in short-address mode. Therefore, set the DTCADMOD.SHORT bit to 0 (full-address mode) when using the displacement addition function.

DISPE Bit (Displacement Addition)

This bit specifies whether to use the SAR + DTCDISP value as the transfer source address.
When setting the DISPE bit to 1, set the MRA.WBDIS bit to 1 (does not write back the transfer information) and set the DTCCR.RRS bit to 0 (transfer information read is not skipped).

20.2.4 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

SAR register is used to set the transfer source start address.
In full-address mode, 32 bits are valid.
In short-address mode, lower 24 bits are valid and upper 8 bits (b31 to b24) are ignored. The address of this register is extended by the value specified by b23.
SAR register cannot be accessed directly from the CPU.

20.2.5 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

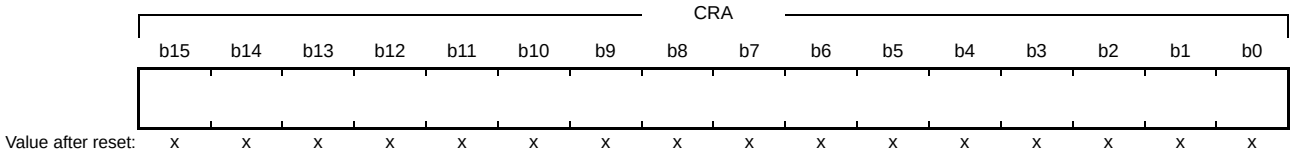
x: Undefined

DAR register is used to set the transfer destination start address.
In full-address mode, 32 bits are valid.
In short-address mode, lower 24 bits are valid and upper 8 bits (b31 to b24) are ignored. The address of this register is extended by the value specified by b23.
DAR register cannot be accessed directly from the CPU.

20.2.6 DTC Transfer Count Register A (CRA)

- Normal transfer mode

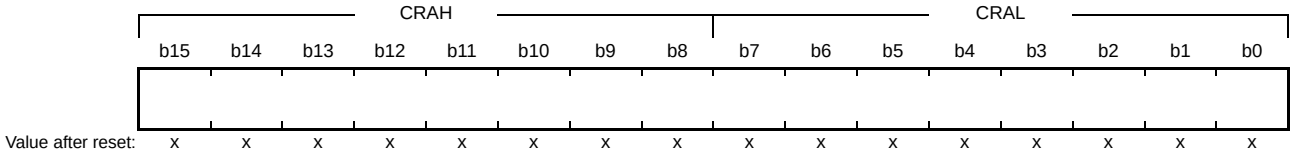
Address(es): (inaccessible directly from the CPU)



x: Undefined

- Repeat transfer mode/block transfer mode

Address(es): (inaccessible directly from the CPU)



x: Undefined

Symbol	Register Name	Description	R/W
CRAL	Transfer Counter A Lower Register	Set transfer count. This register functions as a transfer counter during data transfer.	—
CRAH	Transfer Counter A Upper Register	Set transfer count. This register functions as a reload register during data transfer.	—

Note: The function depends on transfer mode.
Note: Set CRAH and CRAL to the same value in repeat transfer mode and block transfer mode.

This register is for counting the number of transfers and cannot be accessed directly from the CPU.

(1) Normal transfer mode (MRA.MD[1:0] bits = 00b)

CRA register functions as a 16-bit transfer counter in normal transfer mode.
The transfer count is 1, 65535, and 65536 when the set value is 0001h, FFFFh, and 0000h, respectively.
The CRA value is decremented (–1) at each data transfer.

(2) Repeat transfer mode (MRA.MD[1:0] bits = 01b)

The CRAH register retains the transfer count and the CRAL register functions as an 8-bit transfer counter.
The transfer count is 1, 255, and 256 when the set value is 01h, FFh, and 00h, respectively.
The CRAL value is decremented (–1) at each data transfer. When it reaches 00h, the CRAH value is reloaded to the CRAL register.

(3) Block transfer mode (MRA.MD[1:0] bits = 10b)

The CRAH register retains the block size and the CRAL register functions as an 8-bit block size counter.
The transfer count is 1, 255, and 256 when the set value is 01h, FFh, and 00h, respectively.
The CRAL value is decremented (–1) at each data transfer. When it reaches 00h, the CRAH value is reloaded to the CRAL register.

20.2.7 DTC Transfer Count Register B (CRB)

Address(es): (inaccessible directly from the CPU)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

CRB register is used to set the block transfer count for block transfer mode and cannot be accessed directly from the CPU.
The transfer count is 1, 65535, and 65536 when the set value is 0001h, FFFFh, and 0000h, respectively.
The CRB value is decremented (–1) when the final data of a single block size is transferred.
When normal transfer mode or repeat transfer mode is selected, this register is not used and the set value is ignored.

20.2.8 DTC Control Register (DTCCR)

Address(es): DTC.DTCCR 0008 2400h

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

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b4	RRS	DTC Transfer Information Read Skip Enable*1	0: Transfer information read is not skipped. 1: Transfer information read is skipped when vector numbers match.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Set this bit to 0 when using the sequence transfer.

DTCCR register is used to control the DTC operation.

RRS Bit (DTC Transfer Information Read Skip Enable)

The DTC vector number is compared with the vector number in the previous data transfer.
When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the transferred information. However, when the previous transfer was chain transfer, the transferred information is read regardless of the value of the RRS bit.
Furthermore, when the transfer counter (CRA register) became 0 during the previous normal transfer and when the transfer counter (CRB register) became 0 during the previous block transfer, the transferred information is read regardless of the RRS bit value.
If the value of the MRA.WBDIS bit in any transfer information is 1, set the RRS bit to 0. Note that the MRA.WBDIS bit should be set to 1 when the MRC.DISPE bit is set to 1.
Like chain transfer, sequence transfer handles sequences of multiple types of data transfer. When sequence transfer is to be used, set the RRS bit to 0 so that the previous data transfer will not be repeated.

20.2.9 DTC Vector Base Register (DTCVBR)

Address(es): DTC.DTCVBR 0008 2404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

The DTCVBR register is used to set the base address for calculating the address to which the DTC vector is allocated. Values for the upper 4 bits (b31 to b28) cannot be written but reflect the value written to b27. The lower 10 bits are reserved and the values are fixed to 0. Write 0 to the lower 10 bits if necessary.
It can be set in the range of 0000 0000h to 07FF FC00h and F800 0000h to FFFF FC00h in 1-Kbyte units.

20.2.10 DTC Address Mode Register (DTCADMOD)

Address(es): DTC.DTCADMOD 0008 2408h

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

Bit	Symbol	Bit Name	Description	R/W
b0	SHORT	Short-Address Mode Set*1	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

Note 1. Set this bit to 0 (full-address mode) when using the sequence transfer.

DTCADMOD register is used to specify the area accessible by the DTC.

SHORT Bit (Short-Address Mode Set)

This bit is used to select address mode of registers SAR and DAR.
Full-address mode allows the DTC to access to a 4-Gbyte space (0000 0000h to FFFF FFFFh).
Short-address mode allows the DTC to access to a 16-Mbyte space (0000 0000h to 007F FFFFh and FF80 0000h to FFFF FFFFh).

20.2.11 DTC Module Start Register (DTCST)

Address(es): DTC.DTCST 0008 240Ch

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

Bit	Symbol	Bit Name	Description	R/W
b0	DTCST	DTC Module Start	0: DTC module stop 1: DTC module start	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DTCST Bit (DTC Module Start)

Set the DTCST bit to 1 to enable the DTC to accept transfer requests. When this bit is set to 0, transfer requests are no longer accepted.

If this bit is set to 0 during data transfer, the accepted transfer request is active until the processing is completed.

Set the DTCST bit to 0 before making a transition to the module stop state, all-module clock stop mode, software standby mode, or deep software standby mode.

Set the DTCST bit to 1 to resume the data transfer after returning from the module stop state, all-module clock stop mode, or software standby mode.

For details on transitions to the module stop state, all-module clock stop mode, software standby mode, and deep software standby mode, refer to section 20.9, Low Power Consumption Function, and section 11, Low Power Consumption.

20.2.12 DTC Status Register (DTCSTS)

Address(es): DTC.DTCSTS 0008 240Eh

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

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	VECN[7:0]	DTC Active Vector Number Monitoring Flag	These bits indicate the vector number for the request source when data transfer is in progress. The value is only valid if data transfer is in progress (the value of the ACT flag is 1).	R
b14 to b8	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b15	ACT	DTC Active Flag	0: Data transfer is not in progress. 1: Data transfer is in progress.	R

VECN[7:0] Flags (DTC Active Vector Number Monitoring Flag)

While data transfer is in progress, these bits indicate the vector number corresponding to the request source for the transfer.

When the DTCSTS register is read, the value of the VECN[7:0] flags is valid if the value of the ACT flag was 1 (data transfer is in progress) and invalid if the value of the ACT flag was 0 (data transfer is not in progress).

For the correspondence between the DTC request sources and the vector addresses, refer to Table 15.5, Interrupt Vector Table in section 15, Interrupt Controller (ICUE).

ACT Flag (DTC Active Flag)

This flag indicates the state of data transfer operation.

[Setting condition]

- When the data transfer is started by a transfer request.
- When the sequence transfer is resumed.

[Clearing condition]

- When the data transfer is completed in response to a transfer request.
- When the sequence transfer is suspended.

20.2.13 DTC Index Table Base Register (DTCIBR)

Address(es): DTC.DTCIBR 0008 2410h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

The DTCIBR register is used to set the base address for calculating the address to which the DTC index is allocated.
Values for the upper 4 bits (b31 to b28) cannot be written but reflect the value written to b27.
The lower 10 bits (b9 to b0) are reserved bits and fixed to 0. When writing this register, set these bits to 0.
It can be set in the range of 0000 0000h to 07FF FC00h and F800 0000h to FFFF FC00h in 1-Kbyte units.

20.2.14 DTC Operation Register (DTCOR)

Address(es): DTC.DTCOR 0008 2414h

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

Bit	Symbol	Bit Name	Description	R/W
b0	SQTFRL	Sequence Transfer Terminate	Writing 1 to this bit terminates the sequence transfer in progress. This bit is read as 0.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

The DTCOR register sets the operation of the DTC module.

SQTFRL Bit (Sequence Transfer Terminate)

Setting the SQTFRL bit to 1 terminates the sequence transfer in progress.

When the DTCSQE.ESPSEL bit is 1 (Sequence transfer is enabled), follow the procedure shown in Figure 20.2 to terminate the sequence transfer.

Writing 1 to the bit, while no sequence transfer is performed, have no effect.

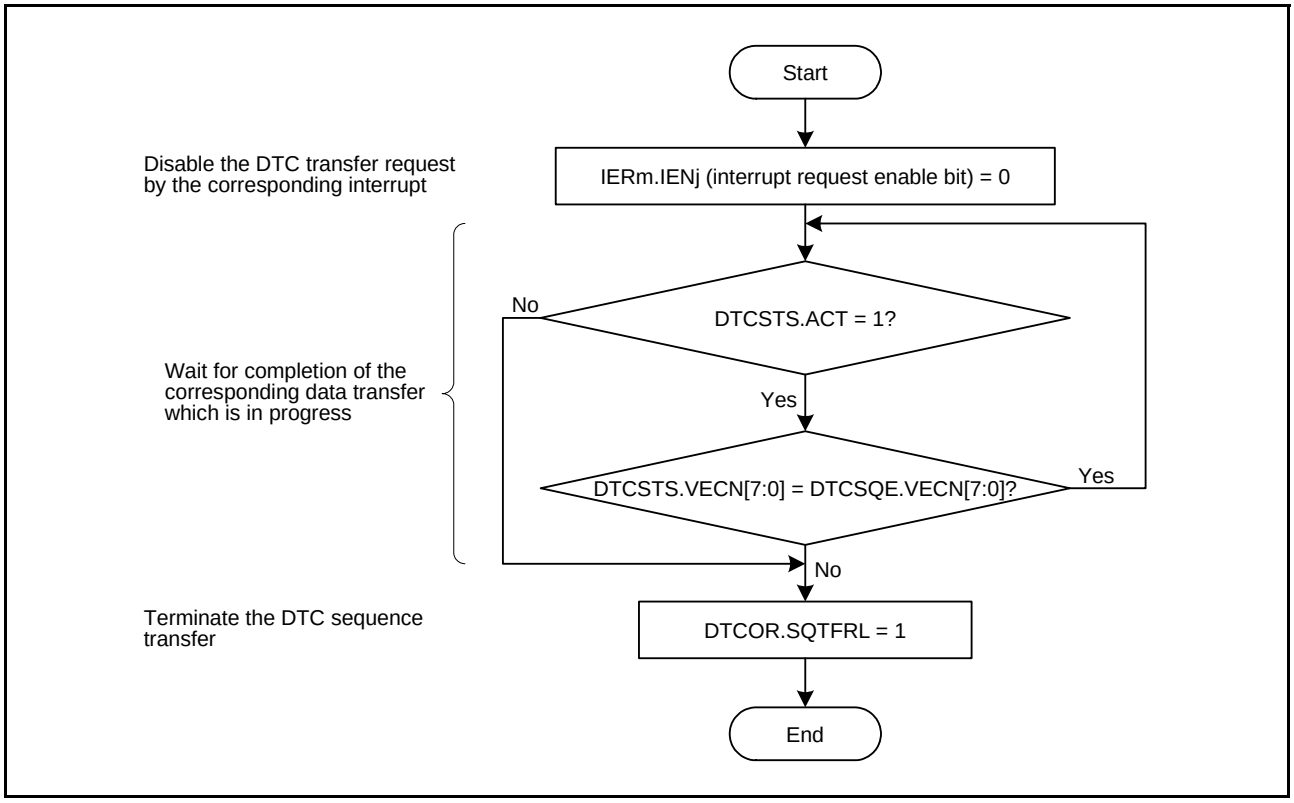


Figure 20.2 Procedure to Terminate Sequence Transfer

20.2.15 DTC Sequence Transfer Enable Register (DTCSQE)

Address(es): DTC.DTCSQE 0008 2416h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ESPSEL	—	—	—	—	—	—	—	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]	Sequence Transfer Vector Number Setting	Specify the vector number by which a sequence transfer is enabled. The value is only valid when the ESPSEL bit is 1.	R/W
b14 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R
b15	ESPSEL	Sequence Transfer Enable	0: Sequence transfer is disabled. 1: Sequence transfer is enabled.	R/W

The DTCSQE register is used to specify sequence transfer. Follow Figure 20.24 for details on the setting procedure.

VECN[7:0] Bit (Sequence Transfer Vector Number Setting)

This bit is used to specify for which vector number to perform sequence transfer. Sequence transfer can occur only for this trigger source.

Table 15.5, Interrupt Vector Table in section 15, Interrupt Controller (ICUE) shows the relationship between the trigger source and the vector number.

ESPSEL Bit (Sequence Transfer Enable)

The ESPSEL bit specifies whether sequence transfer is used.

Set the DTCADMOD.SHORT bit to 0 (full address mode), when setting the ESPSEL bit to 1.

20.2.16 DTC Address Displacement Register (DTCDISP)

Address(es): DTC.DTCDISP 0008 2418h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

The DTCDISP register is used to specify the displacement value to add to the DTC transfer source address.

If MRC.DISPE bit is 1, the value SAR + DTCDISP is used as the transfer source address.

20.3 Request Sources

The DTC data transfer is triggered by an interrupt request. Setting the ICU.DTCERn.DTCE bit (n = interrupt vector number) to 1 selects the corresponding interrupt request as a request source for the DTC.
For the correspondence between the DTC request sources and the vector addresses, refer to Table 15.5, Interrupt Vector Table in section 15, Interrupt Controller (ICUE). For request by software, refer to section 15.2.5, Software Interrupt Generation Register (SWINTR) and section 15.2.6, Software Interrupt 2 Generation Register (SWINT2R) in section 15, Interrupt Controller (ICUE).
Once the DTC has accepted a transfer request, it does not accept another transfer request until transfer for that single request is completed, regardless of the priority of the requests.
When multiple transfer requests are generated during data transfer by the DMAC/DTC, the request with the highest priority on completion of the current transfer is accepted. When multiple transfer requests are generated while the DTCST.DTCST bit is 0 (DTC module stop), the request with the highest priority at the moment when the bit is subsequently set to 1 (DTC module start) is accepted.

The DTC performs the following operations at the start of a single data transfer (or the last of the consecutive transfers in the case of a chain transfer).

- On completion of a specified number of data transfer, the ICU.DTCERn.DTCE bit is set to 0 and an interrupt is requested to the CPU.
- If the MRB.DISEL bit is 1, an interrupt is requested to the CPU on completion of data transfer.
- For the other transfers, the interrupt status flag of the request source is set to 0 at the start of data transfer.

20.3.1 Allocating Transfer Information and DTC Vector Table

The DTC reads the start address of the transfer information corresponding to each request source from the vector table and reads the transfer information starting at that address.
The vector table should be located so that the lower 10 bits of the base address (start address) are 0. Use the DTC vector base register (DTCVBR) to set the base address of the DTC vector table.
Transfer information is allocated in the RAM area. Transfer information can be allocated in the ROM area when the MRA.WBDIS bit is set to 1. The start address of the transfer information n with vector number n should be allocated at DTCVBR + 4n.
Transfer information should be aligned on a 4-byte boundary. The size of a transfer information is 12 bytes in short-address mode or 16 bytes in full-address mode. Use the DTCADMOD.SHORT bit to select short-address mode (SHORT bit = 1) or full-address mode (SHORT bit = 0).
Figure 20.3 shows the relationship between the DTC vector table and transfer information.
Figure 20.4 shows the allocation of transfer information in the RAM area. The lower addresses vary according to the endian of the corresponding allocation area. For details, refer to section 20.10.2, Allocating Transfer Information.

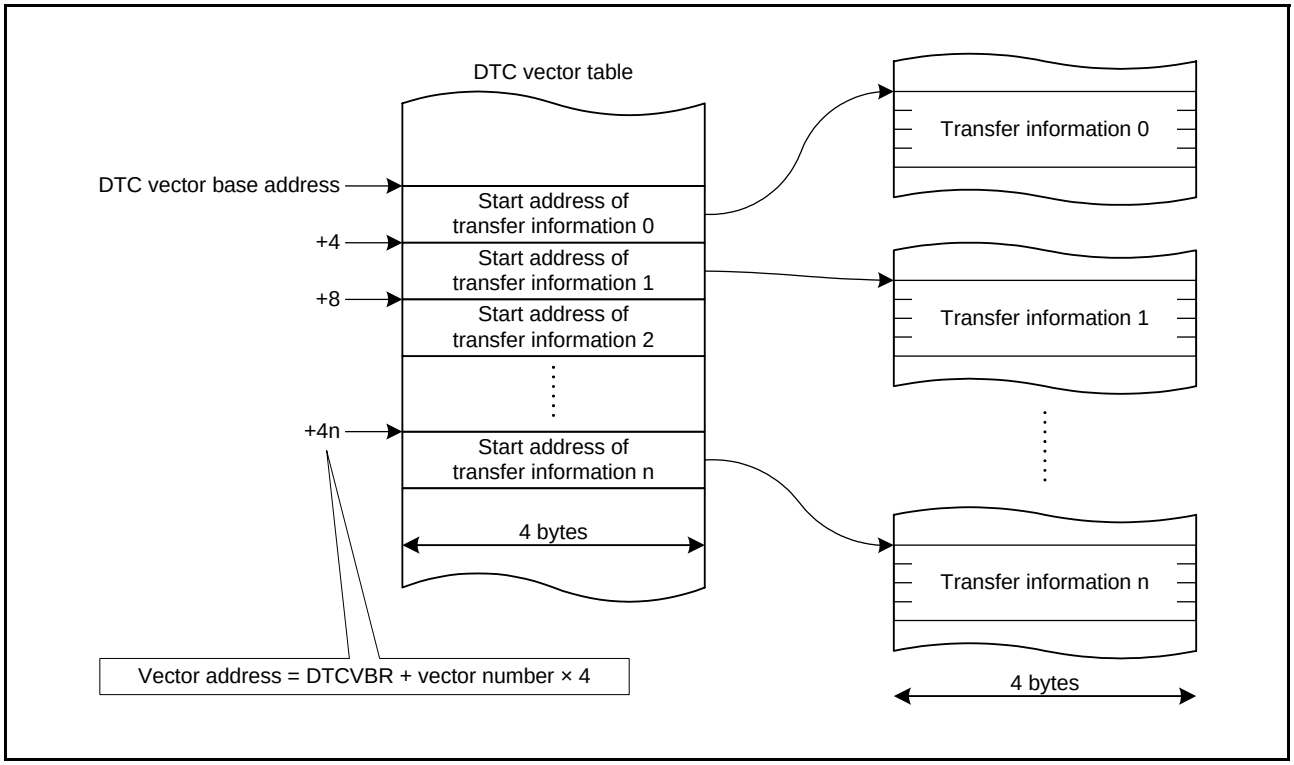


Figure 20.3 DTC Vector Table and Transfer Information

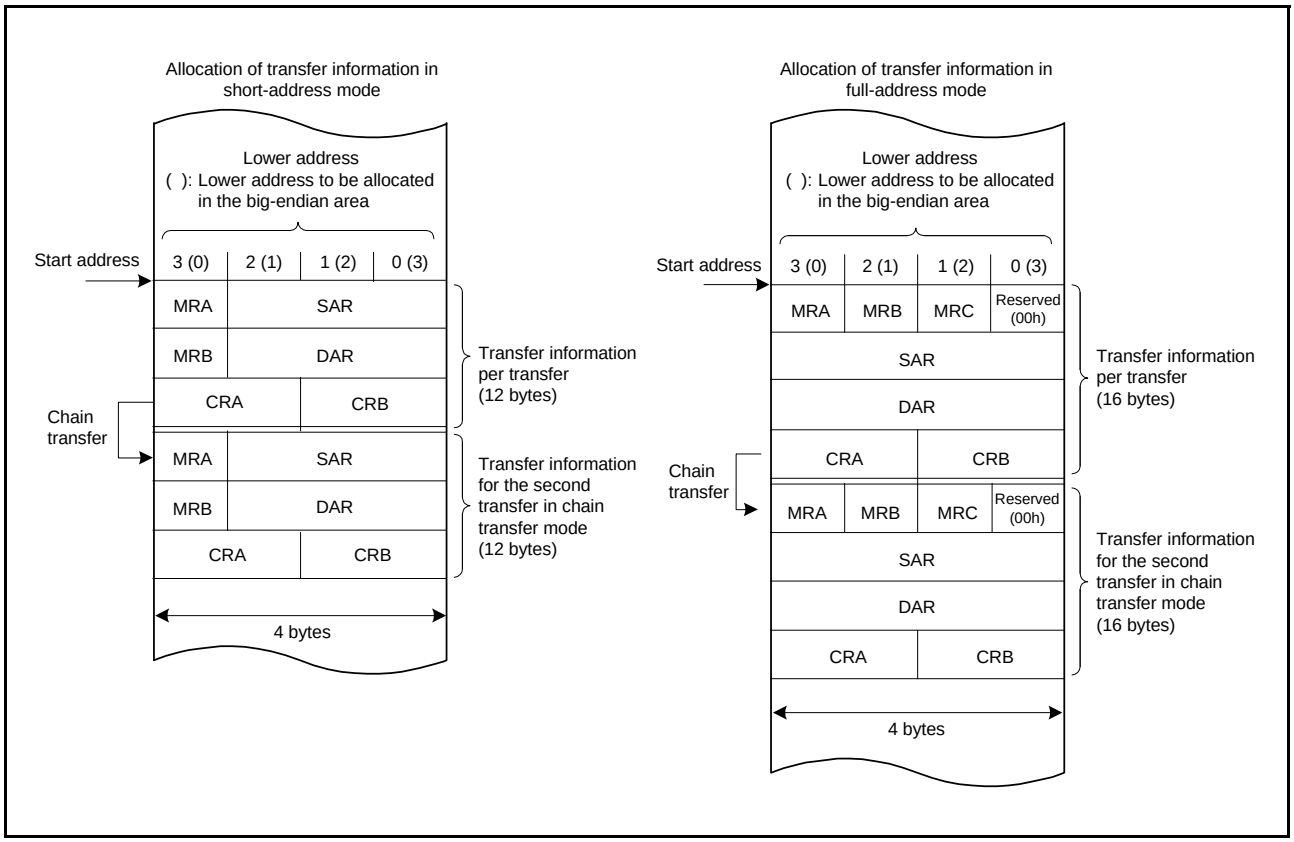


Figure 20.4 Allocation of Transfer Information in the RAM Area

20.4 Operation

The DTC transfers data in accordance with the transfer information. Storage of the transfer information in the RAM area is required before DTC operation.

When the DTC accepts a transfer request, it reads the DTC vector corresponding to the vector number. Next, the DTC reads transfer information from the address pointed by the DTC vector, transfers data, and then writes back the transfer information after the data transfer. Allocating transfer information in the RAM area allows data transfer of arbitrary number of channels.

There are three transfer modes: normal transfer mode, repeat transfer mode, and block transfer mode.

Set a transfer source address in the SAR register and a transfer destination address in the DAR register. The SAR and DAR registers are updated after the transfer according to the respective settings (increment, decrement, or fixed).

Table 20.3 lists transfer modes of the DTC.

Table 20.3 Transfer Modes of the DTC

Transfer Mode	Data Size Transferred on Single Transfer Request	Increment/Decrement of Memory Address	Settable Transfer Count
Normal transfer mode	1 byte/1 word/1 longword	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 65536
Repeat transfer mode*1	1 byte/1 word/1 longword	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 256*3
Block transfer mode*2	Block size specified in CRAH (1 to 256 bytes/1 to 256 words/1 to 256 longwords)	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 65536

Note 1. Set transfer source or transfer destination in the repeat area.
Note 2. Set transfer source or transfer destination in the block area.
Note 3. After data transfer of the specified count, the initial state is restored and the operation is continued (repeated).

Setting the MRB.CHNE bit to 1 allows multiple transfers (chain transfer) on a single transfer request. The setting in combination with the MRB.CHNS bit enables a chain transfer when the specified number of data transfers is completed. Figure 20.5 shows the operation flowchart of the DTC. Table 20.4 lists chain transfer conditions.

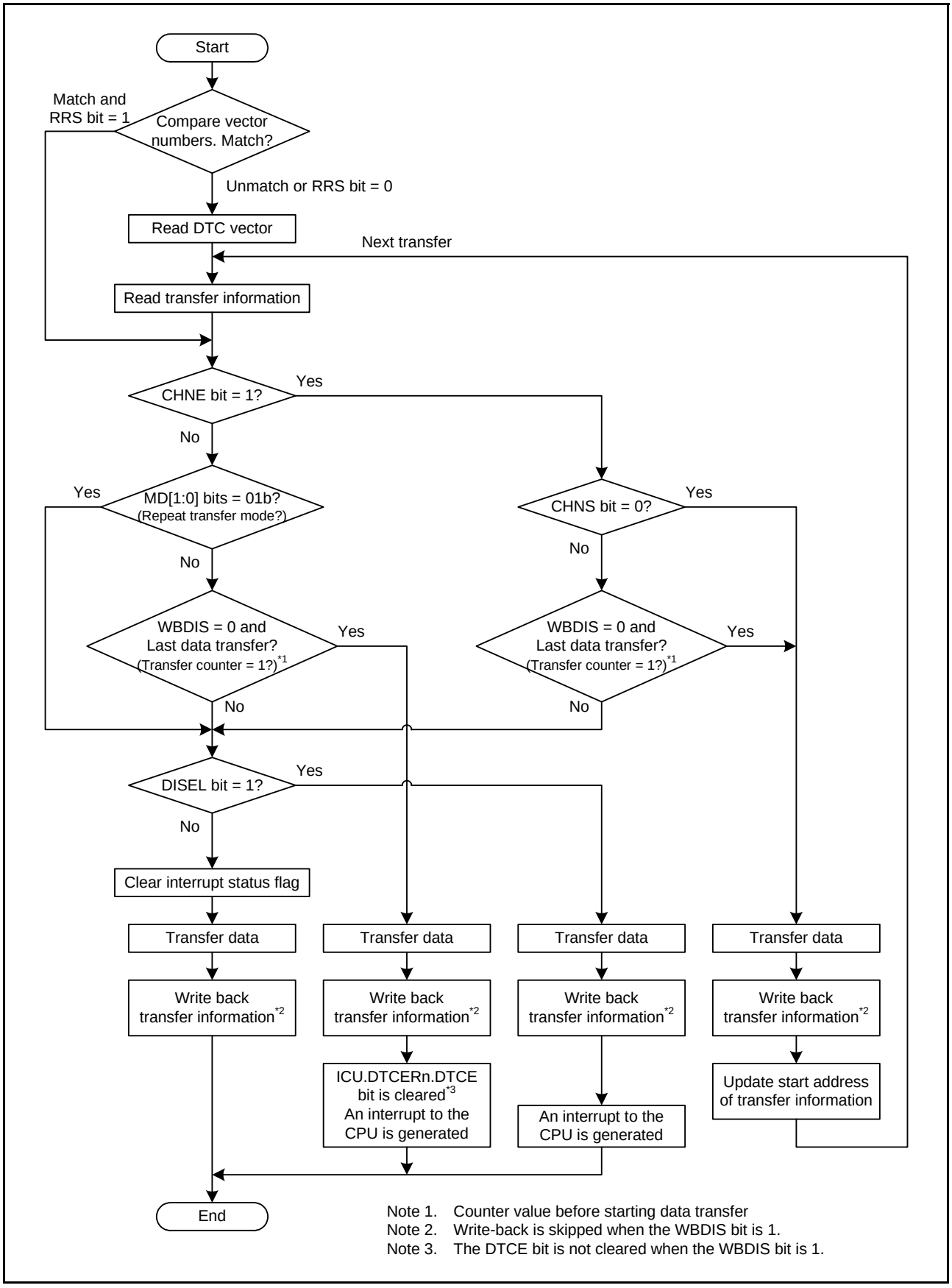


Figure 20.5 Operation Flowchart of the DTC

Table 20.4 Chain Transfer Conditions

First Transfer				Second Transfer*3				
CHNE Bit	CHNS Bit	DISEL Bit	Transfer Counter*1,*2	CHNE Bit	CHNS Bit	DISEL Bit	Transfer Counter*1,*2	Data Transfer
0	—	0	Other than (1 → 0)	—	—	—	—	Ends after the first transfer
0	—	0	(1 → 0)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU
0	—	1	—	—	—	—	—	
1	0	—	—	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	0	Other than (1 → ∗)	—	—	—	—	Ends after the first transfer
1	1	—	(1 → ∗)	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	1	Other than (1 → ∗)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU

Note 1. The transfer counters used depend on transfer modes as follows:
Normal transfer mode: CRA register
Repeat transfer mode: CRAL register
Block transfer mode: CRB register

Note 2. On completion of data transfer, the counters operate as follows:
1 → 0 in normal and block transfer modes
1 → CRAH in repeat transfer mode
(1 → ∗) in the table indicates both of the two operations above.

Note 3. Chain transfer can be selected for the second or subsequent transfers. The condition combination of “second transfer and the CHNE bit is 1” is omitted.

20.4.1 Transfer Information Read Skip Function

Reading of DTC vector and transfer information can be skipped by the setting of the DTCCR.RRS bit.

When a DTC transfer request is accepted, the current DTC vector number is compared with the DTC vector number in the previous data transfer. When these vector numbers match and the RRS bit is 1, the DTC does not read the DTC vector and transfer information, and transfers data according to the transfer information remained in the DTC.

However, when the previous transfer was chain transfer, the DTC vector and transfer information are read. Furthermore, when the transfer counter (CRA register) became 0 during the previous normal transfer and when the transfer counter (CRB register) became 0 during the previous block transfer, transfer information is read regardless of the value of the RRS bit. Figure 20.14 shows an example of transfer information read skip.

When updating the vector table and transfer information, set the RRS bit to 0, update the vector table and transfer information, and then set the RRS bit to 1. Setting the RRS bit to 0 discards the vector numbers retained in the DTC. The updated DTC vector table and transfer information are read in the next data transfer.

20.4.2 Transfer Information Write-Back Skip Function

20.4.2.1 Write-Back Skip by Fixing Addresses

When the MRA.SM[1:0] bits or the MRB.DM[1:0] bits are set to “address is fixed” (00b or 01b), a part of transfer information is not written back. This function is performed independently of the setting of short-address mode or full-address mode.

Table 20.5 lists transfer information write-back skip conditions and applicable registers. The CRA and CRB registers are written back independently of the setting of short-address mode or full-address mode.

Furthermore, in full-address mode, write-back of registers MRA, MRB, and MRC is skipped.

Table 20.5 Transfer Information Write-Back Skip Conditions and Applicable Registers

MRA.SM[1:0] Bits		MRB.DM[1:0] Bits			
b3	b2	b3	b2	**SAR Register**	**DAR Register**
0	0	0	0	Skip	Skip
0	0	0	1		
0	1	0	0		
0	1	0	1		
0	0	1	0	Skip	Write-back
0	0	1	1		
0	1	1	0		
0	1	1	1		
1	0	0	0	Write-back	Skip
1	0	0	1		
1	1	0	0		
1	1	0	1		
1	0	1	0	Write-back	Write-back
1	0	1	1		
1	1	1	0		
1	1	1	1		

20.4.2.2 Write-Back Skip by the MRA.WBDIS Bit

When the MRA.WBDIS bit is 1, the transfer information (SAR, DAR, CRA, and CRB) is not written back regardless of the settings of the transfer information.

The transfer information on the memory is not updated, data can be transferred by the DTC without copying the transfer information from ROM to RAM. Skipping a write-back reduces time for post-processing of the data transfer.

20.4.3 Normal Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single transfer request. The transfer count can be set to 1 to 65536.

Transfer source addresses and transfer destination addresses can be set to increment, decrement, or fixed independently.

This mode enables an interrupt request to the CPU to be generated at the end of specified-count transfer.

Table 20.6 lists register functions in normal transfer mode, and Figure 20.6 shows the memory map of normal transfer mode.

Table 20.6 Register Functions in Normal Transfer Mode

Register	Description	Value Written Back by Writing Transfer Information*1
SAR	Transfer source address	Increment/decrement/fixed*2
DAR	Transfer destination address	Increment/decrement/fixed*2
CRA	Transfer counter A	CRA – 1
CRB	Transfer counter B	Not updated

Note 1. Write-back operation is skipped when the MRA.WBDIS bit is 1.
Note 2. Write-back operation is skipped when address is fixed.

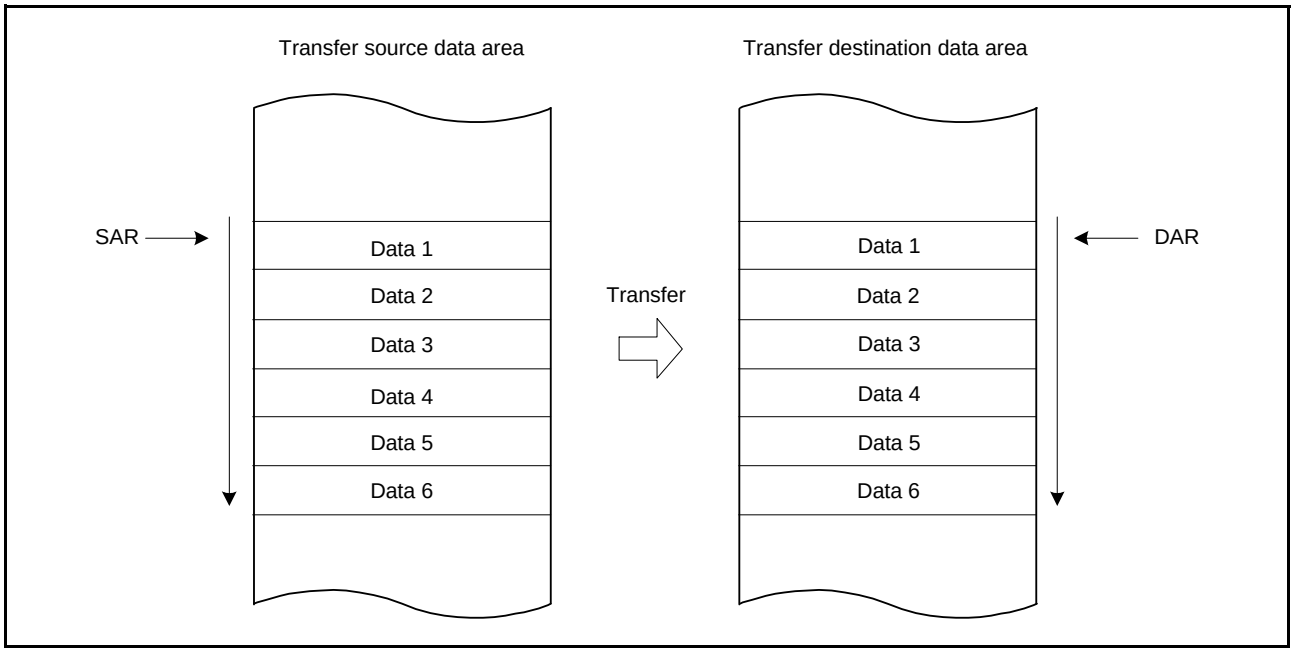


Figure 20.6 Memory Map of Normal Transfer Mode

20.4.4 Repeat Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single transfer request.
Specify either transfer source or transfer destination for the repeat area by the MRB.DTS bit. The transfer count can be set to 1 to 256. When the specified-count transfer is completed, the initial value of the address register specified in the transfer counter and the repeat area is restored and transfer is repeated. The other address register is incremented or decremented continuously or remains unchanged.
When the transfer counter CRAL is decreased to 00h in repeat transfer mode, the CRAL value is updated to the value set in the CRAH register. Thus the transfer counter does not become 00h, which disables an interrupt request to be generated to the CPU when the MRB.DISEL bit is set to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers).
Table 20.7 lists the register functions in repeat transfer mode, and Figure 20.7 shows the memory map of repeat transfer mode.

Table 20.7 Register Functions in Repeat Transfer Mode

		Value Written Back by Writing Transfer Information*1		
			When CRAL = 1	
Register	**Description**	**When CRAL ≠ 1**	**When the MRB.DTS Bit is 0**	**When the MRB.DTS Bit is 1**
SAR	Transfer source address	Increment/decrement/fixed*2	Increment/decrement/fixed*2	SAR register initial value
DAR	Transfer destination address	Increment/decrement/fixed*2	DAR register initial value	Increment/decrement/fixed*2
CRAH	Retains initial value of transfer counter	CRAH	CRAH	
CRAL	Transfer counter A	CRAL – 1	CRAH	
CRB	Transfer counter B	Not updated	Not updated	

Note 1. Write-back operation is skipped when the MRA.WBDIS bit is 1.
Note 2. Write-back operation is skipped when address is fixed.

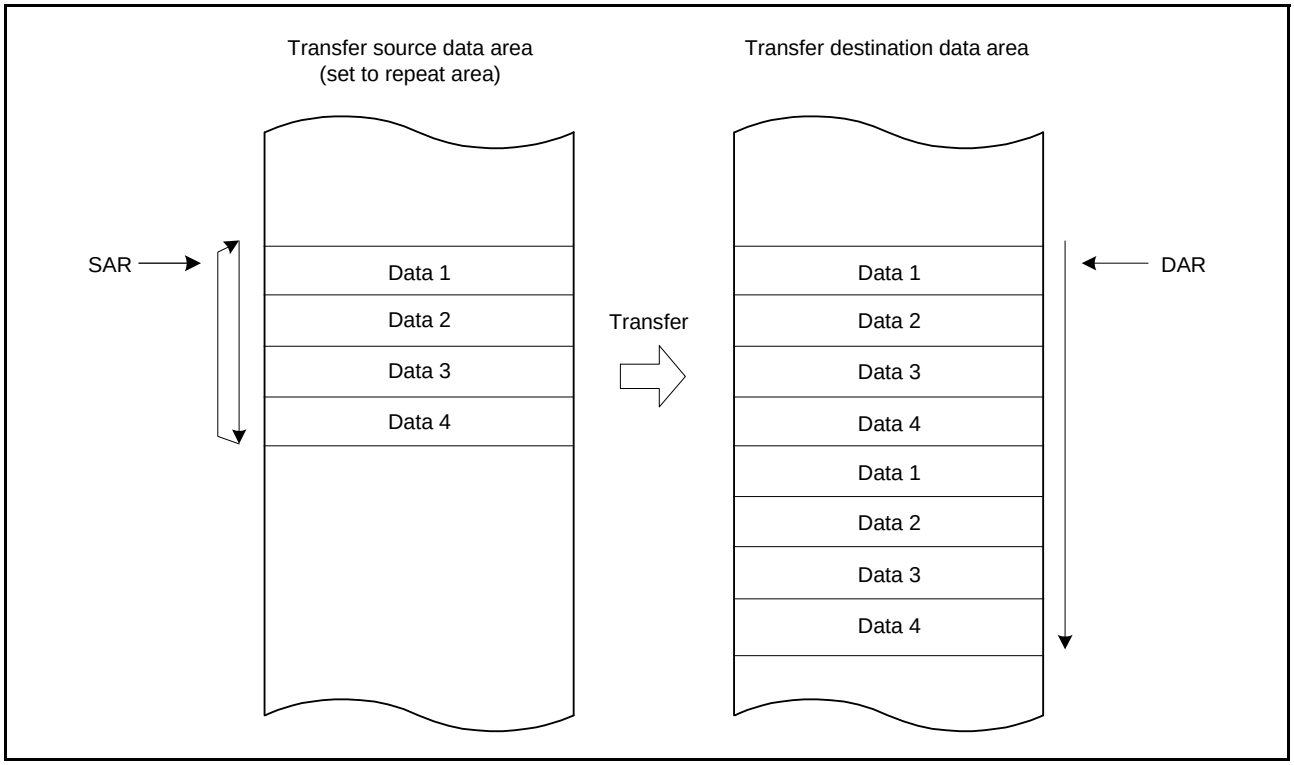


Figure 20.7 Memory Map of Repeat Transfer Mode (Transfer Source: Repeat Area)

20.4.5 Block Transfer Mode

This mode allows single-block data transfer on a single transfer request.
Specify either transfer source or transfer destination for the block area by the MRB.DTS bit. The block size can be set to 1 to 256 bytes, 1 to 256 words, or 1 to 256 longwords.
When transfer of the specified one block is completed, the initial values of the block size counter CRAL and the address register (the SAR register when the MRB.DTS bit is 1 or the DAR register when the DTS bit is 0) specified in the block area are restored. The other address register is incremented or decremented continuously or remains unchanged.
The transfer count (block count) can be set to 1 to 65536. This mode enables an interrupt request to the CPU to be generated at the end of specified-count block transfer.
Table 20.8 lists register functions in block transfer mode, and Figure 20.8 shows the memory map of block transfer mode.

Table 20.8 Register Functions in Block Transfer Mode

		Value Written Back by Writing Transfer Information*1	
Register	**Description**	**When MRB.DTS Bit is 0**	**When MRB.DTS Bit is 1**
SAR	Transfer source address	Increment/decrement/fixed*2	SAR register initial value
DAR	Transfer destination address	DAR register initial value	Increment/decrement/fixed*2
CRAH	Retains initial value of block size	CRAH	
CRAL	Block size counter	CRAH	
CRB	Block transfer counter	CRB – 1	

Note 1. Write-back operation is skipped when the MRA.WBDIS bit is 1.
Note 2. Write-back operation is skipped when address is fixed.

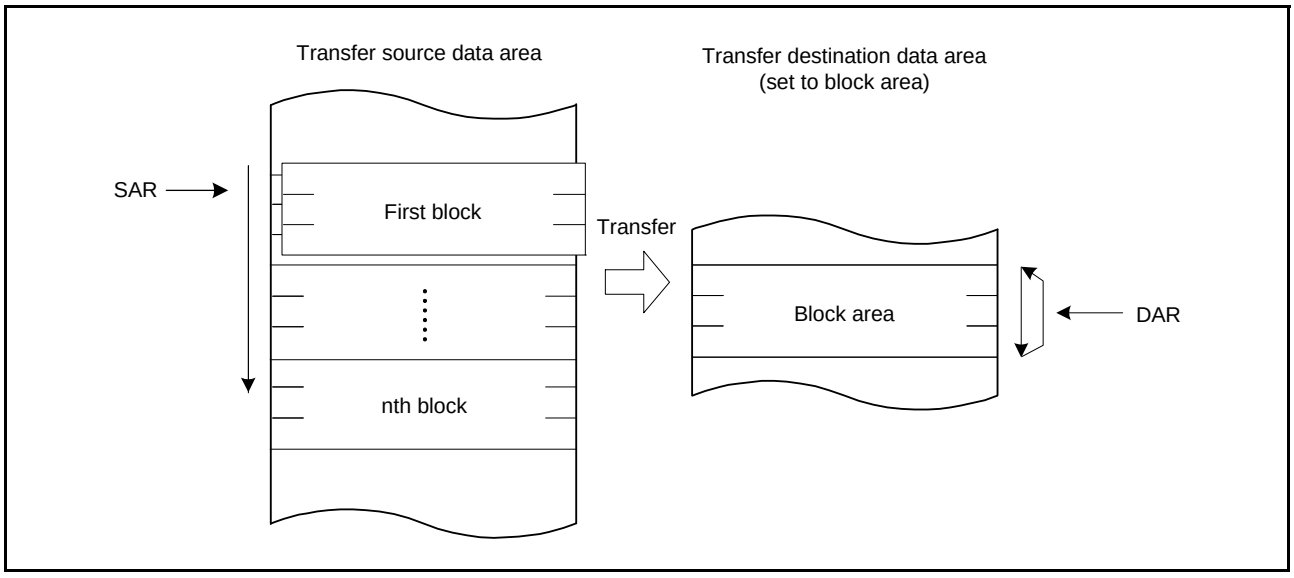


Figure 20.8 Memory Map of Block Transfer Mode (Transfer Destination: Block Area)

20.4.6 Chain Transfer

Setting the MRB.CHNE bit to 1 allows chain transfer to be performed continuously on a single transfer request.
If the MRB.CHNE bit is 1 and the MRB.CHNS bit is 0, an interrupt request to the CPU is not generated when the specified number of data transfers is completed, or while the MRB.DISEL bit is 1 (an interrupt request to the CPU is generated for every data transfer). Data transfer has no effect on the interrupt status flag, which is the request source.
The transfer information (SAR, DAR, CRA, CRB, MRA, MRB, and MRC) that define a data transfer can be specified independently of each other. Figure 20.9 shows chain transfer operation.

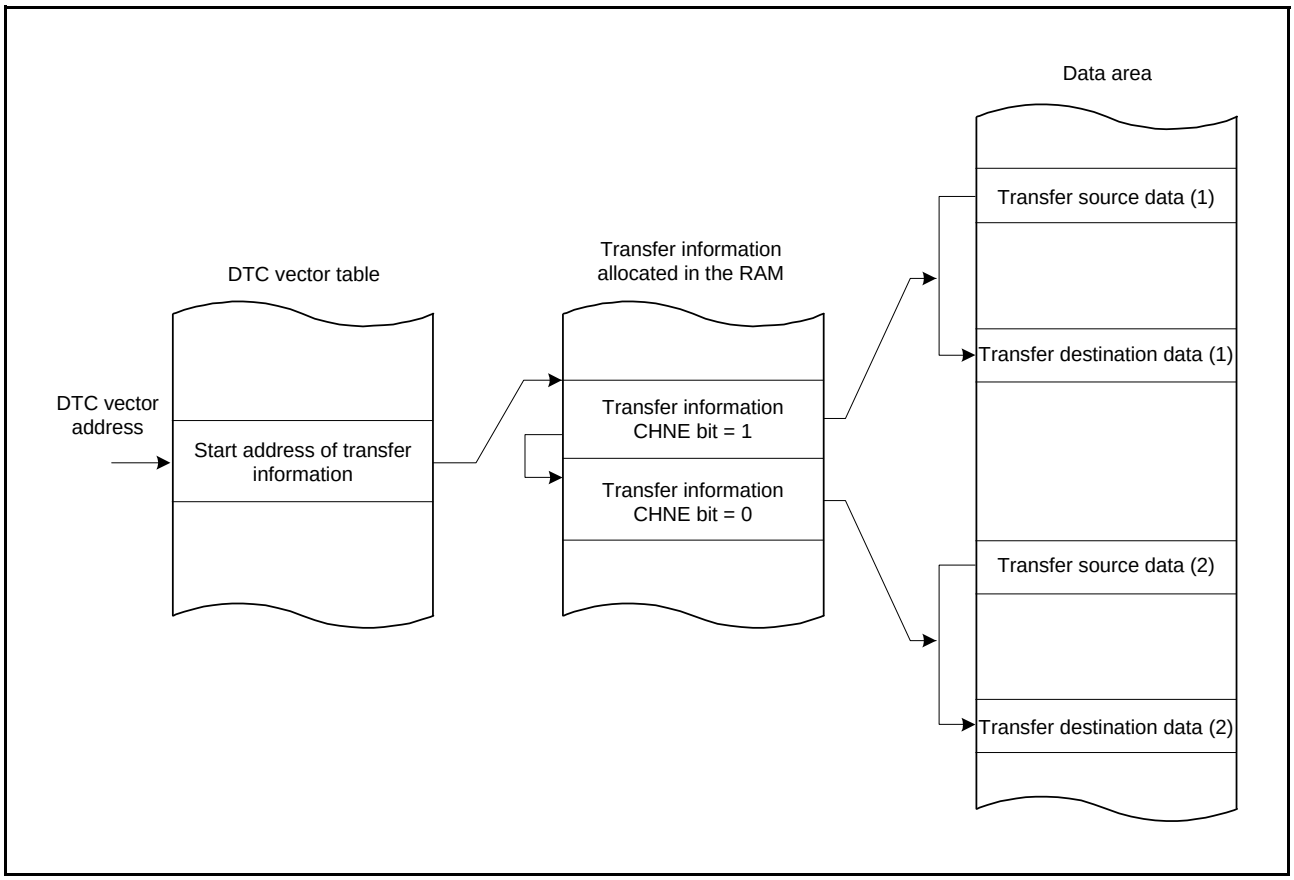


Figure 20.9 Chain Transfer Operation

If the MRB.CHNE bit is 1 and the CHNS bit is 1, chain transfer is performed only after completion of specified number of data transfers. In repeat transfer mode, chain transfer is performed after completion of specified number of data transfers.
For details on chain transfer conditions, refer to Table 20.4, Chain Transfer Conditions.

20.4.7 Operation Timing

Figure 20.10 to Figure 20.14 show examples of DTC operation timing.

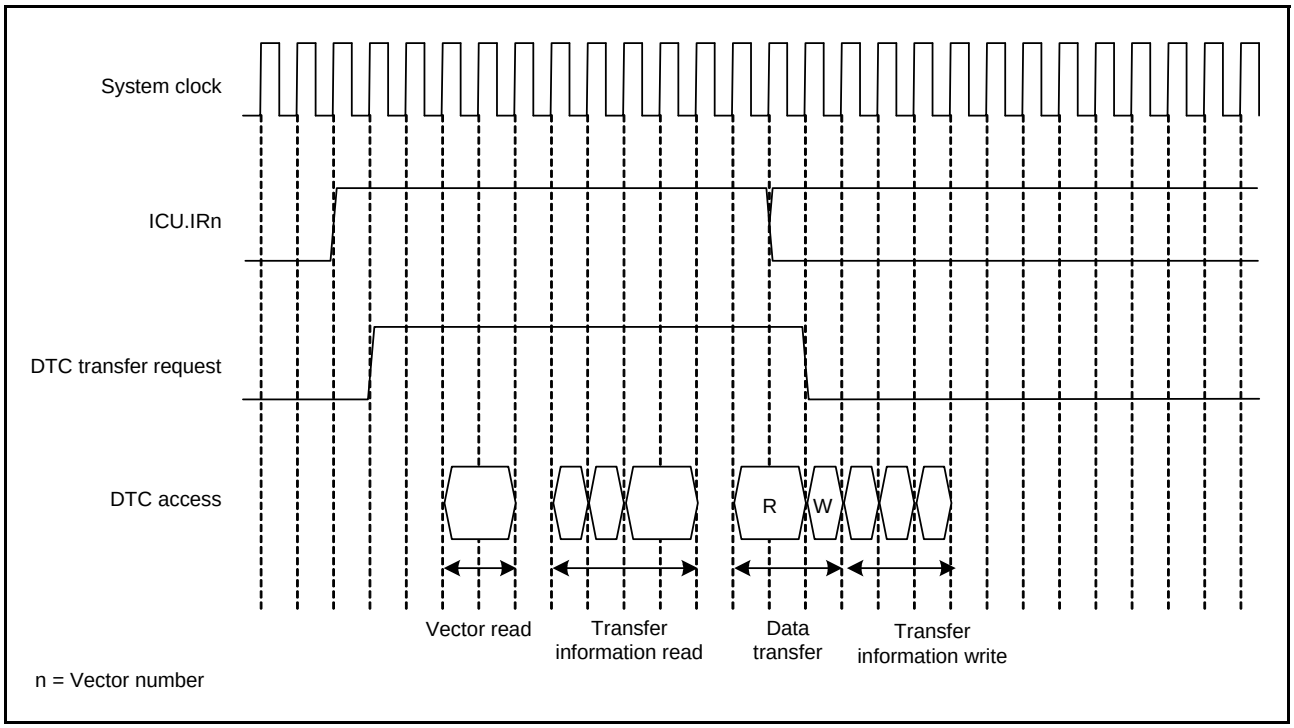


Figure 20.10 Example (1) of DTC Operation Timing (Short-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)

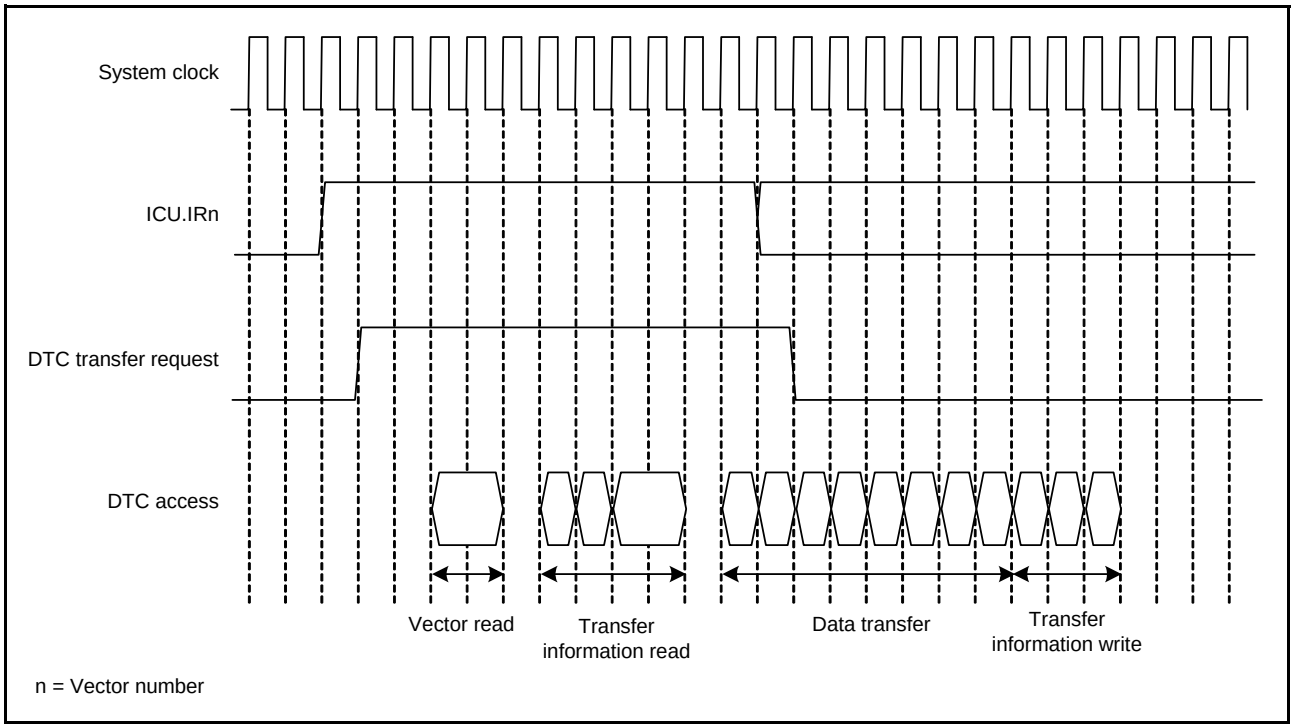


Figure 20.11 Example (2) of DTC Operation Timing (Short-Address Mode, Block Transfer Mode, Block Size = 4)

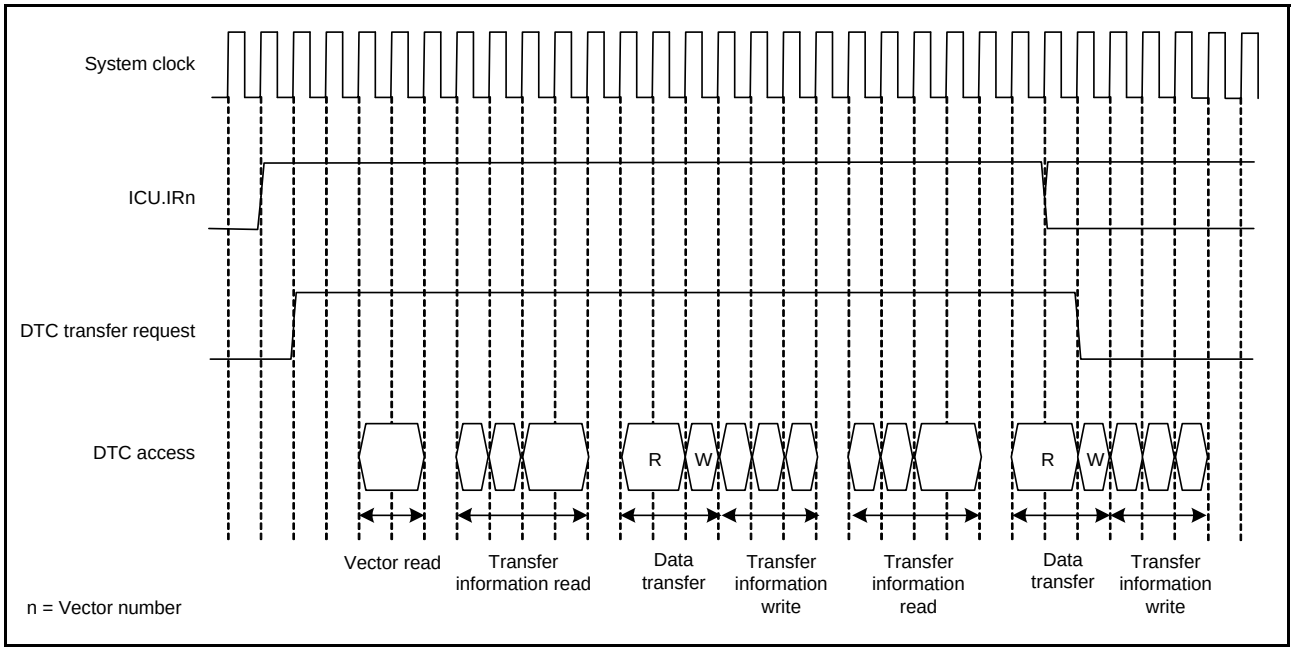


Figure 20.12 Example (3) of DTC Operation Timing (Short-Address Mode, Chain Transfer)

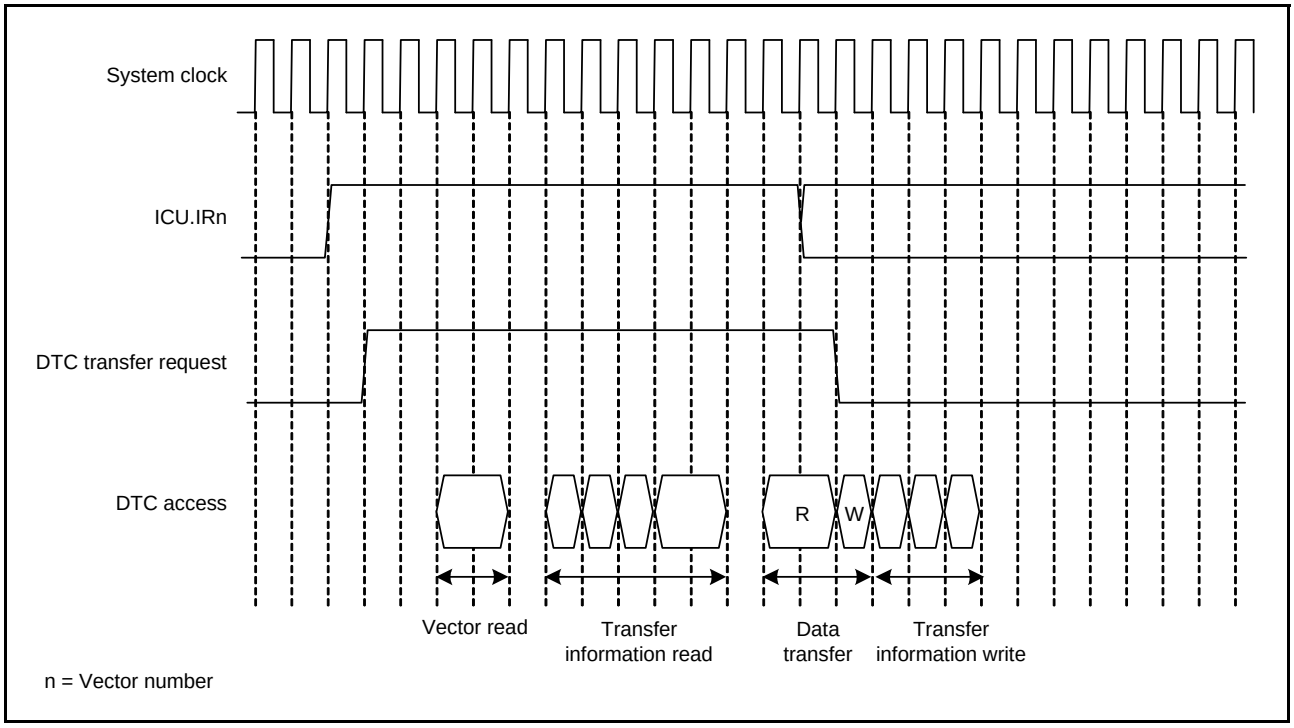
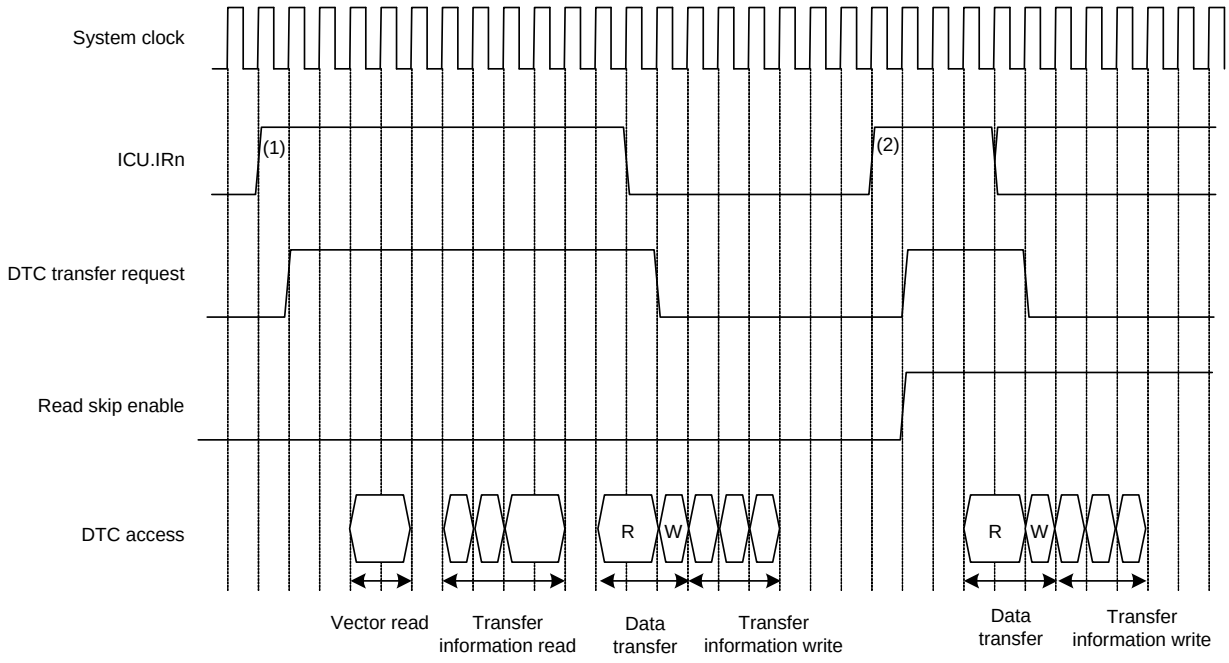


Figure 20.13 Example (4) of DTC Operation Timing (Full-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)



n = Vector number

Note: When request sources (vector numbers) of (1) and (2) are the same and the DTCCR.RRS bit is 1, the transfer information read for request (2) is skipped.

Figure 20.14 Example of Operation When Transfer Information Read Skip is Executed (Vector, Transfer Information, and Transfer Destination Data on the RAM, and Transfer Source Data on the Peripheral Module)

20.4.8 Execution Cycles of the DTC

Table 20.9 lists the execution cycles of single data transfer of the DTC.
For the order of the execution states, refer to section 20.4.7, Operation Timing.

Table 20.9 Execution Cycles of the DTC

Transfer Mode	Vector Read		Transfer Information Read			Transfer Information Write			Data Transfer		Internal Operation	
									Read	Write		
Normal	Cv + 1	0*1	4 × Ci + 1*2	3 × Ci + 1*3	0*1	3 × Ci*4	2 × Ci*5	Ci*6	Cr + 1	Cw	2	0*1
Repeat									Cr + 1	Cw		
Block*7									P × Cr	P × Cw		

Note 1. When transfer information read is skipped
Note 2. In full-address mode
Note 3. In short-address mode
Note 4. When neither SAR nor DAR is set to address-fixed
Note 5. When SAR or DAR is set to address-fixed
Note 6. When SAR and DAR are set to address-fixed
Note 7. When the block size is 2 or more. If the block size is 1, the cycle number for normal transfer is applied.

P: Block size (initial settings of CRAH and CRAL)
Cv: Cycles for access to vector transfer information storage destination
Ci: Cycles for access to transfer information storage destination address
Cr: Cycles for access to data read destination
Cw: Cycles for access to data write destination
(The unit is system clocks (ICLK) for “+ 1” in the Vector Read, Transfer Information Read, and Data Transfer Read columns and “2” in the Internal Operation column.)
(Cv, Ci, Cr, and Cw vary depending on the corresponding access destination. For the number of cycles for respective access destinations, refer to section 53, RAM, section 55, Flash Memory (FLASH), section 5, I/O Registers, and section 16.2.7, External Bus.)

20.4.9 DTC Bus Mastership Release Timing

The DTC does not release the bus mastership during transfer information read and transfer information write. While transfer information is not read or written, bus arbitration is made according to the priority determined by the bus master arbitrator.
For bus arbitration, refer to section 16, Buses.

20.4.10 Sequence Transfer

A sequence transfer can be executed on the request source that is specified in the DTCSQE register. A sequence transfer is started by setting the MRB.INDX bit to 1 and ended by setting the MRB.SQEND bit to 1. Setting the DTCOR.SQTFRL bit to 1, even during a sequence transfer, forcibly ends the transfer and the next DTC transfer request starts new sequence transfer with reference to the index table.
A sequence transfer includes the following processing.

(1) The first data transfer is executed by referring to the DTC vector table in response to a DTC transfer request from the source specified in the DTCSQE register.
(2) The DTC index table is referred based on the value of the lower 8 bits of the first data transferred data in (1) (sequence number).
(3) The transfer information is read from the address obtained from the DTC index table.
(4) A data transfer is executed in accordance with the transfer information. On completion of the data transfer, either one of the following operations is performed by using the values of bits MRB.CHNE and MRB.SQEND.
- A chain transfer is executed when the CHNE bit is 1. The next transfer information is read. Go to (4).
- The sequence transfer is suspended when the CHNE bit is 0 and the SQEND bit is 0. Go to (5).
- The sequence transfer ends when the CHNE bit is 0 and the SQEND bit is 1.

(5) When a DTC transfer request from the source specified in the DTCSQE register is accepted, the suspended sequence transfer is resumed and the next transfer information is read. Go to (4).

Note 1. When the ICU.DTCERn.DTCE bit becomes 0 based on the result of the data transfer, a DTC transfer request is not generated. Set the DTCE bit to 1 to resume sequence transfer. Refer to Figure 20.5 or section 15, Interrupt Controller (ICUE) for the conditions where the DTCE bit becomes 1.

Figure 20.15 and Figure 20.16 shows a basic sequence transfer operation.

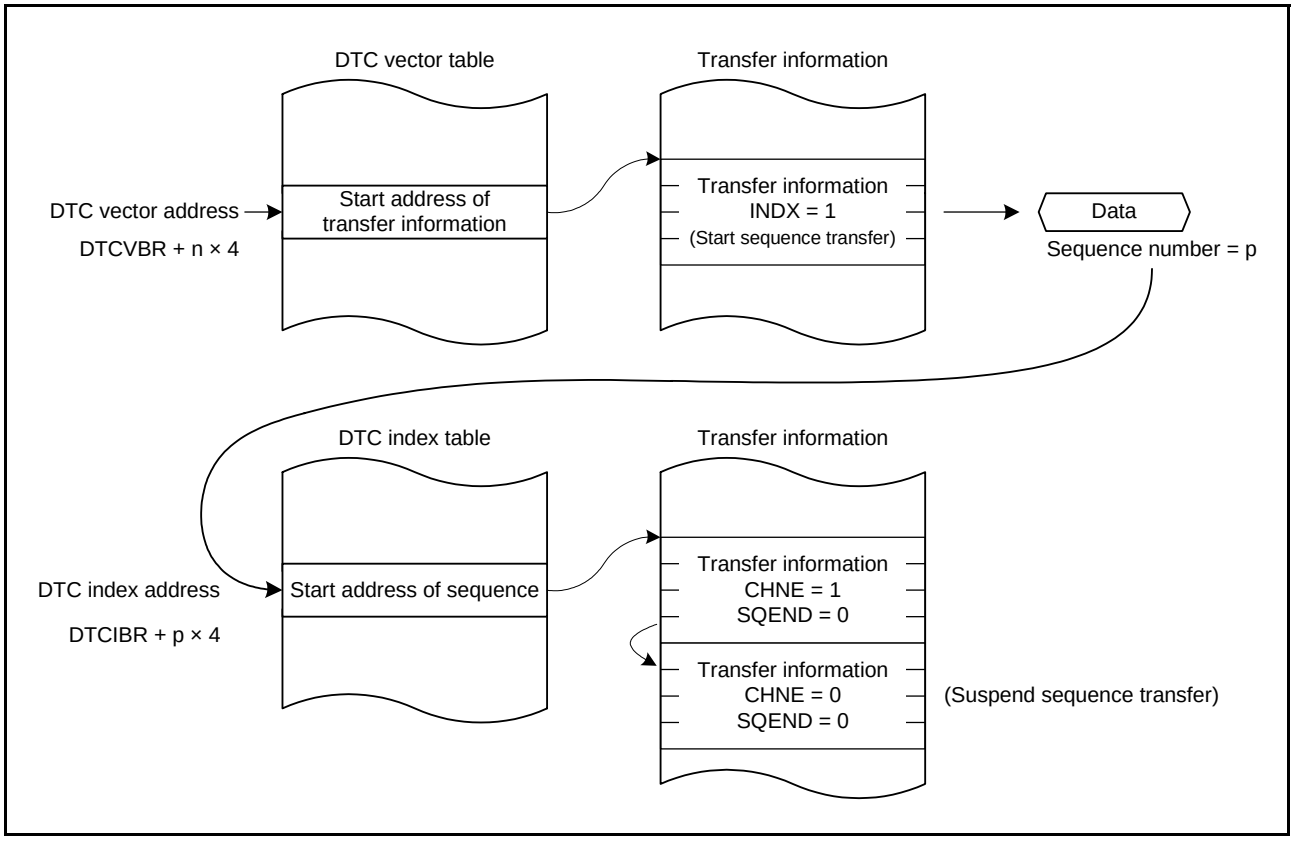


Figure 20.15 Start and Suspension of Sequence Transfer

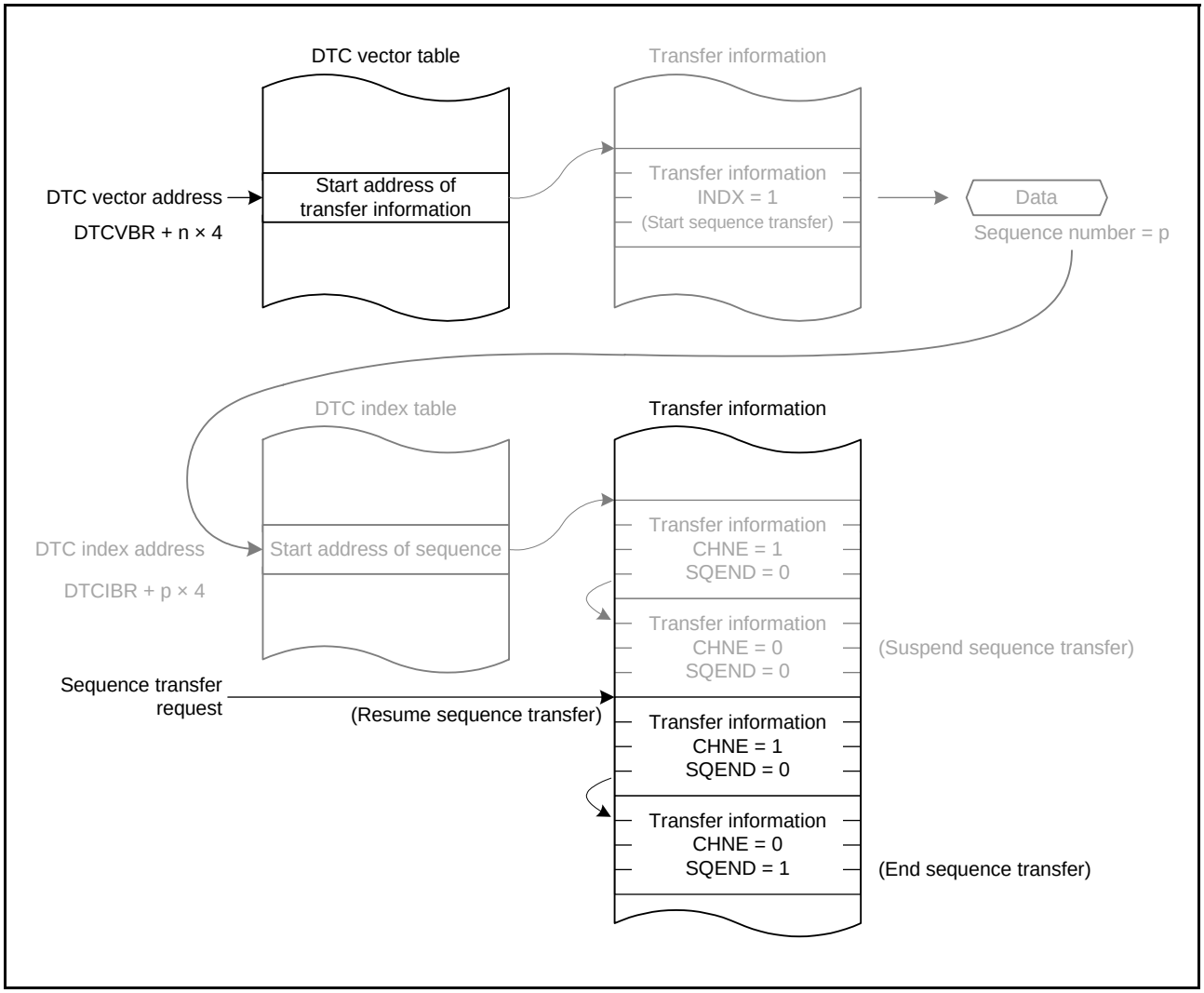


Figure 20.16 Resumption and End of Sequence Transfer

Table 20.10 lists the settings of bits CHNE, SQEND, and INDX during the sequence transfer.

Table 20.10 Sequence Transfer Process and Values of Bits CHNE, SQEND, and INDX

DTC Operations	CHNE Bit	SQEND Bit	INDX Bit
Start sequence transfer	0	0	1*1
Continue sequence transfer	1	0	0
Suspend sequence transfer*2	0	0	0
End sequence transfer	0	1	0
End current sequence transfer and Obtain new sequence number	0	1	1*1
Some other transfer (not sequence transfer)	—	0	0

Note: Do not set the values other than listed above.
Note 1. Set MRA.MD[1:0] bits to 00b (normal transfer mode) when setting the INDX bit to 1.
Note 2. When a sequence transfer is suspended, the ICU.DTCERn.DTCE bit may become 0. Set the DTCE bit to 1 to resume sequence transfer.

Even when a sequence transfer is suspended, a new sequence transfer cannot start until the suspended sequence transfer is eventually completed. When a sequence transfer request is received during suspension of the sequence transfer, the suspended sequence transfer is resumed.

20.4.11 DTC Index Table

The DTC index table is allocated to the area where its start address is configured in the DTCIBR register.
Store the start address of transfer information table p for sequence number p in the address of DTCIBR + p × 4.
The upper 30 bits of the start address is set to the upper 30 bits of the DTC index. Set the CPUSEL bit to select either of reading the transfer information and starting the sequence, or output an interrupt request to the CPU without starting the sequence. For a complicated sequence that the DTC cannot handle, set the CPUSEL bit to 1 to allow the CPU to handle such a sequence.

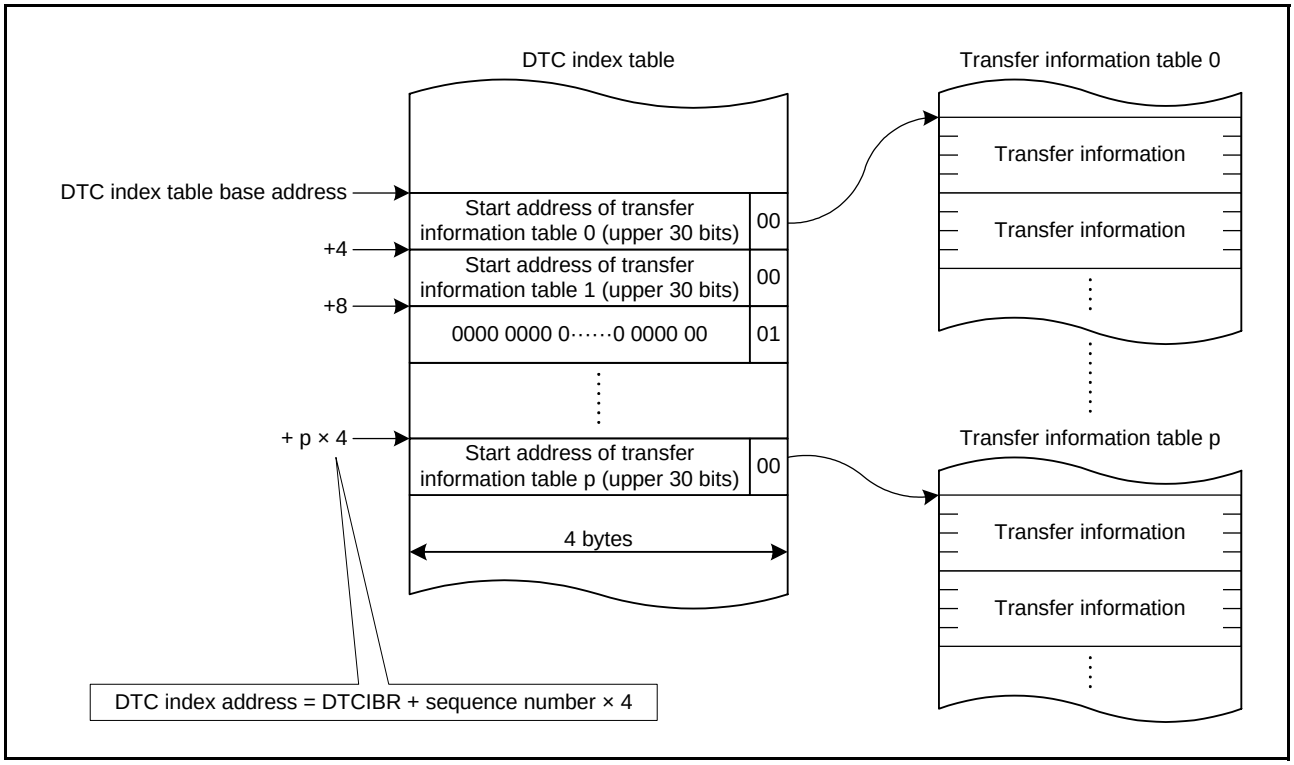
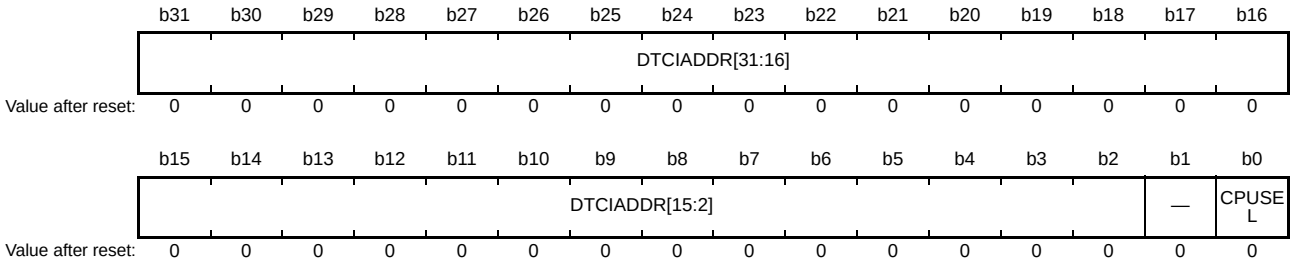


Figure 20.17 DTC Index Table

- DTC Index

Address(es): DTCIBR + p × 4



Bit	Symbol	Bit Name	Description	R/W
b0	CPUSEL	Sequence Transfer/CPU Interrupt Select	0: Continues the sequence transfer (starts the sequence) 1: Ends the sequence transfer and outputs an interrupt request to the CPU	—
b1	—	Reserved	Set this bit to 0.	—
b31 to b2	DTCIADDR[31:2]	Transfer Information Table Address	Set the upper 30 bits of the start address of the transfer information table to these bits. Writing to the upper 4 bits (b31 to b28) is ignored and the values in b31 to b28 become the same value as b27.	—

When the CPUSEL bit in the DTC index that the obtained sequence number indicates is 1, an interrupt request to the CPU is generated. At this time, the ICU.DTCERn.DTCE bit becomes 0. From this point, the interrupt request signal from the request source that is specified in the DTCSQE register is sent to the CPU, but not DTC. After completion of CPU interrupt processing, set the ICU.DTCERn.DTCE bit to 1 to enable DTC transfer request for starting the next sequence transfer.

20.4.12 Example of Sequence Transfer

Figure 20.18 shows a typical examples of a sequence transfer and Figure 20.19 to Figure 20.23 show configurations of the transfer information for the examples of the transfers in the figure.
In these examples, the interrupt source of vector number n is set as the source of the sequence transfer (DTCSQE.VECN[7:0] = n).
Once the DTC transfer request due to the interrupt source of vector number n (hereinafter referred to as “transfer request n”) is input, the DTC refers to the DTC vector table and reads the corresponding transfer information. The lower 8 bits that have been transferred based on the transfer information become a sequence number, selecting one sequence among possible 256 sequences.

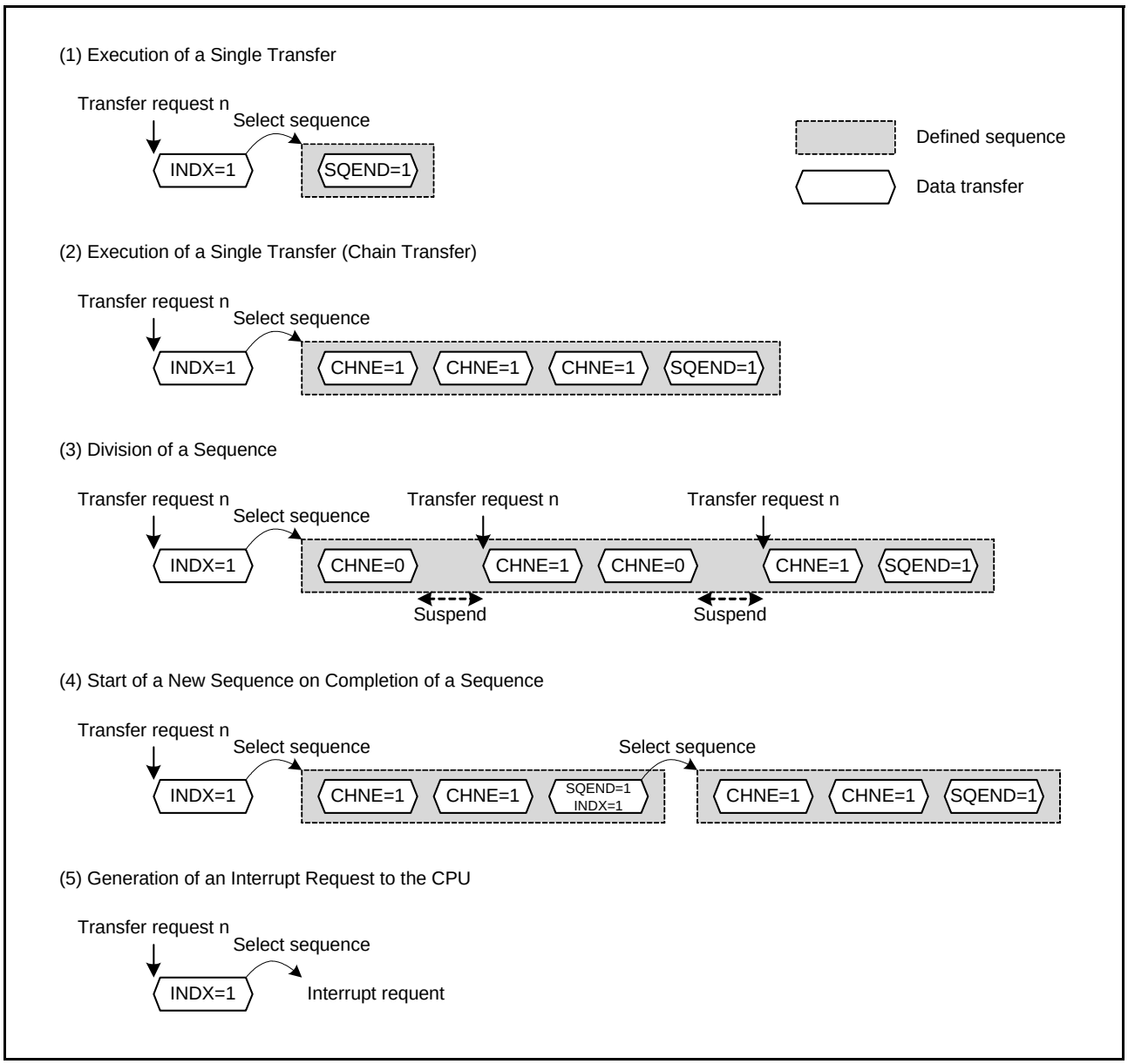


Figure 20.18 Examples of Sequence Transfers

(1) When Executing a Single Transfer

Figure 20.19 shows an example of a single transfer (normal, repeat, or block).
The DTC refers to the DTC index table, and reads the transfer information corresponding to the obtained sequence number p.
Since the values of the CHNE, INDX, and SQEND bits are 0, 0, and 1 respectively, the sequence ends after the specified transfer is executed.

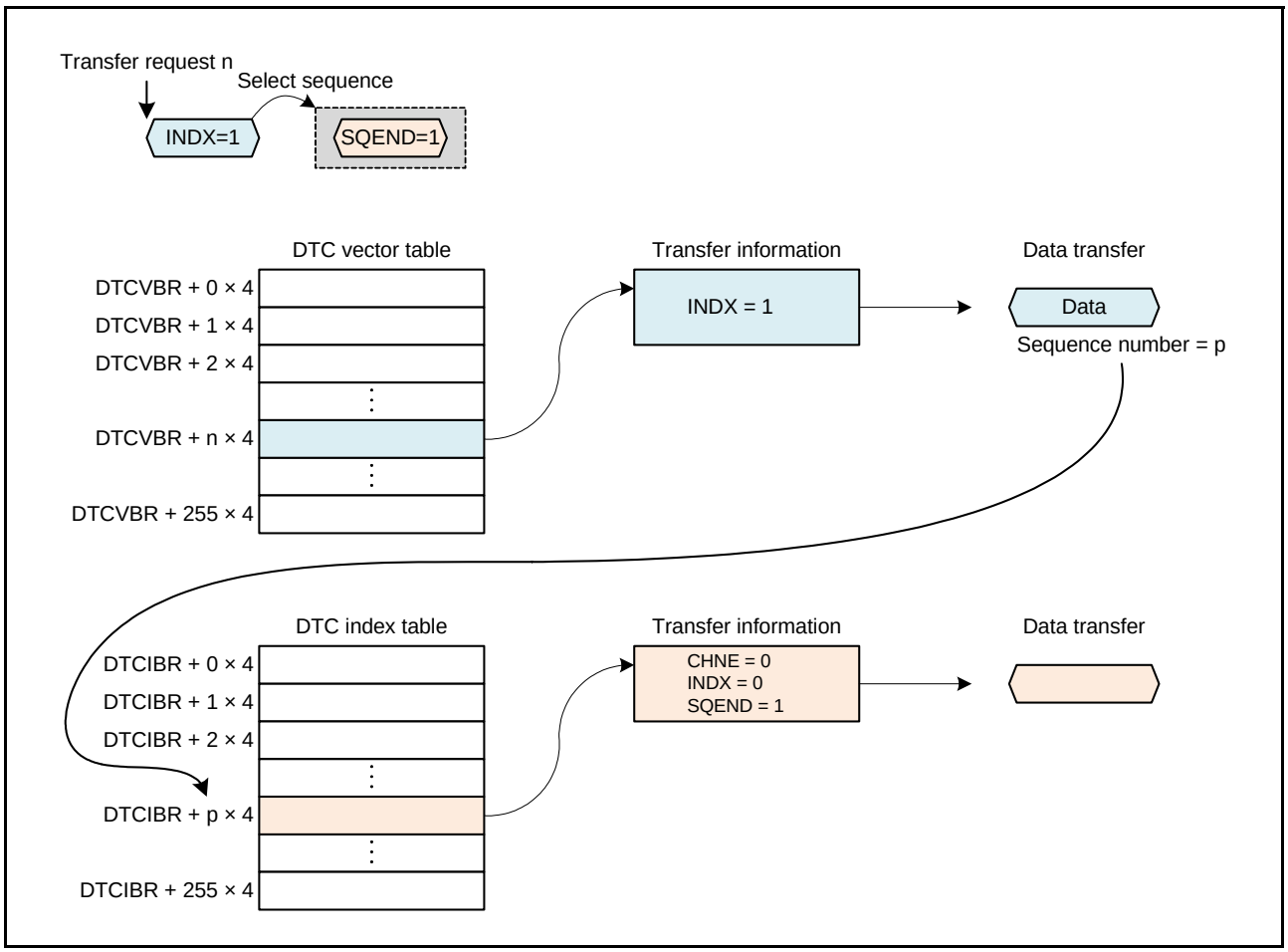


Figure 20.19 Example of a Sequence of Single Transfer

(2) When Executing a Single Chain Transfer

Figure 20.20 is an example of a sequence for a single chain transfer.

The DTC refers to the DTC index table, and reads the transfer information corresponding to the obtained sequence number q.

While the values of the CHNE, INDX, and SQEND bits are 1, 0, and 0 respectively, the specified chain transfer is executed. When the transfer information in which the CHNE, INDX, and SQEND bits are 0, 0, and 1 respectively is read, the sequence ends after the specified transfer is executed.

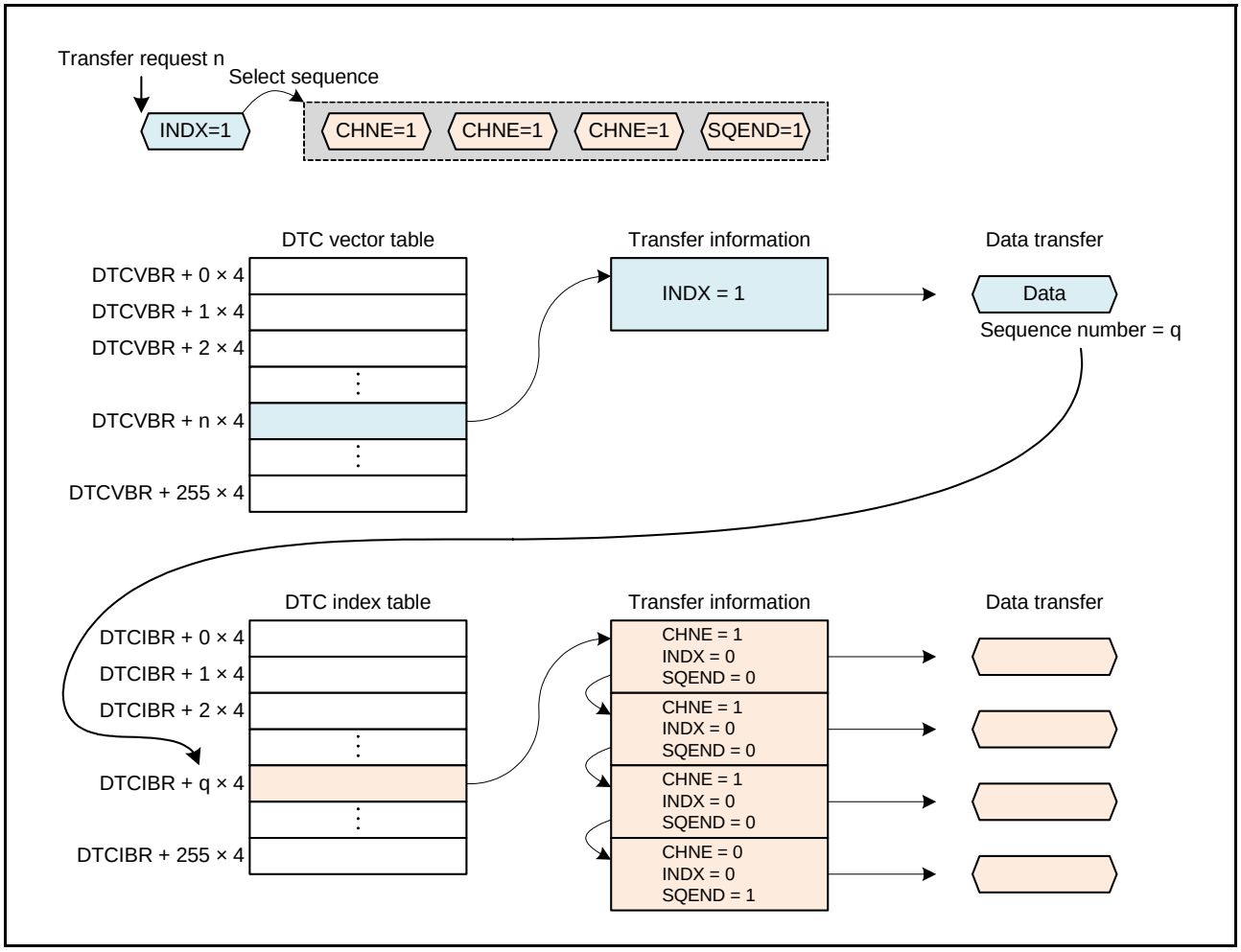


Figure 20.20 Example of Sequence of a Single Chain Transfer

(3) When Dividing a Sequence

Figure 20.21 is an example of the sequence that is divided into 3 parts.
The DTC refers to the DTC index table, and reads the transfer information corresponding to the obtained sequence number r.
Since the values of the CHNE, INDX, and SQEND bits in the transfer information are 0, 0, and 0 respectively, the sequence is suspended after the specified transfer is executed and the DTC waits for the next transfer request n.
When the transfer request n is input during a sequence transfer, the DTC vector table is not referred and the suspended sequence is resumed.
When the transfer information in which the CHNE, INDX, and SQEND bits are 0, 0, and 1 respectively is read, the sequence ends after the specified transfer is executed.

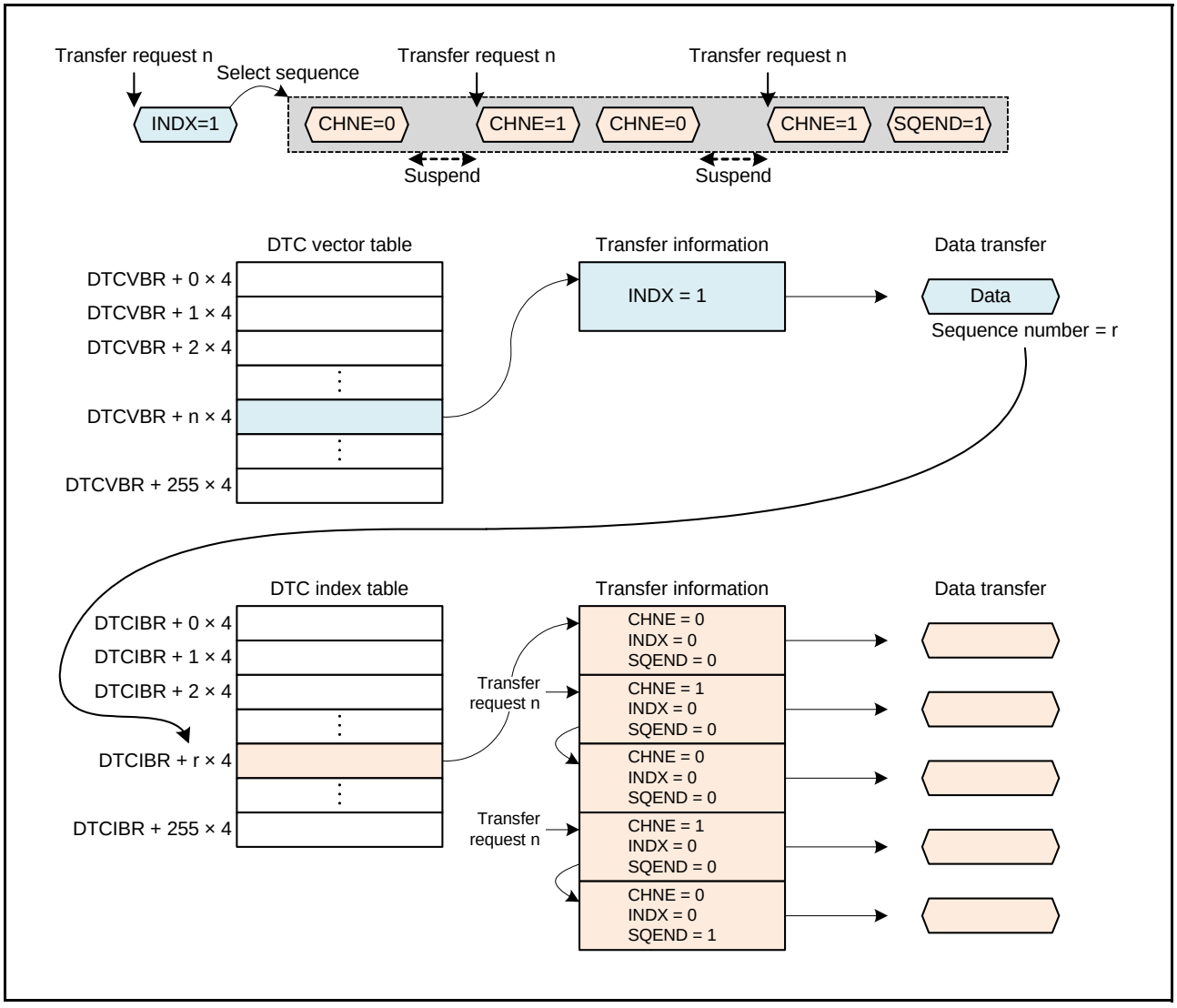


Figure 20.21 Example of Divided Sequence

(4) When Starting a New Sequence on completion of a Sequence

Figure 20.22 is an example for starting the next and new sequence on completion of the first sequence.
The DTC refers to the DTC index table, and reads the transfer information corresponding to the obtained sequence number s.
When the transfer information in which the CHNE, INDX, and SQEND bits are 0, 1, and 1 respectively is read, the specified transfer is executed, then a new sequence number is obtained from the lower 8 bits of the transferred data.
The DTC again refers to the DTC index table, and reads the transfer information corresponding to the obtained sequence number k and then starts a new sequence.
When the transfer information in which the CHNE, INDX, and SQEND bits are 0, 0, and 1 respectively is read, the sequence ends after the specified transfer is executed.

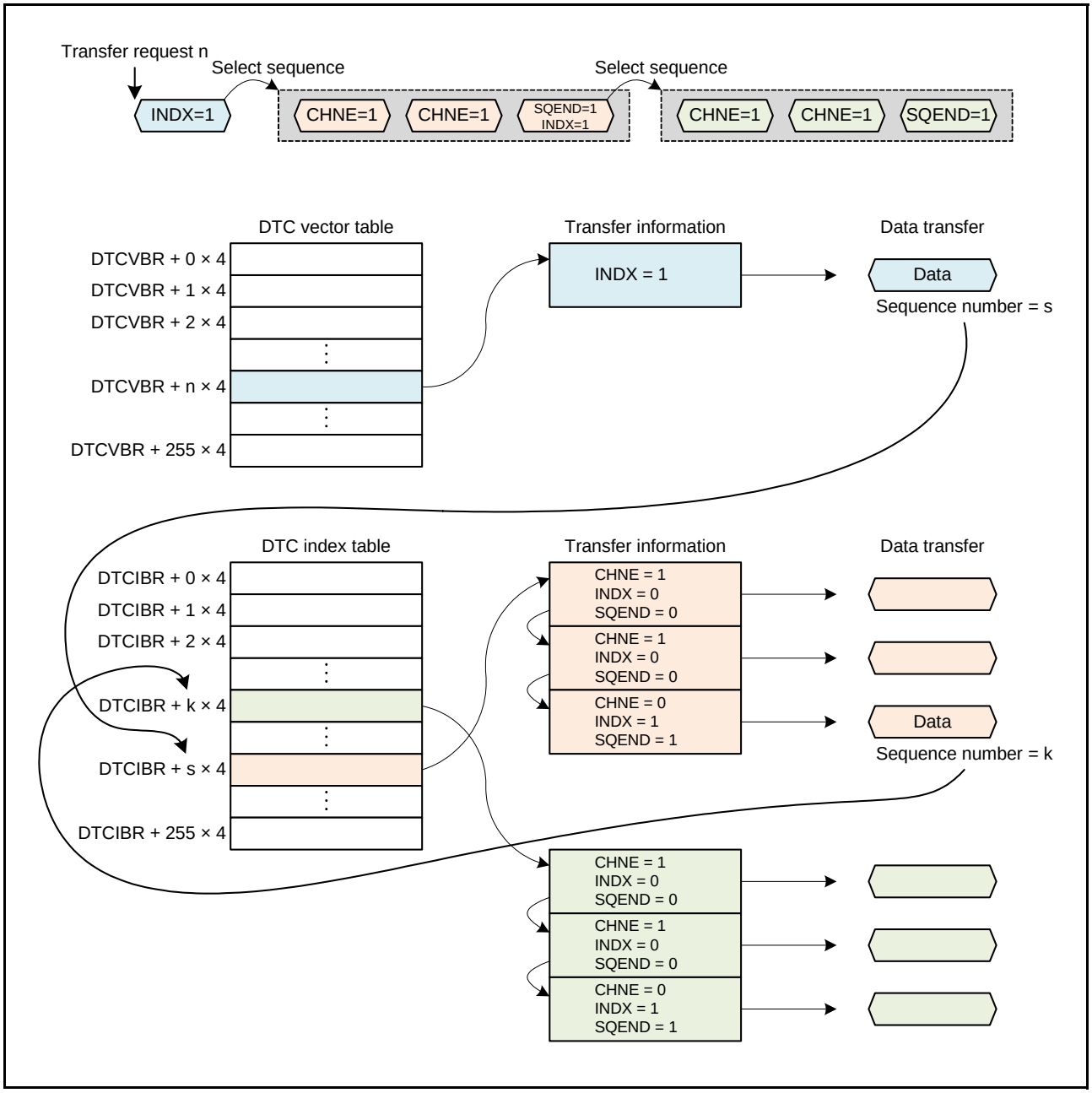


Figure 20.22 Example When Starting a New Sequence on Completion of a Sequence

(5) When Generating an Interrupt Request to the CPU

Figure 20.23 is an example of that an interrupt request is output to the CPU without starting of sequence.
The DTC obtains a DTC index that corresponds to the obtained sequence number t.
When the CPUSEL bit of the obtained DTC index is 1, the DTC ends the sequence transfer without starting the sequence, and then outputs an interrupt request to the CPU.

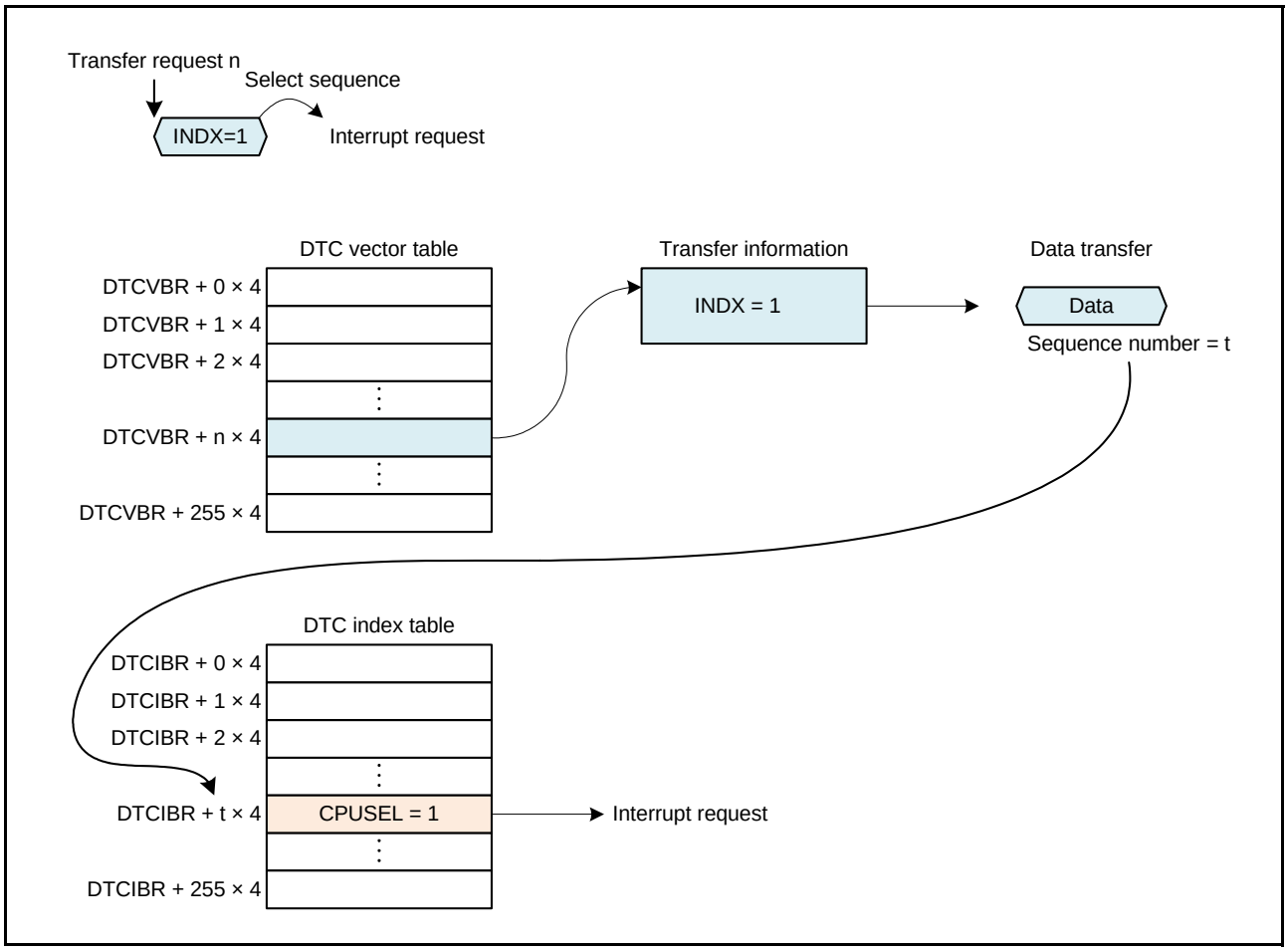
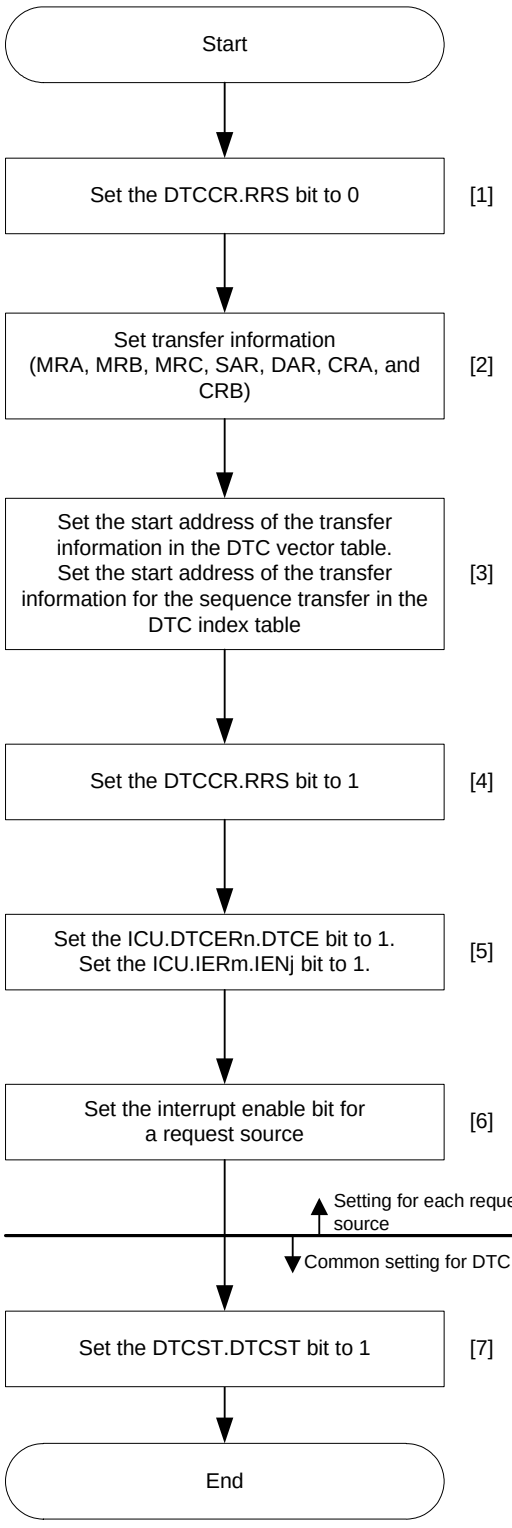


Figure 20.23 Example of Output of an Interrupt Request to the CPU

20.5 DTC Setting Procedure

Before using the DTC, set the DTC vector base register (DTCVBR). When using sequence transfer, also set the DTC index table base register (DTCIBR).

Figure 20.24 shows the procedure to set the DTC.



Set the ICU.IERm.IENj bit corresponding to the request source interrupt to 0 and provide the following settings.

[1] Setting the DTCCR.RRS bit to 0 resets the transfer information read skip flag. After that, transfer information read is not skipped during the data transfer. When transfer information is updated, be sure to make this setting.

[2] Allocate transfer information (MRA, MRB, MRC, SAR, DAR, CRA, and CRB) in the data area. For setting transfer information, refer to section 20.2, Register Descriptions. For how to allocate transfer information, refer to section 20.3.1, Allocating Transfer Information and DTC Vector Table.

[3] Set the start address of the transfer information in the DTC vector table. For how to set the DTC vector table, refer to section 20.3.1, Allocating Transfer Information and DTC Vector Table. When using sequence transfer, set the DTCADMOD.SHORT bit to 0, set registers DTCIBR, DTCSQE, and DTCDISP, and allocate the DTC index table.

[4] Setting the DTCCR.RRS bit to 1 enables skipping of the second and the subsequent transfer information read cycles for continuous data transfer due to the same interrupt source. The RRS bit can be set to 1 at any time, but this setting during data transfer becomes valid from the next transfer. Leave the value of the RRS bit as 0 if the value of the MRA.WBDIS bit or the MRC.DISPE bit in any transfer information is 1 or when sequence transfer is to be used.

[5] Set the ICU.DTCERn.DTCE bit (n = interrupt vector number) that corresponds to the DTC request source to 1. Set the ICU.IERm.IENj bit to 1. For the correspondence between interrupt sources and DTCERn register, refer to Table 15.5, Interrupt Vector Table in section 15, Interrupt Controller (ICUE).

[6] Set the interrupt enable bit of the peripheral module, which is request source, to 1. When a source interrupt is generated, the DTC starts operation. For the setting of the interrupt enable bit, refer to the setting of the peripheral module that is to be an request source.

[7] Set the DTCST.DTCST bit to 1 (DTC module start)*1.

Note 1. The DTCST.DTCST bit can be set even if the setting for each request source is not completed.

Figure 20.24 Procedure to Set the DTC

20.6 Examples of DTC Usage

20.6.1 Normal Transfer

As an example of DTC usage, its employment in the reception of 128 bytes of data by an SCI is described below.

(1) Transfer Information Setting

Set the MRA.MD[1:0] bits to 00b (normal transfer mode), the MRA.SZ[1:0] bits to 00b (byte transfer), and the MRA.SM[1:0] bits to 00b (source address is fixed). Set the MRB.CHNE bit to 0 (chain transfer is disabled), the MRB.DISEL bit to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers), and the MRB.DM[1:0] bits to 10b (DAR is incremented after data transfer). The MRB.DTS bit can be set to any value. Set the RDR register address of the SCI in the SAR register, the start address of the RAM area for data storage in the DAR register, and 128 (0080h) in the CRA register. The CRB register can be set to any value.

(2) DTC Vector Table Setting

The start address of the transfer information for the RXI interrupt is set in the vector table for the DTC.

(3) ICU Setting and DTC Module Activation

Set the corresponding ICU.DTCERn.DTCE bit to 1 and the ICU.IERm.IENj bit to 1.
Set the DTCST.DTCST bit to 1.

(4) SCI Setting

Enable the RXI interrupt by setting the SCR.RIE bit in the SCI to 1. If a reception error occurs during the SCI receive operation, further reception is not performed. Accordingly, make settings so that the CPU can accept receive error interrupts.

(5) DTC Transfer

Every time the reception of 1 byte by the SCI is completed, an RXI interrupt is generated to start the data transfer. The DTC transfers the received byte from the RDR of the SCI to RAM, after which the DAR register is incremented and the CRA register is decremented.

(6) Interrupt Handling

After 128 times of data transfers have been completed and the value in the CRA register becomes 0, an RXI interrupt request is output to the CPU. Complete the process in the handling routine for this interrupt.

20.6.2 Chain Transfer

As an example of chain transfer by the DTC, its employment in the output of pulses by a PPG is described below. Lower-case letters x, y, and k in this text indicate a unit, channel, and bit number, respectively.
Chain transfer is used to transfer pulse output data and change the period of the output trigger for the PPG. For the first half of the chain transfer, repeat transfer mode for transfer to the PPGx.NDRH and PPGx.NDRL registers is specified. For the second half, normal transfer mode for transfer to the MTUy.TGR registers is specified. This is because clearing of the request source and generation of an interrupt on completion of the specified number of data transfer are restricted to the second half of the chain transfer (transfer while the MRB.CHNE bit is 0).
An example of using the compare match interrupt for an MTUy.TGRA register as a request source for the DTC is provided below.

(1) First Transfer Information Setting

Settings should be made for transfer to the PPGx.NDRH and PPGx.NDRL registers. Set the MRA.MD[1:0] bits to 01b (repeat transfer mode), the MRA.SZ[1:0] bits to 01b (word transfer), and the MRA.SM[1:0] bits to 10b (SAR is incremented after data transfer). Set the MRB.CHNE bit to 1 (chain transfer is enabled), the MRB.CHNS bit to 0 (chain transfer is performed after each transfer), the MRB.DTS bit to 1 (source is repeat area), and the MRB.DM[1:0] bits to 00b (destination address is fixed). Set the SAR to the start address of the data table, the DAR register to the address of the PPGx.NDRH register, and the CRAH and CRAL registers to the size of the data table. The CRB register can be set to any value.

(2) Second Transfer Information Setting

Settings should be made for transfer to the MTUy.TGRA register. Set the MRA.MD[1:0] bits to 00b (normal transfer mode), the MRA.SZ[1:0] bits to 01b (word transfer), and the MRA.SM[1:0] bits to 10b (SAR is incremented after data transfer). Set the MRB.CHNE bit to 0 (chain transfer is disabled), the MRB.DISEL bit to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers), and the MRB.DM[1:0] bits to 00b (destination address is fixed). The MRB.DTS bit can be set to any value. Set the SAR register to the start address of the data table, the DAR register to the address of the MTUy.TGRA register, and the CRA register to the size of the data table. The CRB register can be set to any value.

(3) Transfer Information Assignment

Place the second transfer information immediately after the first transfer information.

(4) DTC Vector Table

In the DTC vector table, set the address where the first transfer information starts.

(5) ICU Setting and DTC Module Activation

Set the ICU.DTCERn.DTCE bit corresponding to the TGIA interrupt and the ICU.IERm.IENj bit to 1.
Set the DTCST.DTCST bit to 1.

(6) MTU Setting

In the given MTUy, set the TIOR register so that the TGRA register operates as an output compare register (with output disabled) and make the TIER setting to enable TGIAn interrupt requests.

(7) PPG Setting

Set the default output values in the PPGx.PODRH and PPGx.PODRL registers and the next output values in the PPGx.NDRH and PPGx.NDRL registers. Set 1 to the output bits in PORTm.PDR and PPGx.NDRH, and PPGx.NDRL. Also, select a compare match signal of the MTU as the output trigger in the PPGx.PCR register.

(8) MTU Activation

Set the MTU.TSTR.CSTk bits to 1 to start counting operation of the MTUy.TCNT counter.

(9) DTC Transfer

Every time a compare-match with the MTUy.TGRA register is generated, next output values are transferred to the PPGx.NDRH and PPGx.NDRL registers and the setting for the next output trigger period is transferred to the MTUy.TGRA register.

(10) Interrupt Handling

When the specified number of data transfers is completed (i.e. when the value of the CRA register for second transfer becomes 0), a TGIAn interrupt request is issued for the CPU. Complete the process in the handling routine for this interrupt.

20.6.3 Chain Transfer When the Counter is 0

The second data transfer is performed only when the transfer counter is set to 0 in the first data transfer, and the first data transfer information is repeatedly changed in the second data transfer. Repeating this chain transfer enables transfers to be repeated more than 256 times.
The following shows an example of configuring a 128-Kbyte input buffer to addresses 20 0000h to 21 FFFFh (where the input buffer is set so that its lower address starts with 0000h). Figure 20.25 shows a chain transfer when the counter is 0.

(1) Set normal transfer mode for input data for the first data transfer. Set the following:
Transfer source address: Fixed, the CRA register is 0000h (65,536 times), the MRB.CHNE bit is 1 (chain transfer is enabled), the MRB.CHNS bit is 1 (chain transfer is performed only when the transfer counter becomes 0), and the MRB.DISEL bit is 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers).
(2) Prepare the upper 8 bits (in this case, 21h and 20h) of the start address at every 65,536 times of the transfer destination address for the first data transfer in another area (such as ROM).
(3) For the second data transfer, set repeat transfer mode (source is repeat area) for rewriting the transfer destination address of the first data transfer. The transfer destination is the address where the upper 8 bits of the DAR register in the first transfer information is allocated. In this case, set the MRB.CHNE bit to 0 (chain transfer is disabled) and the MRB.DISEL bit to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers). In this case, set the transfer counter to 2.
(4) When a transfer request is accepted, the first data transfer is executed. When transfer is executed 65,536 times and the transfer counter of the first data transfer becomes 0, the second data transfer is started and the upper 8 bits of the transfer destination address of the first data transfer is set to 21h. At this time, the lower 16 bits of the transfer destination address and the transfer counter of the first data transfer have become 0000h.
(5) In succession, when another transfer request is accepted, the first data transfer is executed. When transfer is executed 65,536 times and the transfer counter of the first data transfer becomes 0, the second data transfer is started and the upper 8 bits of the transfer destination address of the first data transfer is set to 20h. At this time, the lower 16 bits of the transfer destination address and the transfer counter of the first data transfer have become 0000h.
(6) Steps (4) and (5) above are repeated infinitely. Because the second data transfer is in repeat transfer mode, no interrupt request to the CPU is generated.

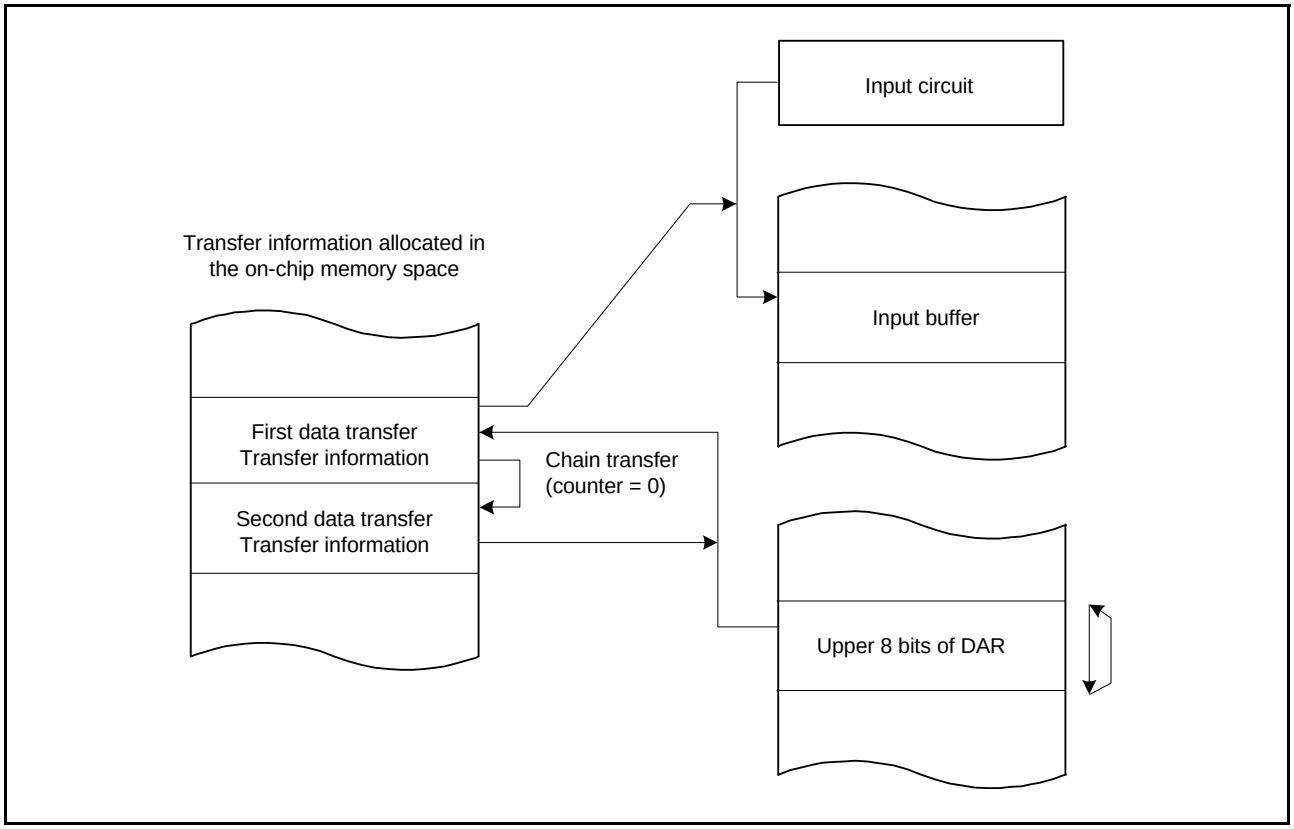


Figure 20.25 Chain Transfer When the Counter is 0

20.6.4 Sequence Transfer

The following is an example of using the SCI receive interrupt as a request source of sequence transfer.

(1) Transfer Information Settings

Set the MRA.MD[1:0] bits to 00b (normal transfer mode), the MRA.SZ[1:0] bits to 00b (byte transfer), and the MRA.SM[1:0] bits to 00b (source address is fixed). Set the MRB.CHNE bit to 0 (chain transfer is disabled), the MRB.DISEL bit to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers), the MRB.DM[1:0] bits to 10b (DAR is incremented after data transfer), the MRB.INDX bit to 1 (start sequence transfer), and the MRB.SQEND bit to 0 (continue the sequence transfer). The MRB.DTS bit can be set to any value. Set the address of the SCIk.RDR register in the SAR register and set the start address of the RAM area which stores the data in the DAR register.
When the MRA.WBDIS bit is set to 1 (Does not write back the transfer information), the values of registers CRA and CRB are ignored.

(2) DTC Vector Table Setting

Set the start address of the transfer information for the corresponding receive data full interrupt (RXI) in the DTC vector table.

(3) DTC Index Table Setting

Set the start address of the transfer information for each sequence in the DTC index table.

(4) ICU Setting and DTC Module Activation

Set the corresponding ICU.DTCERn.DTCE bit to 1 and the ICU.IERm.IENj bit to 1. Set the DTCST.DTCST bit to 1.

(5) SCI Setting

Set the SCIk.SCR.RIE bit to 1 to enable the RXI interrupt. If a reception error occurs during the SCI receive operation, subsequent receptions are not performed. Accordingly, make settings so that the CPU can accept receive error interrupts.

(6) Start of the Sequence Transfer

On completion of reception of 1-byte data by the SCI, an RXI interrupt is generated to start the DTC. The DTC transfers the received data from the SCIk.RDR register to the RAM. The DTC looks up the DTC index table by using the value from the received data (sequence number) and continues to transfer data corresponding to the that number.
When the CPUSEL bit in the DTC index is 1, the DTC does not read the transfer information and sets the ICU.DTCERn.DTCE bit to 0. Then the DTC outputs an interrupt request to the CPU and ends the sequence transfer.

(7) During Suspension of the Sequence Transfer

Set the ICU.DTCERn.DTCE bit to 1 if the bit is 0. The DTC continues to transfer the data for every generation of the DTC transfer request in response to the corresponding RXI interrupt.

(8) End of the Sequence Transfer

Set the MRB.SQEND bit in the last transfer information of the sequence transfer to 1. After execution of this data transfer, the DTC ends the sequence transfer. The DTC starts to refer to the DTC vector table when a DTC transfer request is generated due to the next corresponding RXI interrupt.

20.7 Interrupt Source

When the DTC has finished data transfer of specified count or when data transfer with the MRB.DISEL bit set to 1 (an interrupt request to the CPU is generated each time the data transfer is performed) has been completed, an interrupt to the CPU is generated by the DTC trigger source. Such interrupts to the CPU are controlled according to the PSW.I bit (interrupt enable) of the CPU, the PSW.IPL[3:0] bits (processor interrupt priority level), and the priority level of the interrupt controller.

20.8 Event Link

The DTC outputs an event signal on completing data transfer in response to one request. When the destination for transfer is an external bus or an internal peripheral bus, however, the event signal will be output after completion of writing to the write buffer rather than after completion of writing to the actual destination for transfer.

20.9 Low Power Consumption Function

Before making a transition to the module stop state, all-module clock stop mode, software standby mode, or deep software standby mode, set the DTCST.DTCST bit to 0 (DTC module stop), and then perform the following.

(1) Module Stop Function

Writing 1 (transition to the module-stop state is made) to the MSTPCRA.MSTPA28 bit enables the module stop function of the DTC. If data transfer is in progress at the time 1 is written to the MSTPCRA.MSTPA28 bit, the transition to the module stop state proceeds after data transfer has ended. While the MSTPCRA.MSTPA28 bit is 1, accessing the DTC registers is prohibited.
Writing 0 (release from the module-stop state) to the MSTPCRA.MSTPA28 bit releases the DTC from the module stop state.

(2) All-Module Clock Stop Mode

Make settings according to the procedure under section 11.6.2.1, Transition to All-Module Clock Stop Mode, in section 11, Low Power Consumption.
If any data transfer is in progress at the time the WAIT instruction is executed, the transition to all-module clock stop mode follows the completion of the data transfer.
The DTC is released from the module stop state by writing 0 to the MSTPCRA.MSTPA28 bit following recovery from all-module clock stop mode.

(3) Software Standby and Deep Software Standby Modes

Make settings according to the procedure under section 11.6.3.1, Transition to Software Standby Mode, or section 11.6.4.1, Transition to Deep Software Standby Mode, in section 11, Low Power Consumption.
If any data transfer is in progress at the time the WAIT instruction is executed, the transition to software standby mode or deep software standby mode follows the completion of the data transfer.

(4) Notes on Low Power Consumption Function

For the WAIT instruction and the register setting procedure, refer to section 11.7.6, Timing of WAIT Instructions in section 11, Low Power Consumption.
To perform data transfer after returning from a low power consumption mode, set the DTCST.DTCST bit to 1 again.
To use a request that is generated in all-module clock stop mode or software standby mode as an interrupt request to the CPU but not as a DTC transfer request, specify the CPU as the interrupt request destination according to the description in section 15.7.3.1, Interrupt Request Destination Setting Procedure in section 15, Interrupt Controller (ICUE), and then execute the WAIT instruction.

20.10 Usage Notes

20.10.1 Start Address of Transfer Information

Set multiples of 4 for the start addresses of the transfer information to be specified in the DTC vector table. If any value other than a multiple of 4 is specified, access still proceeds with the lower 2 bits of the address regarded as 00b.

20.10.2 Allocating Transfer Information

Allocate transfer information in the memory area according to the endian of the area as shown in Figure 20.26. For example, when writing CRA and CRB settings in 16-bit units in big endian, write the CRA setting to the address plus 8h (Ch) and the CRB setting to the address plus Ah (Eh). In little endian, write the CRB setting to the address plus 8h (Ch) and the CRA setting to the address plus Ah (Eh). When writing CRA and CRB settings in 32-bit units, allocate the CRA setting at the MSB side of the 32 bits and the CRB setting at the LSB side, and write the settings to the address plus 8h (Ch), regardless of endian.

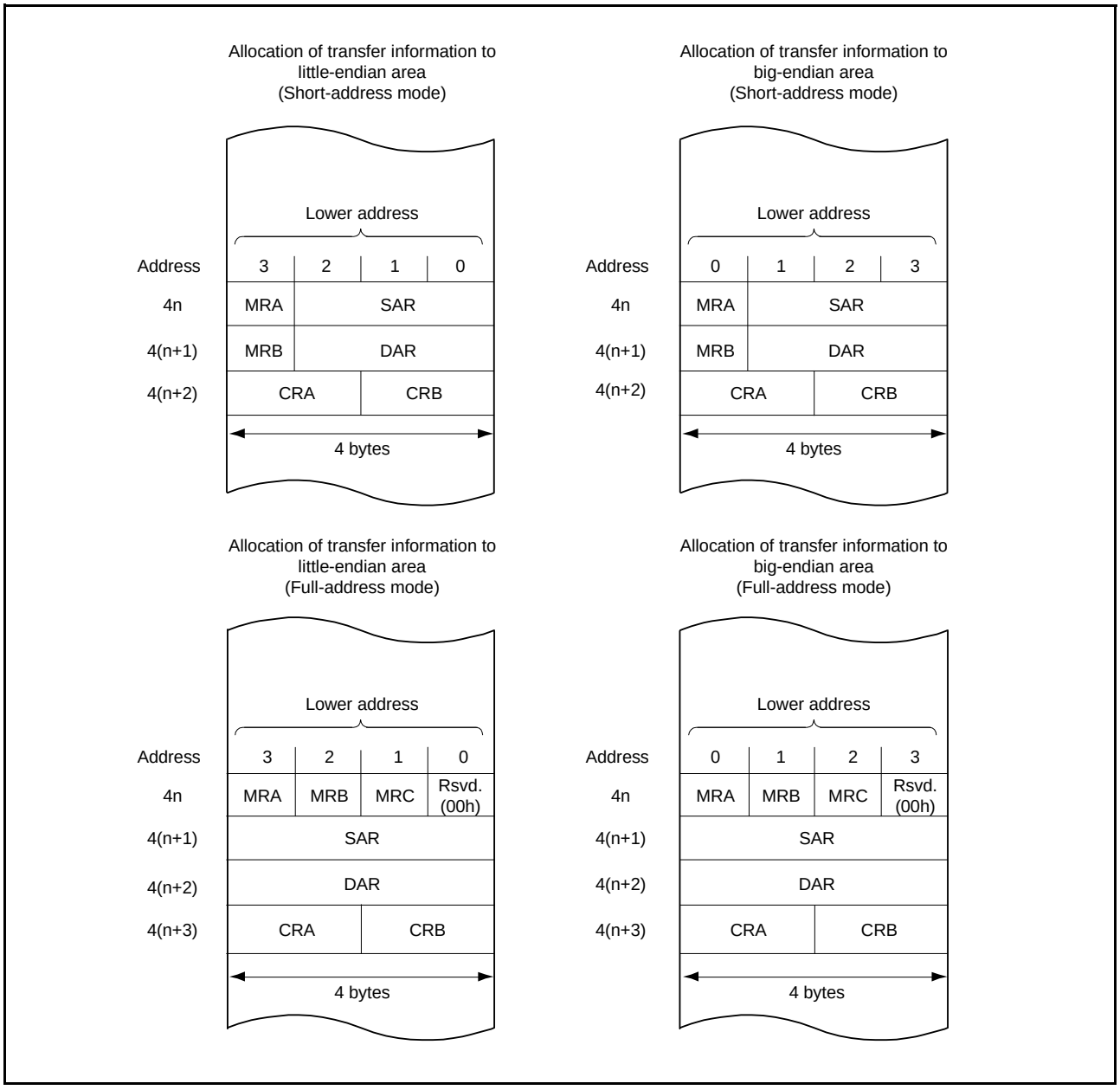


Figure 20.26 Allocation of Transfer Information

20.10.3 Setting the DTC Transfer Request Enable Register in the Interrupt Controller (ICU.DTCERn)

The DMA request should not be issued by setting the DMAC trigger select register (ICU.DMRSRm (m = DMAC channel number)) to the same vector number that has been specified by setting the ICU.DTCERn.DTCE bit to 1 (the corresponding interrupt source is selected as the DTC trigger). For details on the ICU.DTCERn and ICU.DMRSRm registers (m = DMAC channel number), refer to section 15, Interrupt Controller (ICUE).

20.10.4 Notes on Using the Sequence Transfer

When sequence transfer is to be used, make sure that the DTCADMOD.SHORT bit is 0 (full-address mode) and the DTCCR.RRS bit is also 0 (transfer information read is not skipped).
In addition, set the MRB.CHNE bit to 0 (chain transfer is disabled) when setting the MRB.INDX bit to 1 (start sequence transfer and refer the index table) or the MRB.SQEND bit to 1 (end the sequence transfer).

21. Event Link Controller (ELC)

21.1 Overview

The event link controller (ELC) uses the interrupt requests generated by various peripheral modules as event signals, and interconnects (links) peripheral modules. As a result, peripheral modules can directly perform interlinked operation among them without using software.
Event signals can be output regardless of the settings of the corresponding interrupt request enable bits.

Table 21.1 lists the specifications of the ELC, and Figure 21.1 shows a block diagram of the ELC.

Table 21.1 ELC Specifications

Item	Description
Event link function	• 99 types of event signals can be directly interconnected to modules. • Operation for timer modules when inputting an event signal can be selected. • Event linkage operation is possible for port B and port E. Single port*1: Event linkage operation can be set in a single specified port. Port group*1: Event linkage operation can be set by grouping multiple specified ports among total of eight ports.
Low power consumption function	Module stop state can be set.

Note 1. When an input signal to a corresponding pin changes, an event is generated in a single port or in a port group specified as the input.

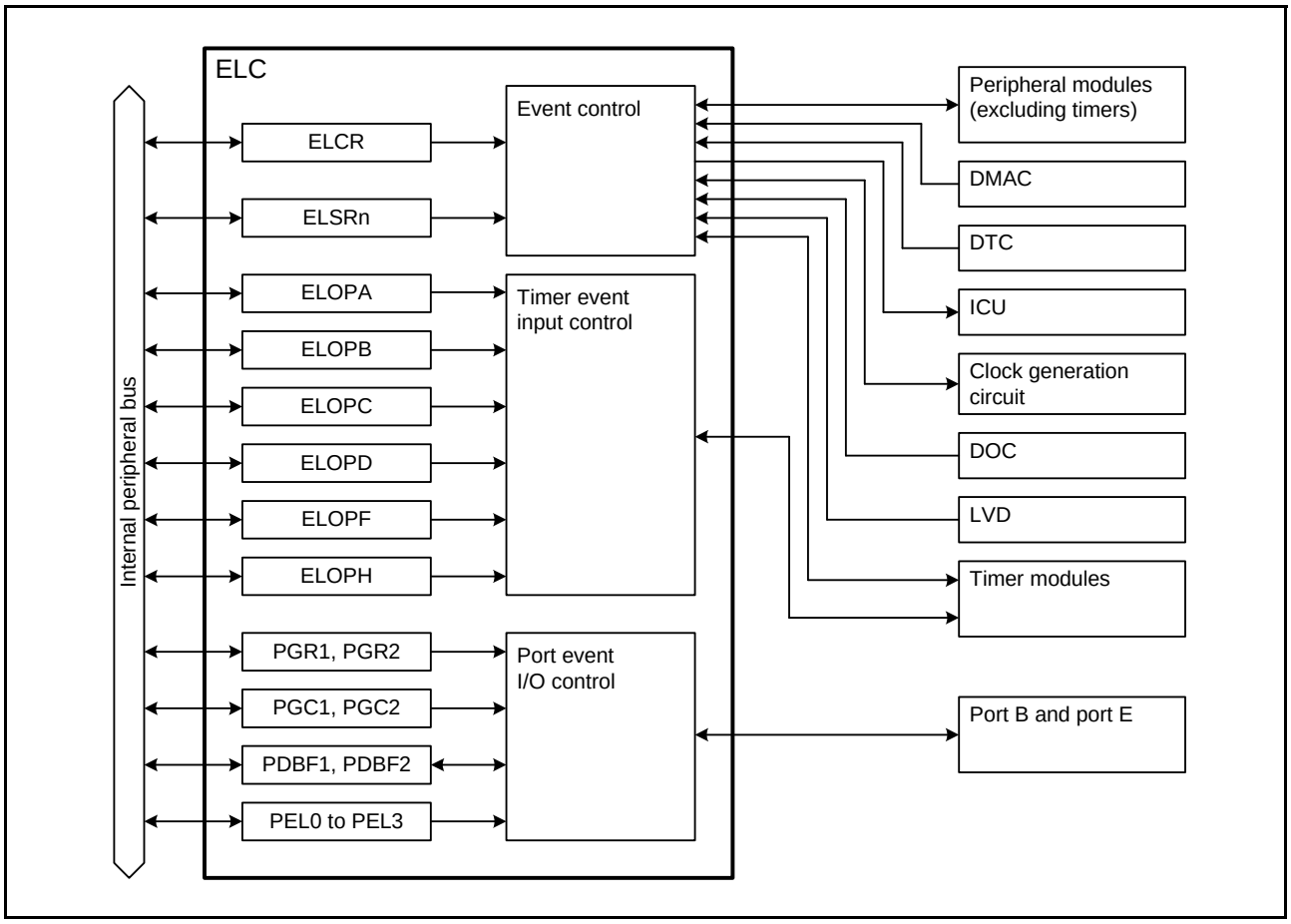


Figure 21.1 ELC Block Diagram (n = 0, 3, 4, 7, 10 to 15, 18 to 28, 33, 35 to 38, 45)

21.2 Register Descriptions

21.2.1 Event Link Control Register (ELCR)

Address(es): ELC.ELCR 0008 B100h

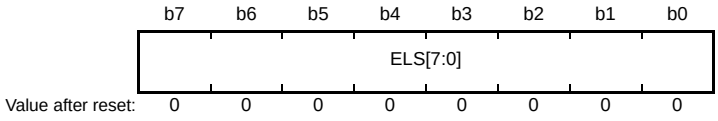
	b7	b6	b5	b4	b3	b2	b1	b0
	ELCON	—	—	—	—	—	—	—
Value after reset:	0	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b7	ELCON	All Event Link Enable	0: ELC function is disabled. 1: ELC function is enabled.	R/W

The ELCR register controls operation of the ELC.

21.2.2 Event Link Setting Register n (ELSRn) (n = 0, 3, 4, 7, 10 to 15, 18 to 28, 33, 35 to 38, 45)

Address(es): ELC.ELSR0 0008 B101h, ELC.ELSR3 0008 B104h, ELC.ELSR4 0008 B105h, ELC.ELSR7 0008 B108h, ELC.ELSR10 0008 B10Bh, ELC.ELSR11 0008 B10Ch, ELC.ELSR12 0008 B10Dh, ELC.ELSR13 0008 B10Eh, ELC.ELSR14 0008 B10Fh, ELC.ELSR15 0008 B110h, ELC.ELSR18 0008 B113h, ELC.ELSR19 0008 B114h, ELC.ELSR20 0008 B115h, ELC.ELSR21 0008 B116h, ELC.ELSR22 0008 B117h, ELC.ELSR23 0008 B118h, ELC.ELSR24 0008 B119h, ELC.ELSR25 0008 B11Ah, ELC.ELSR26 0008 B11Bh, ELC.ELSR27 0008 B11Ch, ELC.ELSR28 0008 B11Dh, ELC.ELSR33 0008 B131h, ELC.ELSR35 0008 B133h, ELC.ELSR36 0008 B134h, ELC.ELSR37 0008 B135h, ELC.ELSR38 0008 B136h, ELC.ELSR45 0008 B13Dh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	ELS[7:0]	Event Link Select	00h: Event signal output to the corresponding peripheral module is disabled. 01h to DFh: Set the number for the event signal to be linked. Settings other than above are prohibited.	R/W

The ELSRn register specifies an event signal to be linked to for each peripheral module. Table 21.2 shows the correspondence between the ELSRn register and the peripheral modules. Table 21.3 shows the correspondence between values set in the ELSRn register and event signals.

Table 21.2 Correspondence between the ELSRn Register and the Peripheral Modules

Register Name	Peripheral Module
ELSR0	MTU0
ELSR3	MTU3
ELSR4	MTU4
ELSR7	CMT1
ELSR10	TMR0
ELSR11	TMR1
ELSR12	TMR2
ELSR13	TMR3
ELSR14	CTSU
ELSR15	S12AD
ELSR18	ICU (Interrupt 1)*1
ELSR19	ICU (Interrupt 2)*1
ELSR20	Output port group 1
ELSR21	Output port group 2
ELSR22	Input port group 1
ELSR23	Input port group 2
ELSR24	Single port 0*2
ELSR25	Single port 1*2
ELSR26	Single port 2*2
ELSR27	Single port 3*2
ELSR28	Clock source switching to LOCO
ELSR33	CMTW0
ELSR35	TPU0
ELSR36	TPU1
ELSR37	TPU2
ELSR38	TPU3
ELSR45	S12AD1

Note 1. Specify an event number from among 63h to 6Ah. Do not set other values.
Note 2. Do not set the DOC data operation condition met signal (6Ah) in the ELSR24, ELSR25, ELSR26, and ELSR27 registers.

Table 21.3 Correspondence between Values Set in ELSRn.ELS[7:0] Bits and Event Signals (1/3)

ELS[7:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
01h	Multifunction timer pulse unit 3	MTU0 compare match 0A
02h		MTU0 compare match 0B
03h		MTU0 compare match 0C
04h		MTU0 compare match 0D
05h		MTU0 compare match 0E
06h		MTU0 compare match 0F
07h		MTU0 overflow
10h		MTU3 compare match 3A
11h		MTU3 compare match 3B
12h		MTU3 compare match 3C
13h		MTU3 compare match 3D
14h		MTU3 overflow
15h		MTU4 compare match 4A
16h		MTU4 compare match 4B
17h		MTU4 compare match 4C
18h		MTU4 compare match 4D
19h		MTU4 overflow
1Ah		MTU4 underflow
1Fh	Compare match timer	CMT1 compare match 1
22h	8-bit timers	TMR0 compare match A0
23h		TMR0 compare match B0
24h		TMR0 overflow
25h		TMR1 compare match A1
26h		TMR1 compare match B1
27h		TMR1 overflow
28h		TMR2 compare match A2
29h		TMR2 compare match B2
2Ah		TMR2 overflow
2Bh		TMR3 compare match A3
2Ch		TMR3 compare match B3
2Dh		TMR3 overflow
2Eh	Realtime clock	RTC periodic event (select 1/256, 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1, or 2 seconds)
31h	Independent watchdog timer	IWDT underflow or refresh error
3Ah	Serial communications interfaces	SCI5 error (receive error or error signal detection)
3Bh		SCI5 receive data full
3Ch		SCI5 transmit data empty
3Dh		SCI5 transmit end
4Eh	I^2C-bus interface	RIIC0 communication error or event generation
4Fh		RIIC0 receive data full
50h		RIIC0 transmit data empty
51h		RIIC0 transmit end

Table 21.3 Correspondence between Values Set in ELSRn.ELS[7:0] Bits and Event Signals (2/3)

ELS[7:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
52h	Serial peripheral interface	RSPI0 error (mode fault, overrun, underrun, or parity error)
53h		RSPI0 idle
54h		RSPI0 receive buffer full
55h		RSPI0 transmit buffer empty
56h		RSPI0 communication end
58h	12-bit A/D converter	S12AD A/D conversion end
5Bh	Voltage detection circuit	LVD1 voltage detection
5Ch		LVD2 voltage detection
5Dh	DMA controller	DMAC0 transfer end
5Eh		DMAC1 transfer end
5Fh		DMAC2 transfer end
60h		DMAC3 transfer end
61h	Data transfer controller	DTC transfer end
62h	Clock generation circuit	Oscillation stop detection of clock generation circuit
63h	I/O ports	Input edge detection of input port group 1
64h		Input edge detection of input port group 2
65h		Input edge detection of single input port 0
66h		Input edge detection of single input port 1
67h		Input edge detection of single input port 2
68h		Input edge detection of single input port 3
69h	Event link controller	Software event
6Ah	Data operation circuit	DOC data operation condition met
6Ch	12-bit A/D converter	S12AD1 A/D conversion end
7Eh	Compare match timer W	CMTW0 compare match
ACh	16-bit timer pulse unit	TPU0 compare match A
ADh		TPU0 compare match B
AEh		TPU0 compare match C
AFh		TPU0 compare match D
B0h		TPU0 overflow
B1h		TPU1 compare match A
B2h		TPU1 compare match B
B3h		TPU1 overflow
B4h		TPU1 underflow
B5h		TPU2 compare match A
B6h		TPU2 compare match B
B7h		TPU2 overflow
B8h		TPU2 underflow
B9h		TPU3 compare match A
BAh		TPU3 compare match B
BBh		TPU3 compare match C
BCh		TPU3 compare match D
BDh		TPU3 overflow

Table 21.3 Correspondence between Values Set in ELSRn.ELS[7:0] Bits and Event Signals (3/3)

ELS[7:0] Bit Value	Peripheral Modules	Event Signal Set in ELSRn
D0h	Serial communication interface	RSCI10 error
D1h		RSCI10 receive data full
D2h		RSCI10 receive data matched
D3h		RSCI10 transmit data empty
D4h		RSCI10 transmit end
D5h		RSCI10 receive data unmatched
D6h		RSCI10 effective edge detected
D7h	High-speed I^2C-bus interface	RIICHS0 communication error or event generation
D8h		RIICHS0 receive data full
D9h		RIICHS0 transmit data empty
DAh		RIICHS0 transmit end
DBh	Serial peripheral interface	RSPIA0 error
DCh		RSPIA0 idle
DDh		RSPIA0 receive buffer full
DEh		RSPIA0 transmit buffer empty
DFh		RSPIA0 communication end
Settings other than above are prohibited.		

21.2.3 Event Link Option Setting Register A (ELOPA)

Address(es): ELC.ELOPA 0008 B11Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	MTU3MD[1:0]		—	—	—	—	MTU0MD[1:0]	
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	MTU0MD[1:0]	MTU0 Operation Select	b1 b0 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*1 1 1: Event output is disabled.	R/W
b5 to b2	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b7, b6	MTU3MD[1:0]	MTU3 Operation Select	b7 b6 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*2 1 1: Event output is disabled.	R/W

Note 1. The MTU0.TCNT value is captured into the MTU0.TGRA register.
Note 2. The MTU3.TCNT value is captured into the MTU3.TGRA register.

The ELOPA register specifies the operations of MTU0 and MTU3 when an event signal is input. Set 11b (event output is disabled) when the ELC function is not used.

21.2.4 Event Link Option Setting Register B (ELOPB)

Address(es): ELC.ELOPB 0008 B120h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	MTU4MD[1:0]	
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	MTU4MD[1:0]	MTU4 Operation Select	b1 b0 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*1 1 1: Event output is disabled.	R/W
b7 to b2	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. The MTU4.TCNT value is captured into the MTU4.TGRA register.

The ELOPB register specifies the operation of MTU4 when an event signal is input. Set 11b (event output is disabled) when the ELC function is not used.

21.2.5 Event Link Option Setting Register C (ELOPC)

Address(es): ELC.ELOPC 0008 B121h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CMT1MD[1:0]		—	—
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b3, b2	CMT1MD[1:0]	CMT1 Operation Select	b3 b2 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event output is disabled.	R/W
b7 to b4	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

The ELOPC register specifies the operation of CMT1 when an event signal is input. Set 11b (event output is disabled) when the ELC function is not used.

21.2.6 Event Link Option Setting Register D (ELOPD)

Address(es): ELC.ELOPD 0008 B122h

	b7	b6	b5	b4	b3	b2	b1	b0
	TMR3MD[1:0]		TMR2MD[1:0]		TMR1MD[1:0]		TMR0MD[1:0]	
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TMR0MD[1:0]	TMR0 Operation Select	b1 b0 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event output is disabled.	R/W
b3, b2	TMR1MD[1:0]	TMR1 Operation Select	b3 b2 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event output is disabled.	R/W
b5, b4	TMR2MD[1:0]	TMR2 Operation Select	b5 b4 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event output is disabled.	R/W
b7, b6	TMR3MD[1:0]	TMR3 Operation Select	b7 b6 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event output is disabled.	R/W

The ELOPD register specifies the operations of TMR0 to TMR3 when an event signal is input. Set 11b (event output is disabled) when the ELC function is not used.

21.2.7 Event Link Option Setting Register F (ELOPF)

Address(es): ELC.ELOPF 0008 B13Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	TPU3MD[1:0]		TPU2MD[1:0]		TPU1MD[1:0]		TPU0MD[1:0]	
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TPU0MD[1:0]	TPU0 Operation Select	b1 b0 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*1 1 1: Event output is disabled.	R/W
b3, b2	TPU1MD[1:0]	TPU1 Operation Select	b3 b2 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*2 1 1: Event output is disabled.	R/W
b5, b4	TPU2MD[1:0]	TPU2 Operation Select	b5 b4 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*3 1 1: Event output is disabled.	R/W
b7, b6	TPU3MD[1:0]	TPU3 Operation Select	b7 b6 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture*4 1 1: Event output is disabled.	R/W

Note 1. The TPU0.TCNT value is captured into the TPU0.TGRA register.
Note 2. The TPU1.TCNT value is captured into the TPU1.TGRA register.
Note 3. The TPU2.TCNT value is captured into the TPU2.TGRA register.
Note 4. The TPU3.TCNT value is captured into the TPU3.TGRA register.

The ELOPF register specifies the operations of TPU0 to TPU3 when an event signal is input. Set 11b (event output is disabled) when the ELC function is not used.

21.2.8 Event Link Option Setting Register H (ELOPH)

Address(es): ELC.ELOPH 0008 B141h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CMTW0MD[1:0]	
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CMTW0MD[1:0]	CMTW0 Operation Select	b1 b0 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Event counter 1 1: Event output is disabled.	R/W
b7 to b2	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

The ELOPH register specifies the operation of CMTW0 when an event signal is input. Set 11b (event output is disabled) when the ELC function is not used.

21.2.9 Port Group Setting Register n (PGRn) (n = 1, 2)

Address(es): ELC.PGR1 0008 B123h, ELC.PGR2 0008 B124h

	b7	b6	b5	b4	b3	b2	b1	b0
	PGR7	PGR6	PGR5	PGR4	PGR3	PGR2	PGR1	PGR0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PGR0	Port Group Setting 0	0: Does not specify the port as a member of the port group. 1: Specifies the port as a member of the port group.	R/W
b1	PGR1	Port Group Setting 1		R/W
b2	PGR2	Port Group Setting 2		R/W
b3	PGR3	Port Group Setting 3		R/W
b4	PGR4	Port Group Setting 4		R/W
b5	PGR5	Port Group Setting 5		R/W
b6	PGR6	Port Group Setting 6		R/W
b7	PGR7	Port Group Setting 7		R/W

The PGRn register specifies a group of I/O ports. Among the ports, ports corresponding to bits set to 1 in the register are selected for a port group.

For example, when the PGR6 and PGR3 bits in the PGR1 register are set to 1, the PB6 and PB3 pins are selected to a port group.

Table 21.4 shows the PGRn register and corresponding ports.

Table 21.4 Registers Related to Port Groups and Corresponding Port Numbers

Port Number	Port Group Setting Register (PGR)	Port Group Control Register (PGC)	Port Buffer Register (PDBF)
Port B	PGR1 register	PGC1 register	PDBF1 register
Port E	PGR2 register	PGC2 register	PDBF2 register

21.2.10 Port Group Control Register n (PGCn) (n = 1, 2)

Address(es): ELC.PGC1 0008 B125h, ELC.PGC2 0008 B126h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PGCO[2:0]			—	PGCOVE	PGCI[1:0]	
Value after reset:	1	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PGCI[1:0]	Event Output Edge Select	b1 b0 0 0: Event signal is output upon detection of the rising edge of the input signal to the port. 0 1: Event signal is output upon detection of the falling edge of the input signal to the port. 1 x: Event signal is output upon detection of both the rising and falling edges of the input signal to the port.	R/W
b2	PGCOVE	PDBF Overwrite	0: Overwriting the PDBFn register is disabled. 1: Overwriting the PDBFn register is enabled.	R/W
b3	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b6 to b4	PGCO[2:0]	Port Group Operation Select	b6 b4 0 0 0: Low is output when an event signal is input. 0 0 1: High is output when an event signal is input. 0 1 0: The output is toggled (inverted) when an event signal is input. 0 1 1: The buffer value is output when an event signal is input. 1 x x: The output data is rotated (from MSB to LSB) in the port group when an event signal is input.	R/W
b7	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

x: Don’t care

For the port group set as an output, the PGCn register specifies the form of outputting the signal from the port when an event signal is input. For the port group set as an input, the PGCn register enables/disables overwriting of the PDBFn register and specifies the conditions of event generation (edge of the input signal).
Specify the I/O direction of the port by the corresponding bit in the PDR register.
Refer to Table 21.4 for the PGCn register and corresponding ports.

21.2.11 Port Buffer Register n (PDBFn) (n = 1, 2)

Address(es): ELC.PDBF1 0008 B127h, ELC.PDBF2 0008 B128h

	b7	b6	b5	b4	b3	b2	b1	b0
	PDBF7	PDBF6	PDBF5	PDBF4	PDBF3	PDBF2	PDBF1	PDBF0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PDBF0	Port Buffer 0	Specify the data to be transferred to the PODR register when an event signal is input. The setting value is valid when the PGCn.PGCO[2:0] bits are 011b or 1xxb. Write access to the bit specified as a member of the input port group is disabled. For details, refer to section 21.3, Operation.	R/W
b1	PDBF1	Port Buffer 1		R/W
b2	PDBF2	Port Buffer 2		R/W
b3	PDBF3	Port Buffer 3		R/W
b4	PDBF4	Port Buffer 4		R/W
b5	PDBF5	Port Buffer 5		R/W
b6	PDBF6	Port Buffer 6		R/W
b7	PDBF7	Port Buffer 7		R/W

The PDBFn register is an 8-bit readable/writable register used in combination with the PGRn register. Refer to section 21.3.6, I/O Port Operation When Event Signal is Input and Event Generation for the PDBFn register operations. Refer to Table 21.4 for the PDBFn register and corresponding ports.

21.2.12 Event Link Port Setting Register m (PELm) (m = 0 to 3)

Address(es): ELC.PEL0 0008 B129h, ELC.PEL1 0008 B12Ah, ELC.PEL2 0008 B12Bh, ELC.PEL3 0008 B12Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PSM[1:0]		PSP[1:0]		PSB[2:0]		
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	PSB[2:0]	Bit Number Specification	Set a bit number for a port to be specified as a single port.	R/W
b4, b3	PSP[1:0]	Port Number Specification	b4 b3 0 0: Setting disabled 0 1: Port B (corresponding to PGR1) 1 0: Port E (corresponding to PGR2) 1 1: Setting prohibited	R/W
b6, b5	PSM[1:0]	Event Link Specification	• For the output port, specify the data to be output from the port. b6 b5 0 0: Low is output when an event signal is input. 0 1: High is output when an event signal is input. 1 x: The output is toggled (inverted) when an event signal is input. • For the input port, select the edge on which the event signal is to be output. b6 b5 0 0: Event signal is output upon detection of the rising edge. 0 1: Event signal is output upon detection of the falling edge. 1 x: Event signal is output upon detection of both the rising and falling edges.	R/W
b7	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

x: Don’t care

The PELm register specifies the single port, the operation upon an event signal input, and the conditions of event generation. This MCU can specify a total of four bits in port B and port E to respective single ports.
Specify the I/O direction of the port by the corresponding bit in the PDR register.

21.2.13 Event Link Software Event Generation Register (ELSEGR)

Address(es): ELC.ELSEGR 0008 B12Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	WI	WE	—	—	—	—	—	SEG
Value after reset:	1	0	1	1	1	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	SEG	Software Event Generation	0: Normal operation 1: Software event is generated.	W
b5 to b1	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b6	WE	SEG Bit Write Enable	0: Write to SEG bit is disabled. 1: Write to SEG bit is enabled.	R/W
b7	WI	ELSEGR Register Write Disable	0: Write to ELSEGR register is enabled. 1: Write to ELSEGR register is disabled.	W

The MOV instruction must be used to write to this register.

SEG Bit (Software Event Generation)

When 1 is written to this bit while the WE bit is 1, a software event is generated.
This bit is read as 0. Even if 1 is written to this bit, this bit does not become 1.

WE Bit (SEG Bit Write Enable)

The SEG bit can be written to only when the WE bit is 1.
To set this bit to 1, write 0 to the WI bit and write 1 to this bit simultaneously.
To set this bit to 0, write 0 to the WI bit and write 0 to this bit simultaneously.

WI Bit (ELSEGR Register Write Disable)

The ELSEGR register can be written to only when the value to be written to the WI bit is 0.
This bit is read as 1.

21.3 Operation

21.3.1 Relation between Interrupt Handling and Event Linking

The peripheral modules incorporated in the MCU are provided with the interrupt request status flags and the interrupt enable bits to enable/disable these interrupt requests. When an interrupt request is generated in a peripheral module, the corresponding interrupt request status flag becomes 1. If the corresponding interrupt request is enabled then, the interrupt is requested to the CPU.

In contrast, the event link controller (ELC) uses the interrupt requests generated by various peripheral modules as event signals, interconnects (links) peripheral modules, and then, makes peripheral modules perform direct interlinked operation among them without using software. Event signals can be output regardless of the setting of the corresponding interrupt enable bit.

Figure 21.2 shows the relation between the interrupt handling and ELC.

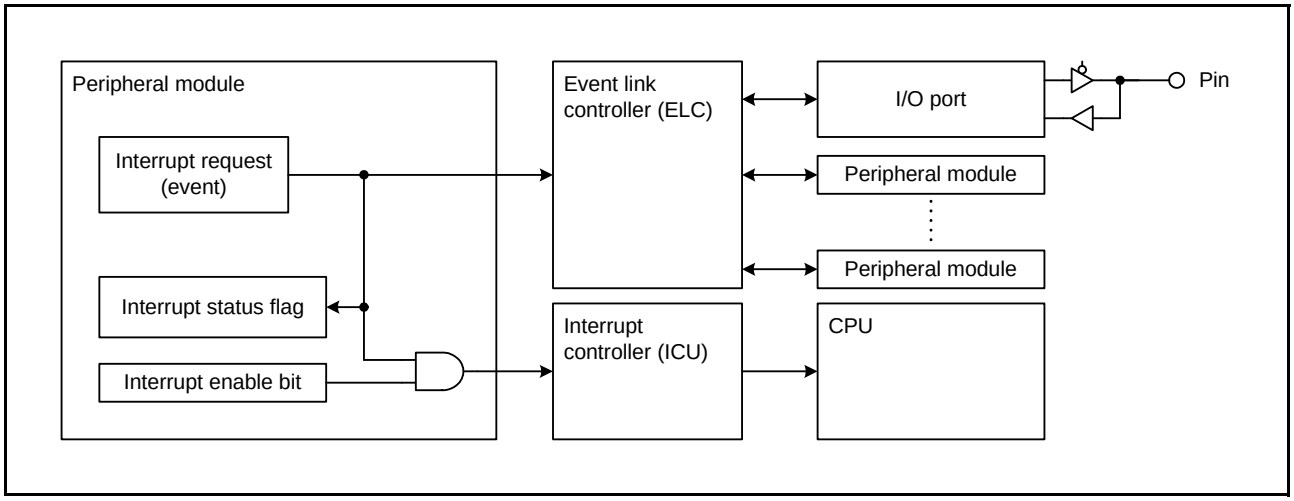


Figure 21.2 Relation between Interrupt Handling and ELC

21.3.2 Event Linkage

When events are specified in the ELSRn registers, the corresponding peripheral modules can be operated at generation of the specified events. A single peripheral module can link only with a single event. Set the ELSRn register after completing the initialization of the peripheral module to operate by an event. Table 21.5 lists the operations of peripheral modules when an event signal is input.

Table 21.5 Operations of Peripheral Modules When Event Signal is Input

Peripheral Module	Operations When Event Signal is Input		
MTU CMT CMTW TMR TPU	The following operations can be selected by setting the ELOPA to ELOPD, ELOPF, and ELOPH registers: • Starts counting when an event signal is input (MTU, CMT, CMTW, TMR, TPU). • Restarts counting when an event signal is input (MTU, CMT, CMTW, TMR, TPU). • Counts the input events (CMT, CMTW, TMR). • Performs input-capture operation when an event signal is input (MTU, TPU).		
CTSU	Starts measurement of electrostatic capacitance when an event signal is input.		
A/D converter	Starts A/D conversion when an event signal is input.		
I/O ports (output)	The value of PODR register (port output data register) changes when an event signal is input (The level output from the corresponding pin changes).	Port group	• Changes the PODR register value to the specified value. • Transfers the PDBFn register value to the PODR register (n = 1, 2). • Rotates the PODR register.
		Single port	Changes the PODR register value to the specified value.
I/O ports (input)	When the signal level of the input pin changes	Port group	Generates an event.
		Single port	
	When an event signal is input	Port group	Transfers the signal level of the input pin to the PDBFn register.
		Single port	This combination cannot be used.
Clock generation circuit	Switches the clock source to the low-speed on-chip oscillator when an event signal is input.*1		
Interrupt controller	Request an interrupt to the CPU, starts DMA transfer, or starts DTC transfer when an event signal is input.		

Note 1. The SCKCR3.CKSEL[2:0] bits are modified to 000b (LOCO) regardless of the value of the protect register (PRCR.PRC0).

21.3.3 Operation of Peripheral Timer Modules When Event Signal is Input

For the timer modules, set the ELOPA to ELOPD, ELOPF, or ELOPH register to specify the operation for when an event signal is input.

(1) Count Start Operation

When an event signal is input, the timer starts counting and the count start bit*1 in each timer control register becomes 1. An event signal that is input while the count start bit is 1 is ignored.

(2) Count Restart Operation

When an event signal is input, the timer counter is cleared. Since the count start bit*1 in each timer control register is retained, counting is restarted when an event signal is input while the count start bit is 1.

(3) Event Counter Operation

Event signal is selected as the timer count source. When an event signal is input, the timer counter is incremented.

(4) Input Capture Operation

When an event signal is input, the timer performs input-capture operation.

Note 1. Refer to the register descriptions on starting the timer in the relevant peripheral timer module section.

21.3.4 Operation of CTSU When Event Signal is Input

When an event signal is input while the CTSUCAP and CTSUSTRT bits in the CTSUCR0 register are set to 1, a measurement is started. Refer to the description of the CTSUCR0.CTSUSTRT bit for details.

21.3.5 Operation of A/D Converter When Event Signal is Input

When an event signal is input, the ADCSR.ADST bit*1 is set to 1 and the A/D converter start A/D conversion.

Note 1. Refer to the bit description in the A/D converter section.

21.3.6 I/O Port Operation When Event Signal is Input and Event Generation

The I/O port operation at an event signal input and conditions for event generation are set by the registers in ELC. The I/O ports that are used to set an event linkage are port B and port E.

(1) Single Ports and Port Groups

There are two event link modes: event link to single ports and event link to port groups. In the former mode, events can be interconnected to any one of the I/O ports. In the latter mode, events can be interconnected to port groups consisting of any two or more bits in the same I/O ports.

A single port can be set by the PELm.PSP[1:0] and PSB[1:0] bits (m = 0 to 3). A port group can be specified by setting two or more bits in the PGRn register (n = 1, 2) to 1. Among the ports corresponding to the bits set to 1 in the PGRn register, a port set as output becomes an output port group member, and a port set as input becomes an input port group member.

If an I/O port is specified as both a single port and a member of a port group, both functions are enabled when the corresponding port is input, whereas only the port group function is enabled when the corresponding port is output.

Set the PDR register to select the direction of the I/O ports.

(2) Event Generation in Single Input Ports

A single port that is set as input generates an event signal when the input signal to the corresponding pin changes. The event generation condition is specified using the PELm.PSM[1:0] bits (m = 0 to 3). An example of operation is shown in Figure 21.3 (1).

(3) Single Output Ports Operation When Event Signal is Input

When an event signal is input to a single port set as output, the output level (the PODR register value) of the corresponding pin changes as specified by the PELm.PSM[1:0] bits. An example of operation is shown in Figure 21.3 (2).

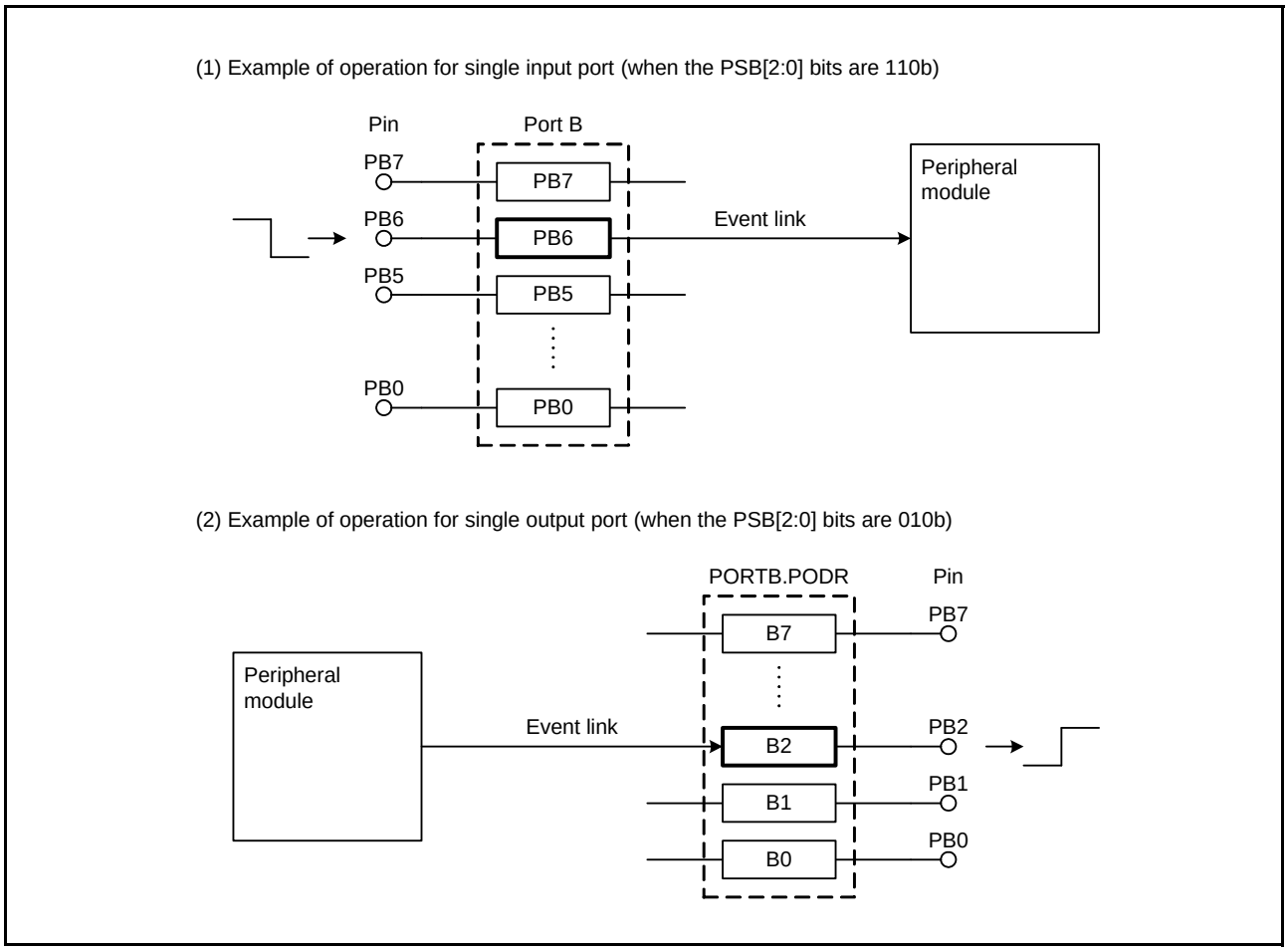


Figure 21.3 Event Linkage Related to Single Ports (Port B)

(4) Event Generation in Input Port Group

An input port group generates an event signal when any of input signals to the corresponding pins change. The event generation condition is specified using the PGCn.PGCI[1:0] bits (n = 1, 2).

(5) Input Port Group Operation When Event Signal is Input

When an event signal is input to an input port group, the level of the corresponding pins is transferred to the PDBFn register. Values of the bits corresponding to ports that are not specified as members of the input port group do not change. An example of operation is shown in Figure 21.4.

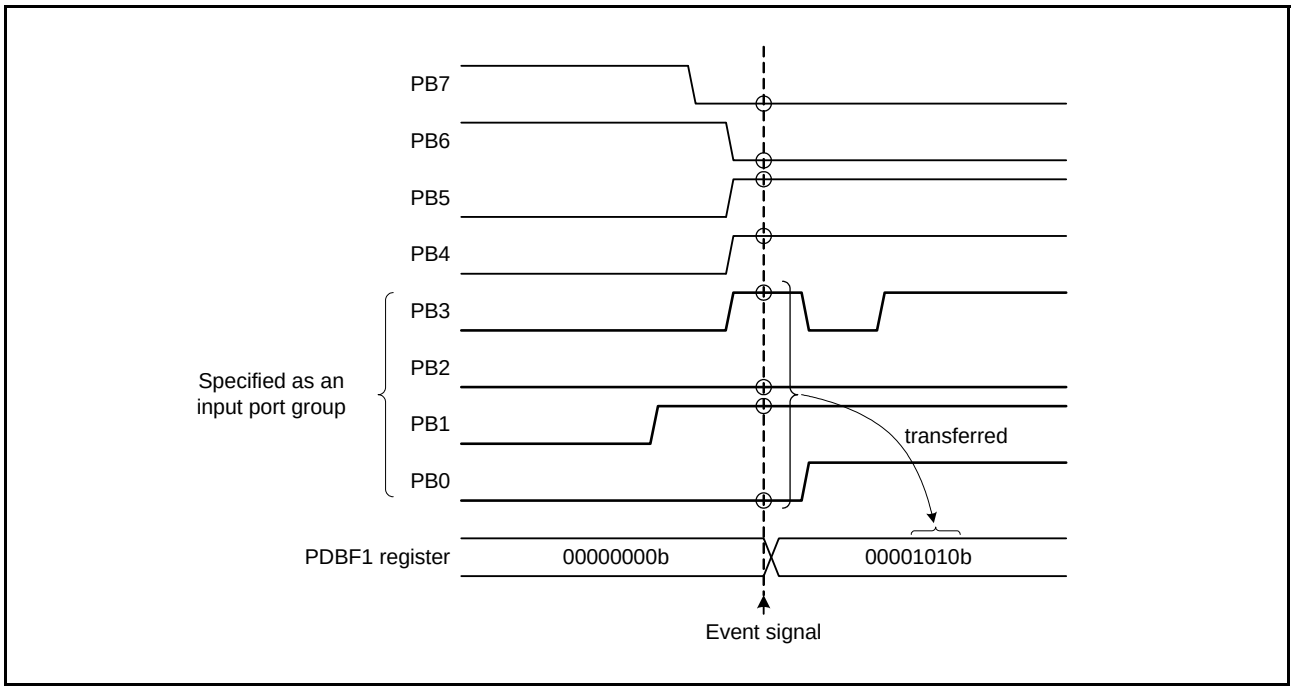


Figure 21.4 Event Linkage Related to Input Port Groups (Port B)

(6) Output Port Group Operation When Event Signal is Input

When an event signal is input to an output port group, the value of the corresponding PODR register changes according to a setting of the PGCn.PGCO[2:0] bits (n = 1, 2). An example of operation is shown in Figure 21.5.

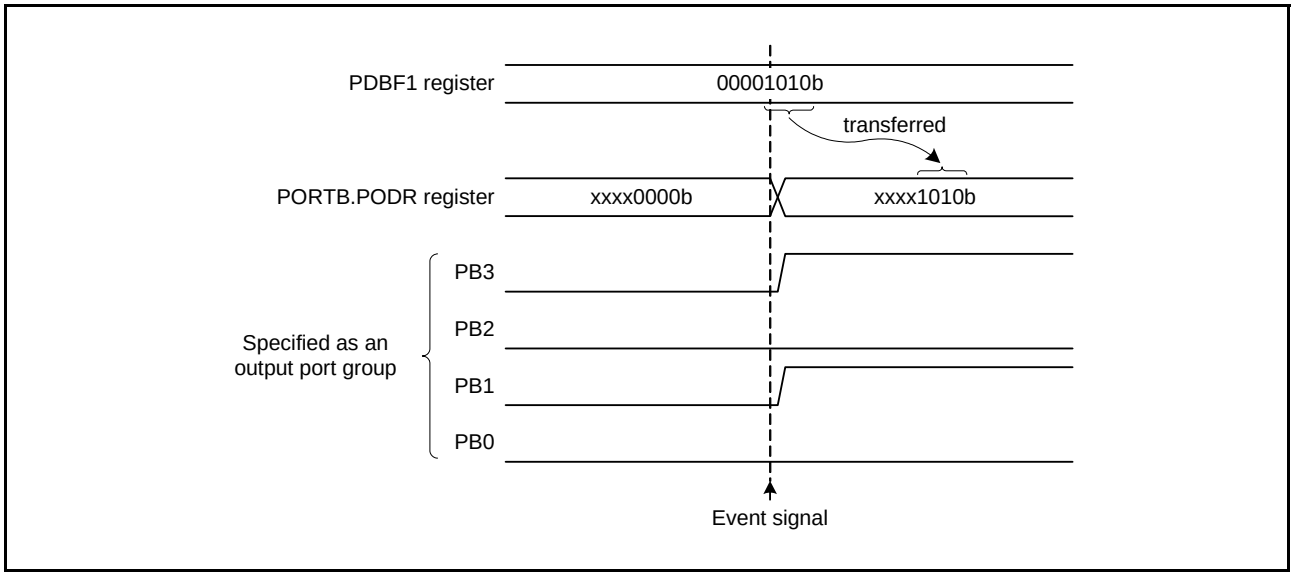


Figure 21.5 Event Linkage Related to Output Port Groups (Port B)

(7) Operation of the PDBFn Registers

(a) Input Port Groups

When an event signal is input to an input port group, the level of the corresponding pins is transferred to the PDBFn register (n = 1, 2). When another event signal is input to the input port group in this condition, different operations are performed depending on the PGCn.PGCOVE bit setting described as below.

- When the PGCn.PGCOVE bit is 0 (overwriting is disabled)
  When the value transferred to the PDBFn register after an input of the last event signal has already been read by the CPU or DTC, the level of the corresponding pins at the time is transferred to the PDBFn register. When the value has not been read, the level of the pins is not transferred to the PDBFn register, and the input event signal is ignored.
- When the PGCn.PGCOVE bit is 1 (overwriting is enabled)
  When another event signal is input to the input port group, the level of the corresponding pins is transferred to the PDBFn register.

(b) Output Port Groups

When an output port group is specified to output the PDBFn register value (PGCn.PGCO[2:0] bits = 011b), the PDBFn register value is transferred to the PODR register following an input of an event signal to the output port group. Data is not transferred to the bits corresponding to the ports that are not specified as members of the output port group.
When output data is specified to rotate in an output port group (PGCn.PGCO[2:0] bits = 1xxb), the data is transferred from the PDBFn register to the PODR register at first event signal, and the PODR register value is rotated from MSB to LSB within the relevant group at second and subsequent signals.
Examples of operation are shown in Figure 21.6.

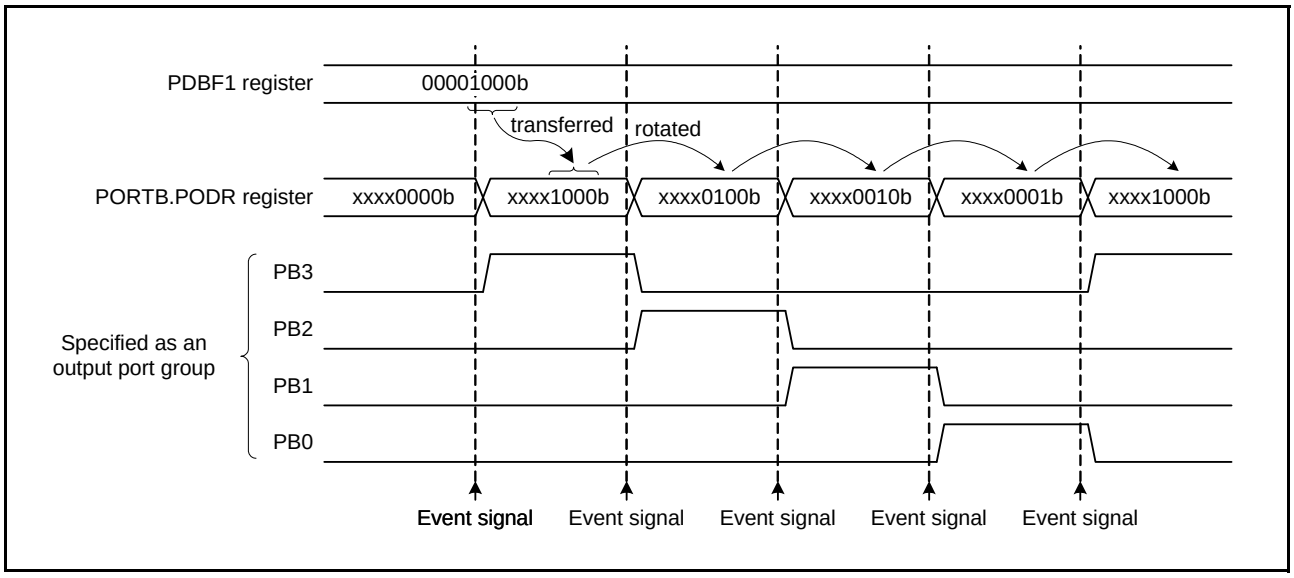


Figure 21.6 Bit-Rotating Operation of Output Port Groups (Port B)

(8) Restrictions on Writing to PODR and PDBF Registers

When the ELCR.ELCON bit is 1 (ELC function is enabled), write access to the PODR and PDBFn registers (n = 1, 2) becomes disabled at the following conditions.

- When a port is specified as a member of the input port group and when the event linkage is set, write access to the corresponding bit in the PDBFn register becomes disabled.
- When a port is specified as a member of the output port group, write access to the corresponding bit in the PODR register becomes disabled.
- When a port is specified as a single output port and when the event linkage for the port is set by the ELSRn register, write access to the corresponding bit in the PODR register becomes disabled.

21.3.7 Example of Procedure for Linking Events

The following describes the procedure for linking events.

(1) Initialize the peripheral module (destination) that operates based on an event signal.
(2) When event linkage is set to a port, set the following registers corresponding to the port.
PODR register: Set the initial values of the output ports.
PDR register: Set the I/O direction of the ports.
PGRn register: To operate ports for a port group, select ports to be specified as port group members (n = 1, 2).
PGCn register: Set the operation of the port group.
PELm register: When a port is operated as a single port, specify the port to be used, an operation of the port at an input of event signal, and the event generation condition (m = 0 to 3).
(3) Set the number of the event signal to the ELSRn register corresponding to the destination peripheral module.
(4) To link an event to a timer module, set any of the ELOPA to ELOPD, ELOPF, and ELOPH registers corresponding to the timer as required.
(5) Set the ELCR.ELCON bit to 1, which enables linkage of all the events.
(6) Set the operation of the peripheral module (source) from which an event signal is output, and activate the module. The preset operation of the destination peripheral module is started by the event signal that is output from the source peripheral module.
(7) To stop event linkage of independent peripheral module, set 00h to the ELSRn register corresponding to the peripheral module. To stop linkage of all the events, set the ELCR.ELCON bit to 0.

Note: If event signal output from the RTC is to be used, make the ELC settings after the RTC settings (initialization, time setting, etc.). Unintended events may be generated if RTC settings are made after the ELC settings.
Note: When using event signal output from the LVD, set the LVD and then the ELC. Set the corresponding ELSRn register to 00h and then disable the LVD.

21.4 Usage Notes

21.4.1 Setting ELSRn Register

(1) Setting ELSR18 and ELSR19 Registers

Specify an event number from among 63h to 6Ah. Do not set the value other than preceding numbers.

(2) Setting ELSR24, ELSR25, ELSR26, and ELSR27 Registers

Do not set the DOC data operation condition met signal (6Ah).

21.4.2 Setting Bit-Rotating Operation of Output Port Groups

When the values of the PDBFn register (n = 1, 2) are changed in the bit-rotating operation mode of the output port group, set the ELSRn register again. Set intervals for generating the event as at least one PCLKB cycle when using it for bit-rotating operation.

21.4.3 Linking DMA/DTC Transfer End Signal as Event

When linking the DMA/DTC transfer end signal as an event signal, do not set the same peripheral module as the DMA/DTC transfer destination and event link destination. If set, the peripheral module might be started before DMA/DTC transfer to the peripheral module is completed.

21.4.4 Clock Settings

To link events, make sure that the ELC and the related peripheral modules are in an operational condition. The peripheral modules cannot operate if they are in the module stop state or in mode which they stop (all-module clock stop mode, software standby mode, or deep software standby mode).

21.4.5 Module Stop Function Setting

ELC operation can be disabled or enabled using module stop control register B (MSTPCRB). After reset is released, the ELC function is disabled. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

22. I/O Ports

22.1 Overview

The pins of an I/O port function as general I/O port pins, I/O pins for peripheral modules, interrupt input pins, or bus control pins.

Each pin is also configurable as an I/O pin of a peripheral module or an input pin for an interrupt. All pins function as input pins immediately after a reset, and pin functions are switched by register settings. The setting of each pin is specified by the registers for the corresponding I/O port and peripheral modules.

Each port has the port direction register (PDR) that selects input or output direction, the port output data register (PODR) that holds data for output, the port input register (PIDR) that indicates the pin states, the open-drain control register y (ODRy, y = 0, 1) that selects the output type of each pin, the pull-up resistor control register (PCR) that controls on/off of the input pull-up resistors, the drive capacity control register (DSCR, DSCR2) that selects the drive capacity, and the port mode register (PMR) that specifies the pin function of each port.

For details on PMR, refer to section 23, Multi-Function Pin Controller (MPC).

The configuration of the I/O ports differs depending on the package. Table 22.1 shows the specifications of I/O ports, Table 22.2 lists the port functions.

Table 22.1 Specifications of I/O Ports

Port	Package 145 or 144 Pins	Number of Pin	Package 100 Pins	Number of Pin	Package 64 Pins	Number of Pin	Package 48 Pins	Number of Pin
PORT0	P00 to P03, P05, P07	6	P05, P07	2	P05*1	1	Not provided	0
PORT1	P12 to P17	6	P12 to P17	6	P12, P13, P16, P17	4	P12, P13, P16, P17	4
PORT2	P20 to P27	8	P20 to P27	8	P26, P27	2	P26, P27	2
PORT3	P30 to P37	8	P30 to P37	8	P30, P31, P34 to P37	6	P30, P31, P34 to P37	6
PORT4	P40 to P47	8	P40 to P47	8	P40 to P43	4	P40 to P43	4
PORT5	P50 to P56	7	P50 to P55	6	P53	1	P53	1
PORT6	P60 to P67	8	Not provided	0	Not provided	0	Not provided	0
PORT7	P70 to P77*2	8	Not provided	0	Not provided	0	Not provided	0
PORT8	P80 to P83, P86, P87	6	Not provided	0	Not provided	0	Not provided	0
PORT9	P90 to P93	4	Not provided	0	Not provided	0	Not provided	0
PORTA	PA0 to PA7	8	PA0 to PA7	8	PA1, PA2, PA4, PA6, PA7	5	PA1, PA2, PA4, PA6	4
PORTB	PB0 to PB7	8	PB0 to PB7	8	PB5 to PB7	3	PB5 to PB7	3
PORTC	PC0 to PC7	8	PC0 to PC7	8	PC0, PC1, PC4 to PC7	6	PC4 to PC7	4
PORTD	PD0 to PD7	8	PD0 to PD7	8	PD2 to PD7	6	PD2 to PD5	4
PORTE	PE0 to PE7	8	PE0 to PE7	8	PE0 to PE2, PE6, PE7	5	PE6, PE7	2
PORTF	PF5	1	Not provided	0	Not provided	0	Not provided	0
PORTH	PH1, PH2	2	PH1, PH2	2	PH1, PH2	2	Not provided	0
PORTJ	PJ3, PJ5	2	PJ3	1	Not provided	0	Not provided	0
	Total of pins	114	Total of pins	81	Total of pins	45	Total of pins	34

Note 1. Not provided on the 64-pin TFBGA.
Note 2. The 145-pin TFLGA (0.65-mm pitch) product does not have the P71 and P72 pins.

Table 22.2 Port Functions (1/2)

Port	Pin	Input Pull-up	Open-Drain Output	Driving Ability Switching	5-V Tolerant
PORT0	P00 to P02	✓	✓	Normal drive/high drive/high-speed interface high-drive	—
	P03, P05	✓	✓	Fixed to high driving ability output	—
	P07	✓	✓	Fixed to high driving ability output	✓
PORT1	P12 to P14	✓	✓	Normal drive/high drive/high-speed interface high-drive	✓
	P15, P16	✓	✓	Fixed to high driving ability output	✓
	P17	✓	✓	High drive/high-speed interface high-drive	✓
PORT2	P20, P21	✓	✓	High drive/high-speed interface high-drive	✓
	P22, P23	✓	✓	High drive/high-speed interface high-drive	—
	P24 to P26	✓	✓	Fixed to high driving ability output	—
	P27	✓	✓	Normal drive/high drive/high-speed interface high-drive	—
PORT3	P30, P31	✓	✓	High drive/high-speed interface high-drive	✓*1
	P32, P33	✓	✓	Fixed to high driving ability output	✓*1
	P34, P37	✓	✓	Fixed to high driving ability output	—
	P35	—	—	—	—
	P36	✓	✓	Fixed to normal output	—
PORT4	P40 to P47	✓	✓	Fixed to normal output	—
PORT5	P50 to P52, P54 to P56	✓	✓	Normal drive/high drive/high-speed interface high-drive	—
	P53	✓	✓	High drive/high-speed interface high-drive	—
PORT6	P60 to P66	✓	✓	Fixed to high driving ability output	—
	P67	✓	✓	Fixed to high driving ability output	✓
PORT7	P70	✓	✓	High drive/high-speed interface high-drive	—
	P71	✓	✓	Fixed to high driving ability output	—
	P72, P74 to P77	✓	✓	Normal drive/high drive/high-speed interface high-drive	—
	P73	✓	✓	High drive/high-speed interface high-drive	✓
PORT8	P80 to P83	✓	✓	Normal drive/high drive/high-speed interface high-drive	—
	P86, P87	✓	✓	High drive/high-speed interface high-drive	—
PORT9	P90 to P93	✓	✓	Normal drive/high drive/high-speed interface high-drive	—
PORTA	PA0 to PA7	✓	✓	Normal drive/high drive/high-speed interface high-drive	—
PORTB	PB0 to PB7	✓	✓	Normal drive/high drive/high-speed interface high-drive	—
PORTC	PC0 to PC3	✓	✓	Normal drive/high drive/high-speed interface high-drive	✓
	PC4 to PC7	✓	✓	Normal drive/high drive/high-speed interface high-drive	—

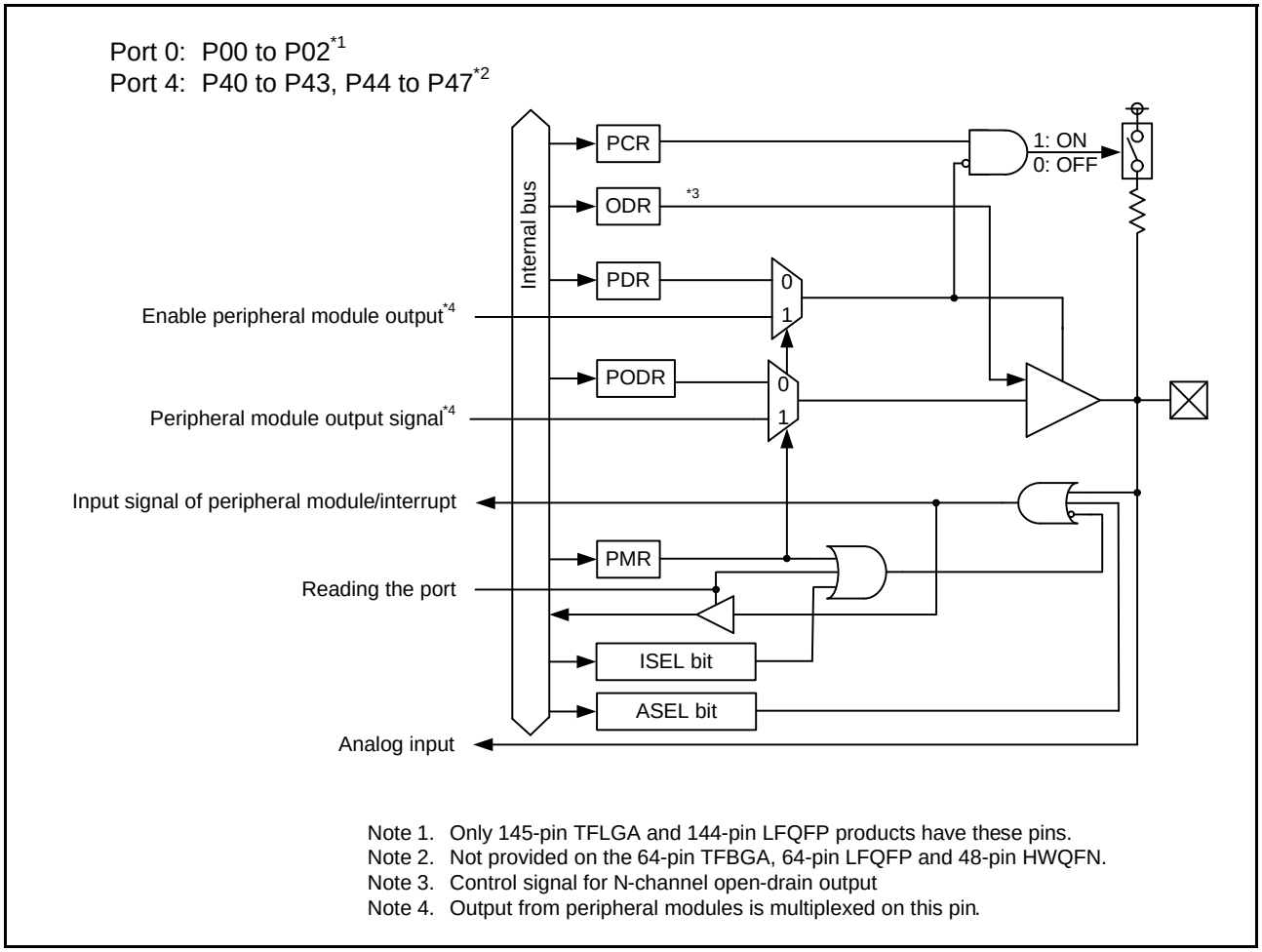
Table 22.2 Port Functions (2/2)

Port	Pin	Input Pull-up	Open-Drain Output	Driving Ability Switching	5-V Tolerant
PORTD	PD0 to PD7	✓	✓	Normal drive/high drive/high-speed interface high-drive	—
PORTE	PE0 to PE7	✓	✓	Normal drive/high drive/high-speed interface high-drive	—
PORTF	PF5	✓	✓	Fixed to high driving ability output	—
PORTH	PH1, PH2	✓	✓	Normal drive/high drive/high-speed interface high-drive	—
PORTJ	PJ3	✓	✓	Fixed to high driving ability output	✓*1
	PJ5	✓	✓	Fixed to high driving ability output	—

Note 1. Note that pins P30, P31, and P32 are not 5-V tolerant when they are set for the TAMPIn or RTCICn (n = 0 to 2) functions. The same applies to pin PJ3 when it is set for the EXCIN function.

Specifying input pull-up, open-drain output, switching of driving ability, or 5-V tolerance is available for other signals on pins that also function as general I/O pins.

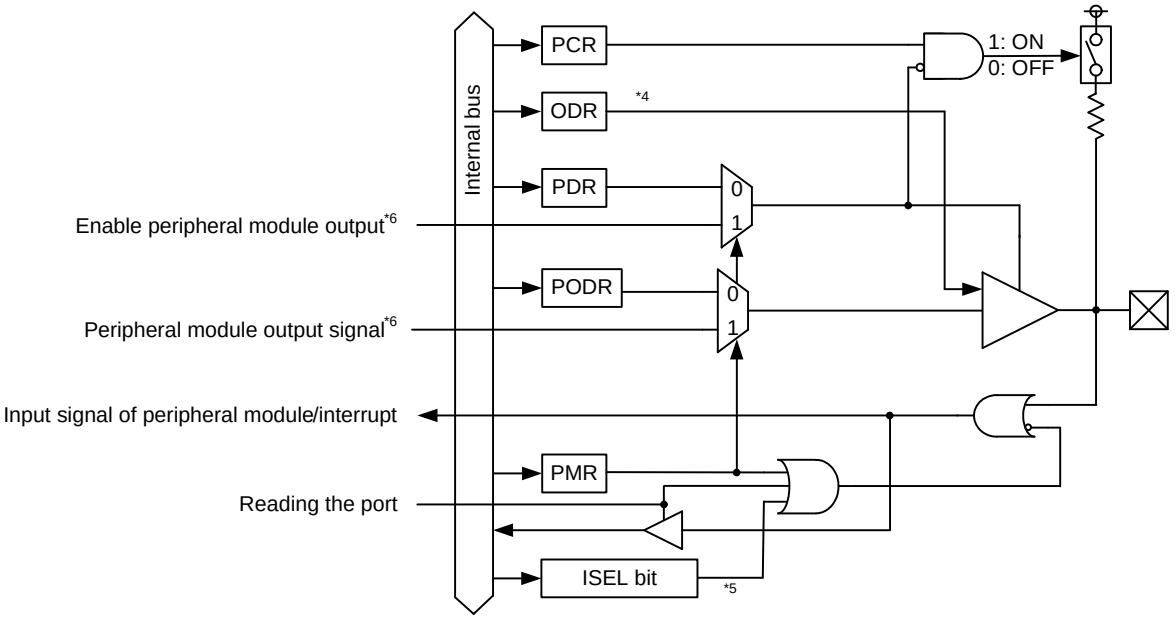
22.2 I/O Port Configuration



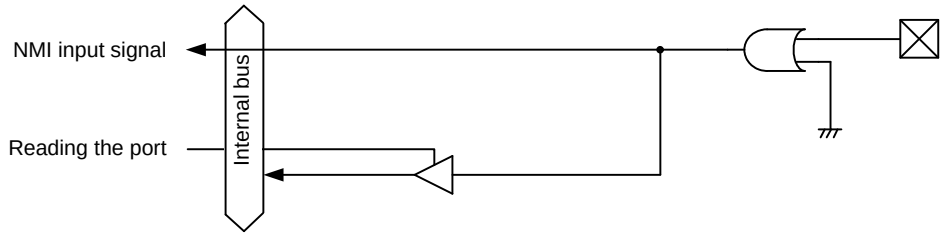
Note 1. Only 145-pin TFLGA and 144-pin LFQFP products have these pins.
Note 2. Not provided on the 64-pin TFBGA, 64-pin LFQFP and 48-pin HWQFN.
Note 3. Control signal for N-channel open-drain output
Note 4. Output from peripheral modules is multiplexed on this pin.

Figure 22.1 I/O Port Configuration (1)

Port 0: P03[*1], P05[*2], P07[*3]
Port 1: P12, P13, P14[*3], P15[*3], P16, P17
Port 2: P20 to P23[*3]
Port 3: P33[*3], P34, P36, P37
Port 5: P56[*1]
Port 8: P80 to P83[*1], P86[*1], P87[*1]
Port F: PF5[*1]
Port J: PJ5[*1]



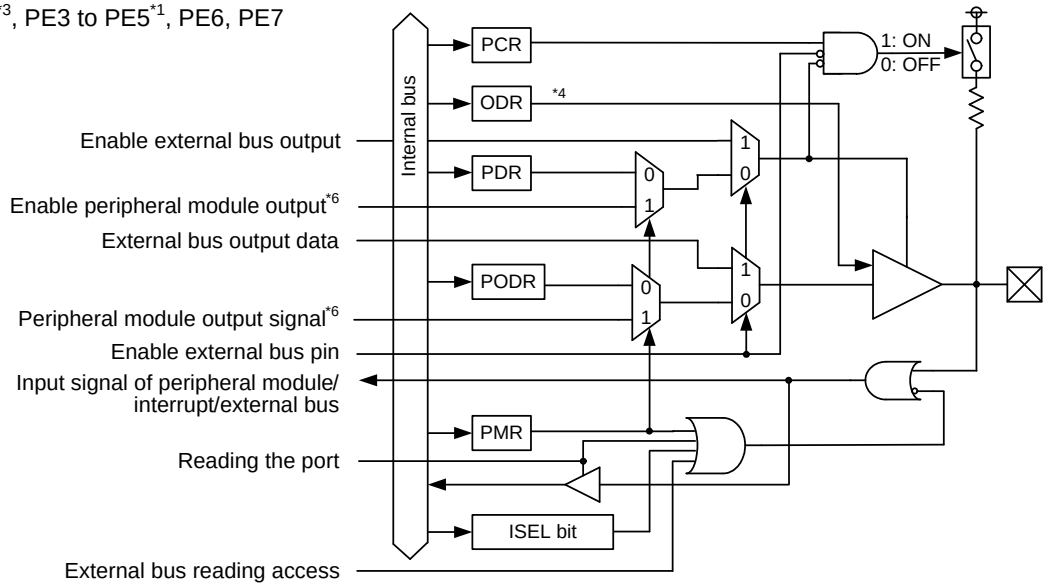
Port 3: P35



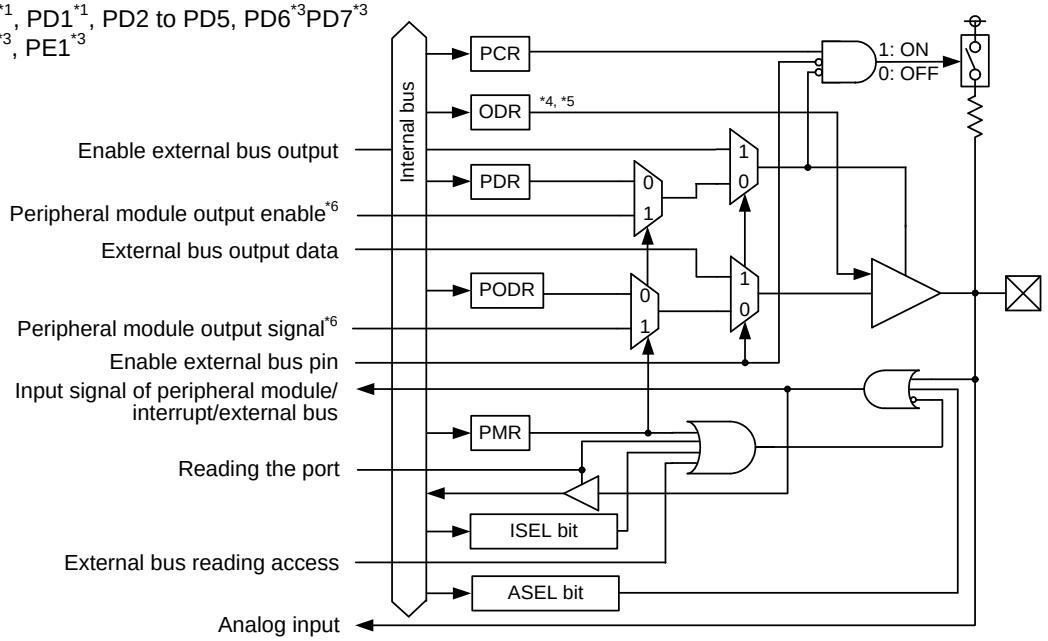
Note 1. Only 145-pin TFLGA and 144-pin LFQFP products have these pins.
Note 2. Not provided on the 64-pin TFBGA and 48-pin HWQFN.
Note 3. Not provided on the 64-pin TFBGA, 64-pin LFQFP and 48-pin HWQFN.
Note 4. Control signal for N-channel open-drain output
Note 5. An external interrupt function is multiplexed on this pin.
Note 6. Output from peripheral modules is multiplexed on this pin.

Figure 22.2 I/O Port Configuration (2)

Port 2: P24[*1], P25[*1], P26, P27
Port 5: P50 to P52[*1], P54[*1], P55[*1]
Port 6: P60 to P67[*2]
Port 7: P70 to P77[*2]
Port 9: P91 to P93[*2]
Port A: PA0[*1], PA1, PA2, PA3[*1], PA4, PA5[*1], PA6, PA7[*3]
Port B: PB0 to PB2[*1], PB4[*1], PB5 to PB7
Port C: PC0[*3], PC1[*3], PC2[*1], PC4 to PC7
Port E: PE2[*3], PE3 to PE5[*1], PE6, PE7

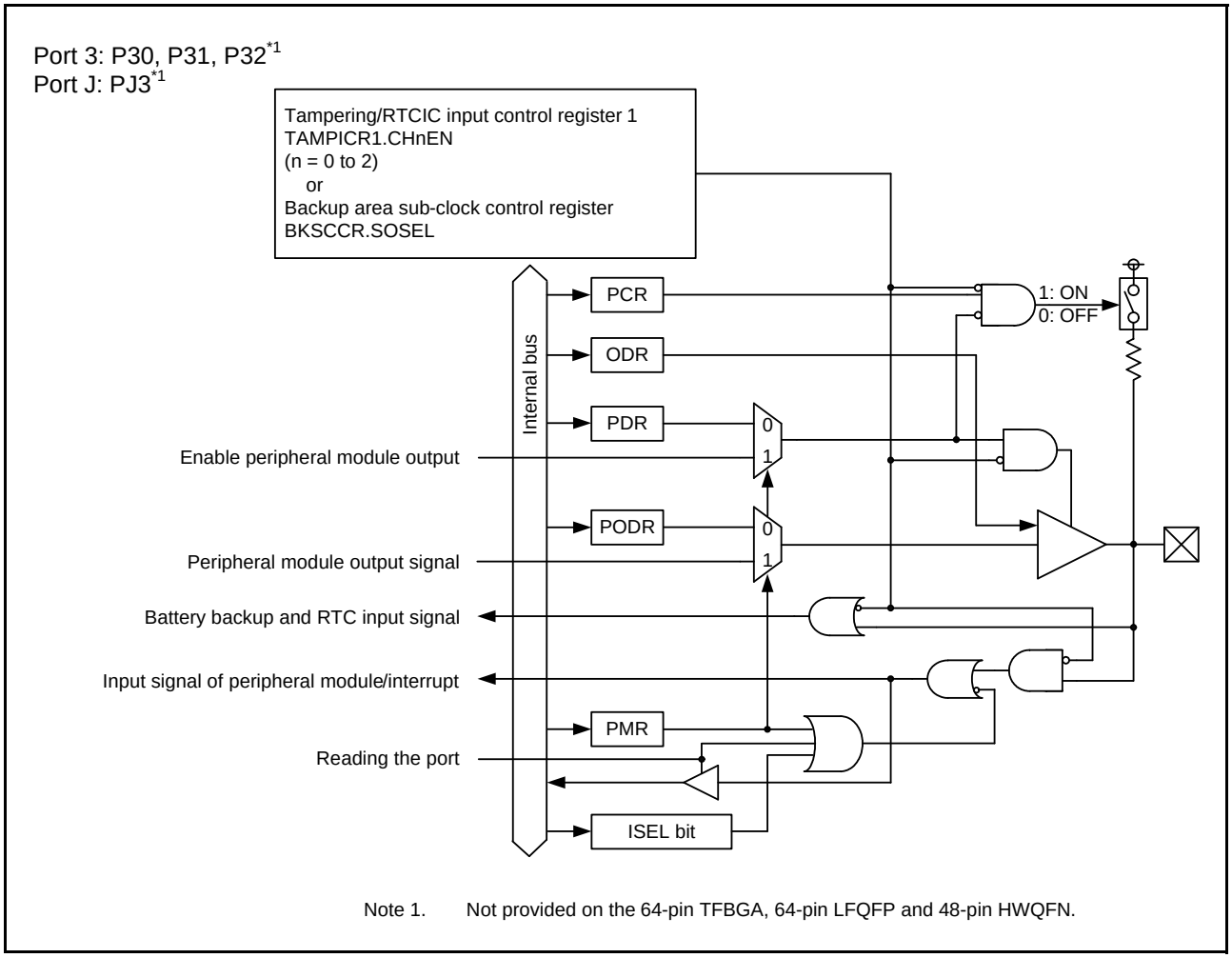


Port 9: P90[*2]
Port D: PD0[*1], PD1[*1], PD2 to PD5, PD6[*3]PD7[*3]
Port E: PE0[*3], PE1[*3]



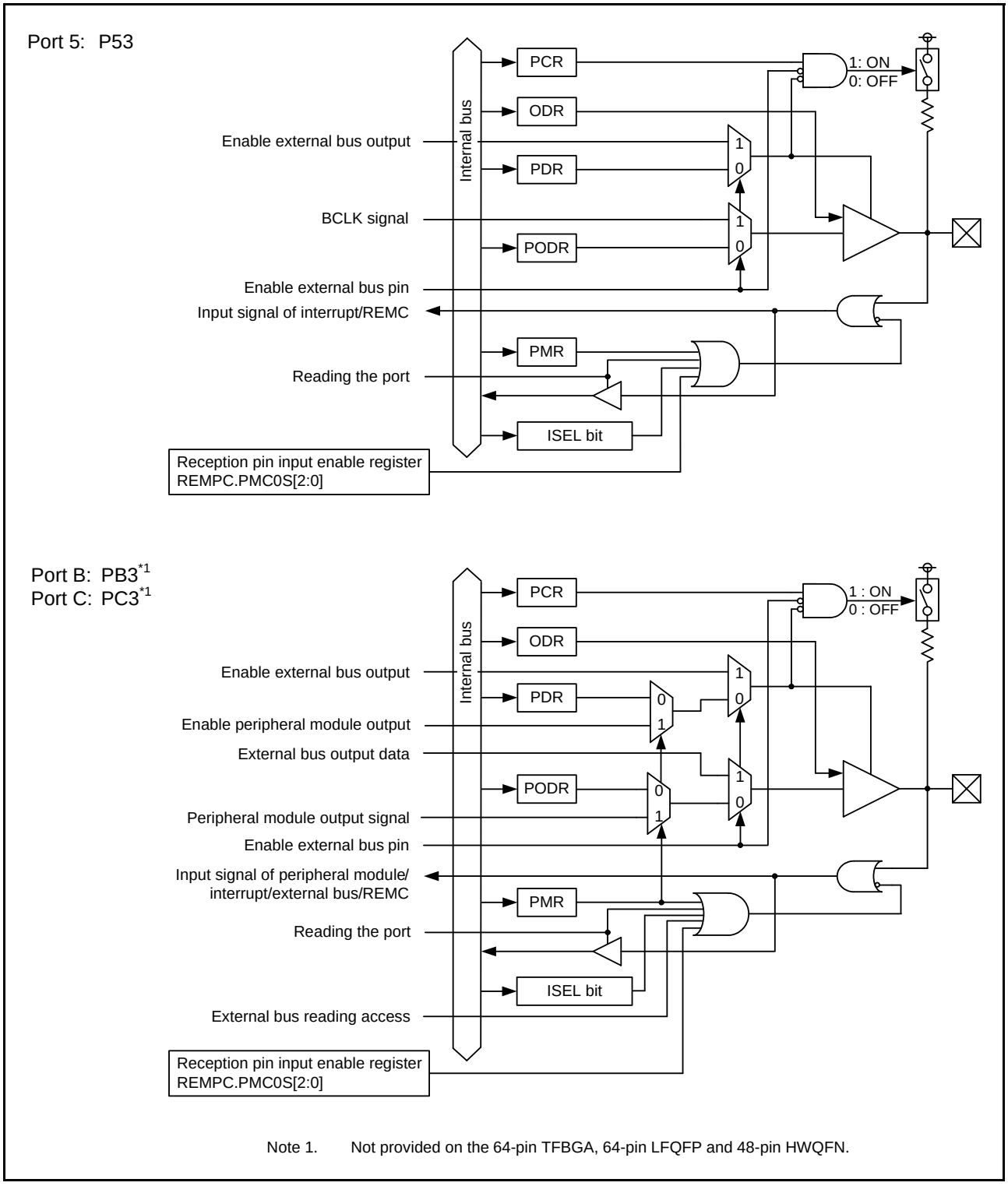
Note 1. Not provided on the 64-pin TFBGA, 64-pin LFQFP and 48-pin HWQFN.
Note 2. Only 145-pin TFLGA and 144-pin LFQFP products have these pins.
Note 3. Not provided on the 48-pin HWQFN.
Note 4. Control signal for N-channel open-drain output
Note 5. Control signal for P-channel open-drain output (only for PE1)
Note 6. Output from peripheral modules is multiplexed on this pin.

Figure 22.3 I/O Port Configuration (3)



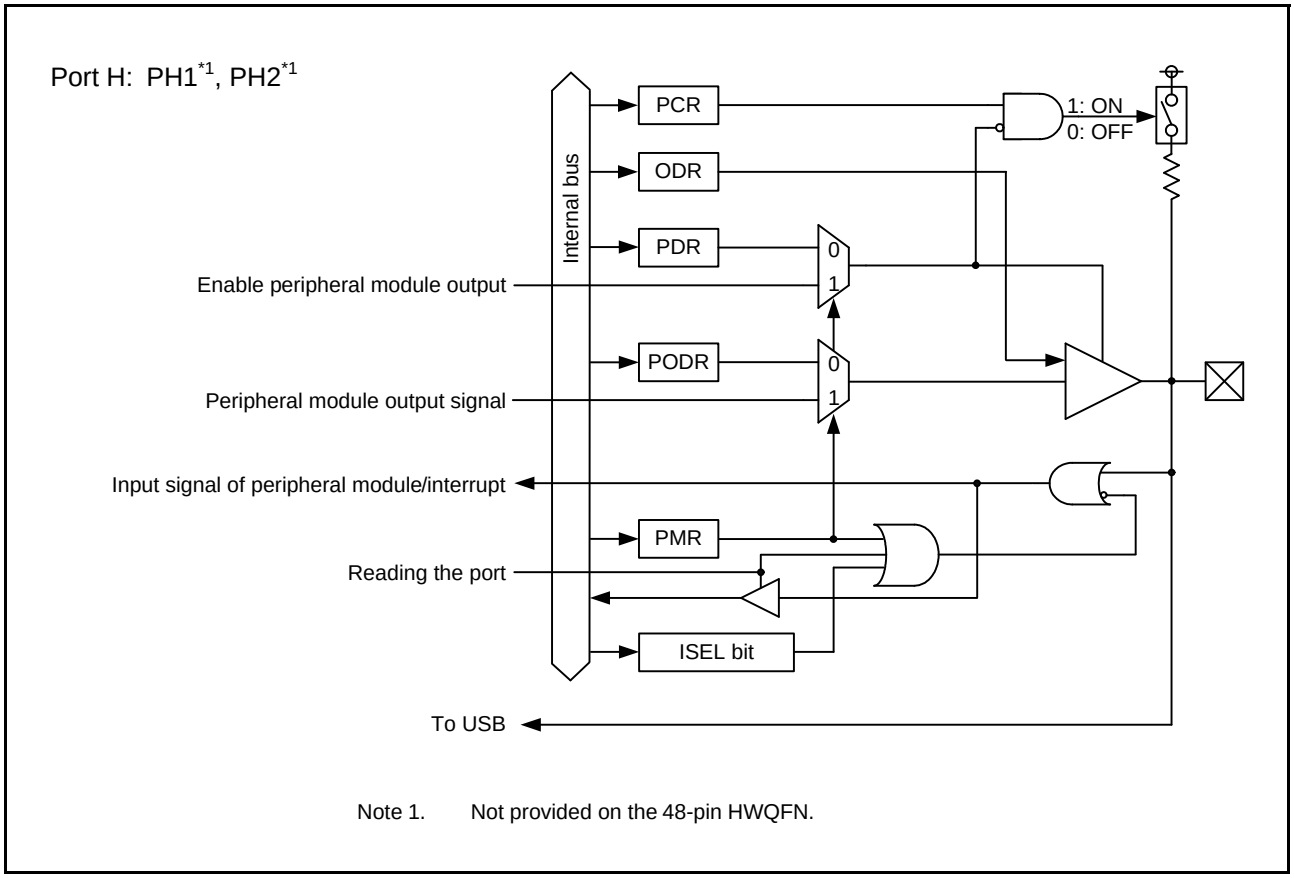
Note 1. Not provided on the 64-pin TFBGA, 64-pin LFQFP and 48-pin HWQFN.

Figure 22.4 I/O Port Configuration (4)



Note 1. Not provided on the 64-pin TFBGA, 64-pin LFQFP and 48-pin HWQFN.

Figure 22.5 I/O Port Configuration (5)



Note 1. Not provided on the 48-pin HWQFN.

Figure 22.6 I/O Port Configuration (6)

22.3 Register Descriptions

22.3.1 Port Direction Register (PDR)

Address(es): PORT0.PDR 0008 C000h, PORT1.PDR 0008 C001h, PORT2.PDR 0008 C002h, PORT3.PDR 0008 C003h, PORT4.PDR 0008 C004h, PORT5.PDR 0008 C005h, PORT6.PDR 0008 C006h, PORT7.PDR 0008 C007h, PORT8.PDR 0008 C008h, PORT9.PDR 0008 C009h, PORTA.PDR 0008 C00Ah, PORTB.PDR 0008 C00Bh, PORTC.PDR 0008 C00Ch, PORTD.PDR 0008 C00Dh, PORTE.PDR 0008 C00Eh, PORTF.PDR 0008 C00Fh, PORTH.PDR 0008 C011h, 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 F, H, and J

PDR is a register which is used to select the input or output direction for individual pins of the corresponding port when the pins are configured as the general I/O pins.
Each bit of PORTm.PDR corresponds to each pin of port m; I/O direction can be specified in 1-bit units.
However, the bits that correspond to port m on the 144-pin product but do not exist on a product with fewer pins are reserved. When writing, write 1 (output) to these bits.
Each bit of PDR corresponding to port m that does not exist is reserved. Make settings according to the description in section 22.4, Initialization of the Port Direction Register (PDR).
The B5 bit in PORT3.PDR is reserved, because the P35 pin is input only.

22.3.2 Port Output Data Register (PODR)

Address(es): PORT0.PODR 0008 C020h, PORT1.PODR 0008 C021h, PORT2.PODR 0008 C022h, PORT3.PODR 0008 C023h, PORT4.PODR 0008 C024h, PORT5.PODR 0008 C025h, PORT6.PODR 0008 C026h, PORT7.PODR 0008 C027h, PORT8.PODR 0008 C028h, PORT9.PODR 0008 C029h, PORTA.PODR 0008 C02Ah, PORTB.PODR 0008 C02Bh, PORTC.PODR 0008 C02Ch, PORTD.PODR 0008 C02Dh, PORTE.PODR 0008 C02Eh, PORTF.PODR 0008 C02Fh, PORTH.PODR 0008 C031h, 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 F, H, 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 144-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.PODR is reserved, because the P35 pin is input only. Data is not output from the corresponding pins even if these bits are set.
The bit corresponding to a pin that does not exist is reserved. A reserved bit is always read as 0. The write value should always be 0.

22.3.3 Port Input Register (PIDR)

Address(es): PORT0.PIDR 0008 C040h, PORT1.PIDR 0008 C041h, PORT2.PIDR 0008 C042h, PORT3.PIDR 0008 C043h, PORT4.PIDR 0008 C044h, PORT5.PIDR 0008 C045h, PORT6.PIDR 0008 C046h, PORT7.PIDR 0008 C047h, PORT8.PIDR 0008 C048h, PORT9.PIDR 0008 C049h, PORTA.PIDR 0008 C04Ah, PORTB.PIDR 0008 C04Bh, PORTC.PIDR 0008 C04Ch, PORTD.PIDR 0008 C04Dh, PORTE.PIDR 0008 C04Eh, PORTF.PIDR 0008 C04Fh, PORTH.PIDR 0008 C051h, 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 F, H, and J

PIDR is a register which reflects individual pin states of the port.

The pin states of port m can be read with the PORTm.PIDR, regardless of the values of PORTm.PDR and PORTm.PMR.

The NMI pin state is reflected in the P35 bit. However, the states of pins when the PmnPFS.ASEL bit is set to 1 cannot be read.

The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as undefined, and cannot be modified.

22.3.4 Port Mode Register (PMR)

Address(es): PORT0.PMR 0008 C060h, PORT1.PMR 0008 C061h, PORT2.PMR 0008 C062h, PORT3.PMR 0008 C063h, PORT4.PMR 0008 C064h, PORT5.PMR 0008 C065h, PORT6.PMR 0008 C066h, PORT7.PMR 0008 C067h, PORT8.PMR 0008 C068h, PORT9.PMR 0008 C069h, PORTA.PMR 0008 C06Ah, PORTB.PMR 0008 C06Bh, PORTC.PMR 0008 C06Ch, PORTD.PMR 0008 C06Dh, PORTE.PMR 0008 C06Eh, PORTF.PMR 0008 C06Fh, PORTH.PMR 0008 C071h, PORTJ.PMR 0008 C072h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Pin Mode Control	0: Uses the pin as a general I/O pin. 1: Uses the pin as an I/O port for peripheral modules.	R/W
b1	B1	Pm1 Pin Mode Control		R/W
b2	B2	Pm2 Pin Mode Control		R/W
b3	B3	Pm3 Pin Mode Control		R/W
b4	B4	Pm4 Pin Mode Control		R/W
b5	B5	Pm5 Pin Mode Control		R/W
b6	B6	Pm6 Pin Mode Control		R/W
b7	B7	Pm7 Pin Mode Control		R/W

m = 0 to 9, A to F, H, 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 144-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.

22.3.5 Open-Drain Control Register 0 (ODR0)

Address(es): PORT0.ODR0 0008 C080h, PORT1.ODR0 0008 C082h, PORT2.ODR0 0008 C084h, PORT3.ODR0 0008 C086h, PORT4.ODR0 0008 C088h, PORT5.ODR0 0008 C08Ah, PORT6.ODR0 0008 C08Ch, PORT7.ODR0 0008 C08Eh, PORT8.ODR0 0008 C090h, PORT9.ODR0 0008 C092h, PORTA.ODR0 0008 C094h, PORTB.ODR0 0008 C096h, PORTC.ODR0 0008 C098h, PORTD.ODR0 0008 C09Ah, PORTE.ODR0 0008 C09Ch, PORTH.ODR0 0008 C0A2h, PORTJ.ODR0 0008 C0A4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	B6	—	B4	B3	B2	—	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Output Type Select	For pins other than the port PE1 pin	R/W
b1	—	Reserved	Odd bit / Even bit	R/W
b2	B2	Pm1 Output Type Select	X 0: CMOS output	R/W
b3	B3*1	PE1 Output Type Select	X 1: N-channel open-drain output	R/W
b4	B4	Pm2 Output Type Select	(b1, b3, b5, b7: Reserved)	R/W
b5	—	Reserved	For port PE1 pin	R/W
b6	B6	Pm3 Output Type Select	b3 b2	R/W
b7	—	Reserved	0 0: CMOS output	R/W
			0 1: N-channel open-drain output	
			1 0: PMOS open-drain output	
			1 1: Setting prohibited	

m = 0 to 9, A to E, H, and J

Note 1. The bit for a pin other than PE1 is reserved.

ODR0 is a register which is used to select an output type for the pins of the port.

In the registers other than PORTE.ODR0, the odd bits (b1, b3, b5, and b7) are reserved.

However, the output type of the port PE1 pin is specified by the combination of b3 and b2.

Bits that correspond to port m on the 144-pin product but do not exist on a product with fewer pins are reserved. When writing, write 0 (CMOS output) to these bits.

The bit corresponding to a pin that does not exist is also reserved. A reserved bit is always read as 0. The write value should always be 0.

22.3.6 Open-Drain Control Register 1 (ODR1)

Address(es): PORT0.ODR1 0008 C081h, PORT1.ODR1 0008 C083h, PORT2.ODR1 0008 C085h, PORT3.ODR1 0008 C087h, PORT4.ODR1 0008 C089h, PORT5.ODR1 0008 C08Bh, PORT6.ODR1 0008 C08Dh, PORT7.ODR1 0008 C08Fh, PORT8.ODR1 0008 C091h, PORTA.ODR1 0008 C095h, PORTB.ODR1 0008 C097h, PORTC.ODR1 0008 C099h, PORTD.ODR1 0008 C09Bh, PORTE.ODR1 0008 C09Dh, PORTF.ODR1 0008 C09Fh, 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: N-channel 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 8, A to F, and J

ODR1 is used to select an output type for each pin of the port.

The odd bits (b1, b3, b5, and b7) in the ODR1 register are reserved.

Bits that correspond to port m on the 144-pin product but do not exist on a product with fewer pins are reserved. When writing, write 0 (CMOS output) to these bits.

The bit corresponding to a pin that does not exist is also reserved. A reserved bit is always read as 0. The write value should always be 0.

22.3.7 Pull-Up Resistor Control Register (PCR)

Address(es): PORT0.PCR 0008 C0C0h, PORT1.PCR 0008 C0C1h, PORT2.PCR 0008 C0C2h, PORT3.PCR 0008 C0C3h, PORT4.PCR 0008 C0C4h, PORT5.PCR 0008 C0C5h, PORT6.PCR 0008 C0C6h, PORT7.PCR 0008 C0C7h, PORT8.PCR 0008 C0C8h, PORT9.PCR 0008 C0C9h, PORTA.PCR 0008 C0CAh, PORTB.PCR 0008 C0CBh, PORTC.PCR 0008 C0CCh, PORTD.PCR 0008 C0CDh, PORTE.PCR 0008 C0CEh, PORTF.PCR 0008 C0CFh, PORTH.PCR 0008 C0D1h, 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 F, H, and 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 other than the WAIT pin, a general port output pin, or a peripheral module output pin, clear the corresponding bit to 0.
However, when a pin is used as an address bus or bus control signal and the MCU transitions to software standby mode or deep software standby mode while the output port enable bit (SBYCR.OPE) in the standby control register is cleared to 0, the value in the PCR register becomes enabled.
The pull-up resistor is also disabled in the reset state.
The other bits are also reserved because they correspond to pins that do not exist is reserved. A reserved bit is always read as 0. The write value should always be 0.

22.3.8 Drive Capacity Control Register (DSCR)

Address(es): PORT0.DSCR 0008 C0E0h, PORT1.DSCR 0008 C0E1h, PORT2.DSCR 0008 C0E2h, PORT5.DSCR 0008 C0E5h, PORT7.DSCR 0008 C0E7h, PORT8.DSCR 0008 C0E8h, PORT9.DSCR 0008 C0E9h, PORTA.DSCR 0008 C0EAh, PORTB.DSCR 0008 C0EBh, PORTC.DSCR 0008 C0ECh, PORTD.DSCR 0008 C0EDh, PORTE.DSCR 0008 C0EEh, PORTH.DSCR 0008 C0F1h

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Drive Capacity Control	0: Normal drive output 1: High-drive output	R/W
b1	B1	Pm1 Drive Capacity Control		R/W
b2	B2	Pm2 Drive Capacity Control		R/W
b3	B3	Pm3 Drive Capacity Control		R/W
b4	B4	Pm4 Drive Capacity Control		R/W
b5	B5	Pm5 Drive Capacity Control		R/W
b6	B6	Pm6 Drive Capacity Control		R/W
b7	B7	Pm7 Drive Capacity Control		R/W

m = 0 to 2, 5, 7 to 9, A to E, and H

DSCR is a register which is used to switch the drive capacity of the port.
When pins are set for high-speed interface high-drive in the DSCR2 register, the drive capacity cannot be changed. For setting of the drive capacity by the DSCR and DSCR2 registers, refer to Table 22.3, Drive Capacity Setting by DSCR and DSCR2 Registers.
The bit corresponding to a pin whose drive capacity cannot be switched, or to high drive output is readable and writable, but the drive capacity cannot be changed.
The bit corresponding to a pin that does not exist is reserved. A reserved bit is always read as 0. The write value should always be 0.

22.3.9 Drive Capacity Control Register 2 (DSCR2)

Address(es): PORT0.DSCR2 0008 C128h, PORT1.DSCR2 0008 C129h, PORT2.DSCR2 0008 C12Ah, PORT3.DSCR2 0008 C12Bh, PORT5.DSCR2 0008 C12Dh, PORT7.DSCR2 0008 C12Fh, PORT8.DSCR2 0008 C130h, PORT9.DSCR2 0008 C131h, PORTA.DSCR2 0008 C132h, PORTB.DSCR2 0008 C133h, PORTC.DSCR2 0008 C134h, PORTD.DSCR2 0008 C135h, PORTE.DSCR2 0008 C136h, PORTH.DSCR2 0008 C139h

	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 2	0: Normal drive/high-drive output*1 1: High-speed interface high-drive output	R/W
b1	B1	Pm1 Drive Capacity Control 2		R/W
b2	B2	Pm2 Drive Capacity Control 2		R/W
b3	B3	Pm3 Drive Capacity Control 2		R/W
b4	B4	Pm4 Drive Capacity Control 2		R/W
b5	B5	Pm5 Drive Capacity Control 2		R/W
b6	B6	Pm6 Drive Capacity Control 2		R/W
b7	B7	Pm7 Drive Capacity Control 2		R/W

m = 0 to 3, 5, 7 to 9, A to E, and H

Note 1. Pins that support switching drive capacity by the DSCR register depend on the setting of the DSCR register.

DSCR2 is a register which is used to switch the drive capacity of the port.

Table 22.3 shows the setting of drive capacity by the DSCR and DSCR2 registers.

The bit corresponding to a pin whose drive capacity cannot be switched, or to high drive output for the high-speed interface is readable and writable, but the drive capacity cannot be changed.

The bit corresponding to a pin that does not exist is reserved. A reserved bit is always read as 0. The write value should always be 0.

Set the register only based on an instruction in the application guide.

Table 22.3 Drive Capacity Setting by DSCR and DSCR2 Registers

PORTm.DSCR2.Bx	PORTm.DSCR.Bx	Drive Capacity*1
0	0	Normal drive output
0	1	High-drive output
1	Don’t care	High-speed interface high-drive output

Note 1. When drive capacity is fixed, or drive capacity of a pin whose drive capacity cannot be switched cannot be changed.

22.4 Initialization of the Port Direction Register (PDR)

Initialize reserved bits in the PDR register according to Table 22.4, Table 22.5, Table 22.6, and Table 22.7.

- The blank columns in Table 22.4, Table 22.5, Table 22.6, and Table 22.7 indicate the bits corresponding to the pins listed in Table 22.1, Specifications of I/O Ports.
  The corresponding bits should be set to 1 (output) or 0 (input) depending on the user system.
  However, the PORT3.PDR.B5 bit of the input-only P35 pin is reserved.
  This bit should be set to 0 (input).
- The columns other than the blank columns in Table 22.4, Table 22.5, Table 22.6, and Table 22.7 indicate reserved bits.
  A reserved bit should be set to 0 (input) or 1 (output) according to Table 22.4, Table 22.5, Table 22.6, and Table 22.7.
  When setting a value to a reserved bit, access in byte units.

Table 22.4 PDR Register Settings in 145-Pin, 144-Pin Packages

	PDR Register							
Port Symbol	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
PORT0		1		1				
PORT1							1	1
PORT2								
PORT3			0					
PORT4								
PORT5	1							
PORT6								
PORT7*1								
PORT8			1	1				
PORT9	1	1	1	1				
PORTA								
PORTB								
PORTC								
PORTD								
PORTE								
PORTF	1	1		1	1	1	1	1
PORTH	1	1	1	1	1			1
PORTJ	1	1		1		1	1	1

Note 1. Since 145-pin TFLGA (0.65-mm pitch) product does not have the P71 and P72 pins, set b1 and b2 to 1.

Table 22.5 PDR Register Settings in 100-Pin Packages

	PDR Register							
Port Symbol	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
PORT0		1		1	1	1	1	1
PORT1							1	1
PORT2								
PORT3			0					
PORT4								
PORT5	1	1						
PORT6	1	1	1	1	1	1	1	1
PORT7	1	1	1	1	1	1	1	1
PORT8	1	1	1	1	1	1	1	1
PORT9	1	1	1	1	1	1	1	1
PORTA								
PORTB								
PORTC								
PORTD								
PORTE								
PORTF	1	1	1	1	1	1	1	1
PORTH	1	1	1	1	1			1
PORTJ	1	1	1	1		1	1	1

Table 22.6 PDR Register Settings in 64-Pin Packages

	PDR Register							
Port Symbol	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
PORT0*1	1	1		1	1	1	1	1
PORT1			1	1			1	1
PORT2			1	1	1	1	1	1
PORT3			0		1	1		
PORT4	1	1	1	1				
PORT5	1	1	1	1		1	1	1
PORT6	1	1	1	1	1	1	1	1
PORT7	1	1	1	1	1	1	1	1
PORT8	1	1	1	1	1	1	1	1
PORT9	1	1	1	1	1	1	1	1
PORTA			1		1			1
PORTB				1	1	1	1	1
PORTC					1	1		
PORTD							1	1
PORTE			1	1	1			
PORTF	1	1	1	1	1	1	1	1
PORTH	1	1	1	1	1			1
PORTJ	1	1	1	1	1	1	1	1

Note 1. Since 64-pin TFBGA product does not have P05 pin, set b5 to 1.

Table 22.7 PDR Register Settings in 48-Pin Packages

	PDR Register							
Port Symbol	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
PORT0	1	1	1	1	1	1	1	1
PORT1			1	1			1	1
PORT2			1	1	1	1	1	1
PORT3			0		1	1		
PORT4	1	1	1	1				
PORT5	1	1	1	1		1	1	1
PORT6	1	1	1	1	1	1	1	1
PORT7	1	1	1	1	1	1	1	1
PORT8	1	1	1	1	1	1	1	1
PORT9	1	1	1	1	1	1	1	1
PORTA	1		1		1			1
PORTB				1	1	1	1	1
PORTC					1	1	1	1
PORTD	1	1					1	1
PORTE			1	1	1	1	1	1
PORTF	1	1	1	1	1	1	1	1
PORTH	1	1	1	1	1	1	1	1
PORTJ	1	1	1	1	1	1	1	1

22.5 Handling of Unused Pins

Details on the handling of unused pins are given in Table 22.8.

Table 22.8 Handling of Unused Pins

Pin Name	Handling
EMLE	Connect this pin to VSS via a resistor (pulling down).
BSCANP	Connect this pin to VSS via a resistor (pulling down).
MD	Use this as a mode pin.
RES#	Connect this pin to VCC via a resistor (pulling up).
VCC_USB	Connect this pin to VCC
VSS_USB	Connect this pin to VSS
VBATT	Connect this pin to VCC
USB1_DP	Keep these pins open.
USB1_DM	
P35/NMI	Connect this pin to VCC via a resistor (pulling up).
P36/EXTAL	Set the MOSCCR.MOSTP bit to 1 (the main clock oscillator is stopped) when not using the main clock When this pin is not used as port P36, handle as port 0 to 9, A to F, and J.
P37/XTAL	Set the MOSCCR.MOSTP bit to 1 (the main clock oscillator is stopped) when not using the main clock When this pin is not used as port P37, handle as port 0 to 9, A to F, and J When an external clock is input to the EXTAL pin, keep the pin open.
PJ3/EXCIN	Set the BKSCCR.SOSEL bit to 0 when this pin is not to be used as EXCIN. When this pin is not used as port PJ3, handle as port 0 to 9, A to F, and J.
PH1/USB0_DP	Set the MSTPB.MSTPB19 bit to 1 when this pin is not to be used as USB0_DP. When this pin is not used as port PH1, handle as port 0 to 9, A to F, and J.
PH2/USB0_DM	Set the MSTPB.MSTPB19 bit to 1 when this pin is not to be used as USB0_DM. When this pin is not used as port PH2, handle as port 0 to 9, A to F, and J.
XCIN	Connect this pin to VSS via a resistor (pulling down).
XCOUT	Keep this pin open.
Port 0 to 9, A to F, and J	• If the direction setting is for input (PORTn.PDR = 0), the corresponding pin is connected to VCC (pulled up) via a resistor or to VSS (pulled down) via a resistor.*1 • If the direction setting is for output (PORTn.PDR = 1), the pin is released.*1, *2
VREFH0	Connect this pin to AVCC0.
VREFL0	Connect this pin to AVSS0.
AVCC0	Connect this pin to VCC when not using the 12-bit A/D converter (unit 0).
AVSS0	Connect this pin to VSS when not using the 12-bit A/D converter (unit 0).
AVCC1	Connect this pin to VCC when not using the 12-bit A/D converter (unit 1).
AVSS1	Connect this pin to VSS when not using the 12-bit A/D converter (unit 1).

Note 1. Clear the PORTn.PMR bit, the PmnPFS.ISEL bit and the PmnPFS.ASEL bit to 0.
Note 2. In the case of release when the setting is for output, the port is an input over the period from release from the reset state to the pin becoming an output. Since the voltage on the pin is undefined while it is an input, this may lead to an increase in the current drawn.

23. Multi-Function Pin Controller (MPC)

23.1 Overview

The multi-function pin controller (MPC) selects and assigns input/output of peripheral functions and interrupt input signals from multiple ports. The MPC also assigns the port of external bus related signals.

Table 23.1 lists the functions assigned to each multiplexed pin. The symbols ✓ and × in the table indicate whether the pin is available or unavailable for the package. Selecting a single function for multiple pins is prohibited.

Table 23.1 Functions Assigned to Each Multiplexed Pin (1/18)

Module/Function	Channel	Pin Functions	Allocation Port	Package 145-pin 144-pin	100-pin	64-pin	48-pin
Interrupt		NMI (input)	P35	✓	✓	✓	✓
EXDMA controller	EXDMAC0	EDREQ0 (input)	P22	✓	✓	×	×
			P55	✓	✓	×	×
			P80	✓	×	×	×
		EDACK0 (output)	P23	✓	✓	×	×
			P54	✓	✓	×	×
			P81	✓	×	×	×
	EXDMAC1	EDREQ1 (input)	P24	✓	✓	×	×
			P33	✓	✓	×	×
			P82	✓	×	×	×
		EDACK1 (output)	P25	✓	✓	×	×
			P56	✓	×	×	×
			P83	✓	×	×	×
			PJ3	✓	✓	×	×
Interrupt	IRQ0	IRQ0-DS (input)	P30	✓	✓	✓	✓
		IRQ0 (input)	P50	✓	✓	×	×
			P60	✓	×	×	×
			P70	✓	×	×	×
			P90	✓	×	×	×
			PA0	✓	✓	×	×
			PD0	✓	✓	×	×
			PH1	✓	✓	✓	×
	IRQ1	IRQ1-DS (input)	P31	✓	✓	✓	✓
		IRQ1 (input)	P51	✓	✓	×	×
			P61	✓	×	×	×
			P71	✓*2	×	×	×
			PD1	✓	✓	×	×
			PH2	✓	✓	✓	×
	IRQ2	IRQ2-DS (input)	P32	✓	✓	×	×
		IRQ2 (input)	P12	✓	✓	✓	✓
			P52	✓	✓	×	×
			P62	✓	×	×	×
			P82	✓	×	×	×
			PB2	✓	✓	×	×
			PD2	✓	✓	✓	✓
	IRQ3	IRQ3-DS (input)	P33	✓	✓	×	×
		IRQ3 (input)	P13	✓	✓	✓	✓
			P23	✓	✓	×	×
			P53	✓	✓	✓	✓
			P63	✓	×	×	×
			P83	✓	×	×	×
			PB3	✓	✓	×	×
			PD3	✓	✓	✓	✓

Table 23.1 Functions Assigned to Each Multiplexed Pin (2/18)

Module/Function	Channel	Pin Functions	Allocation Port	Package 145-pin 144-pin	100-pin	64-pin	48-pin
Interrupt	IRQ4	IRQ4-DS (input)	PB1	✓	✓	×	×
		IRQ4 (input)	P14	✓	✓	×	×
			P34	✓	✓	✓	✓
			P54	✓	✓	×	×
			P64	✓	×	×	×
			PB4	✓	✓	×	×
			PD4	✓	✓	✓	✓
			PF5	✓	×	×	×
	IRQ5	IRQ5-DS (input)	PA4	✓	✓	✓	✓
		IRQ5 (input)	P15	✓	✓	×	×
			P25	✓	✓	×	×
			PA5	✓	✓	×	×
			PC5	✓	✓	✓	✓
			PD5	✓	✓	✓	✓
			PE5	✓	✓	×	×
	IRQ6	IRQ6-DS (input)	PA3	✓	✓	×	×
		IRQ6 (input)	P16	✓	✓	✓	✓
			P26	✓	✓	✓	✓
			P56	✓	×	×	×
			PB6	✓	✓	✓	✓
			PD6	✓	✓	✓	×
			PE6	✓	✓	✓	✓
	IRQ7	IRQ7-DS (input)	PE2	✓	✓	✓	×
		IRQ7 (input)	P17	✓	✓	✓	✓
			P27	✓	✓	✓	✓
			P77	✓	×	×	×
			PA7	✓	✓	✓	×
			PD7	✓	✓	✓	×
			PE7	✓	✓	✓	✓
	IRQ8	IRQ8-DS (input)	P40	✓	✓	✓	✓
		IRQ8 (input)	P00	✓	×	×	×
			P20	✓	✓	×	×
			P73	✓	×	×	×
			P80	✓	×	×	×
			PE0	✓	✓	✓	×
	IRQ9	IRQ9-DS (input)	P41	✓	✓	✓	✓
		IRQ9 (input)	P01	✓	×	×	×
			P21	✓	✓	×	×
			P81	✓	×	×	×
			P91	✓	×	×	×
			PE1	✓	✓	✓	×

Table 23.1 Functions Assigned to Each Multiplexed Pin (3/18)

Module/Function	Channel	Pin Functions	Allocation Port	Package 145-pin 144-pin	100-pin	64-pin	48-pin
Interrupt	IRQ10	IRQ10-DS (input)	P42	✓	✓	✓	✓
		IRQ10 (input)	P02	✓	×	×	×
			P55	✓	✓	×	×
			P72	✓*2	×	×	×
			P92	✓	×	×	×
			PA2	✓	✓	✓	✓
			PC2	✓	✓	×	×
	IRQ11	IRQ11-DS (input)	P43	✓	✓	✓	✓
		IRQ11 (input)	P03	✓	×	×	×
			P93	✓	×	×	×
			PA1	✓	✓	✓	✓
			PC3	✓	✓	×	×
			PE3	✓	✓	×	×
			PJ3	✓	✓	×	×
	IRQ12	IRQ12-DS (input)	P44	✓	✓	×	×
		IRQ12 (input)	P24	✓	✓	×	×
			P74	✓	×	×	×
			PB0	✓	✓	×	×
			PC1	✓	✓	✓	×
			PC4	✓	✓	✓	✓
			PE4	✓	✓	×	×
	IRQ13	IRQ13-DS (input)	P45	✓	✓	×	×
		IRQ13 (input)	P05	✓	✓	✓*3	×
			P65	✓	×	×	×
			P75	✓	×	×	×
			PB5	✓	✓	✓	✓
			PC6	✓	✓	✓	✓
			PJ5	✓	×	×	×
	IRQ14	IRQ14-DS (input)	P46	✓	✓	×	×
		IRQ14 (input)	P66	✓	×	×	×
			P76	✓	×	×	×
			P86	✓	×	×	×
			PA6	✓	✓	✓	✓
			PC0	✓	✓	✓	×
			PC7	✓	✓	✓	✓
	IRQ15	IRQ15-DS (input)	P47	✓	✓	×	×
		IRQ15 (input)	P07	✓	✓	×	×
			P22	✓	✓	×	×
			P67	✓	×	×	×
			P87	✓	×	×	×
			PB7	✓	✓	✓	✓

Table 23.1 Functions Assigned to Each Multiplexed Pin (4/18)

Module/Function	Channel	Pin Functions	Allocation Port	Package 145-pin 144-pin	100-pin	64-pin	48-pin
Multi-function timer unit 3	MTU0	MTIOC0A (input/output)	P34	✓	✓	✓	✓
			PB3	✓	✓	×	×
		MTIOC0B (input/output)	P13	✓	✓	✓	✓
			P15	✓	✓	×	×
			PA1	✓	✓	✓	✓
		MTIOC0C (input/output)	P32	✓	✓	×	×
			PB1	✓	✓	×	×
		MTIOC0D (input/output)	P33	✓	✓	×	×
			PA3	✓	✓	×	×
	MTU1	MTIOC1A (input/output)	P20	✓	✓	×	×
			PE4	✓	✓	×	×
		MTIOC1B (input/output)	P21	✓	✓	×	×
			PB5	✓	✓	✓	✓
	MTU2	MTIOC2A (input/output)	P26	✓	✓	✓	✓
			PB5	✓	✓	✓	✓
		MTIOC2B (input/output)	P27	✓	✓	✓	✓
			PE5	✓	✓	×	×
	MTU3	MTIOC3A (input/output)	P14	✓	✓	×	×
			P17	✓	✓	✓	✓
			PC1	✓	✓	✓	×
			PC7	✓	✓	✓	✓
		MTIOC3B (input/output)	P17	✓	✓	✓	✓
			P22	✓	✓	×	×
			P80	✓	×	×	×
			PB7	✓	✓	✓	✓
			PC5	✓	✓	✓	✓
			PE1	✓	✓	✓	×
		MTIOC3C (input/output)	P16	✓	✓	✓	✓
			P56	✓	×	×	×
			PC0	✓	✓	✓	×
			PC6	✓	✓	✓	✓
			PJ3	✓	✓	×	×
		MTIOC3D (input/output)	P16	✓	✓	✓	✓
			P23	✓	✓	×	×
			P81	✓	×	×	×
			PB6	✓	✓	✓	✓
			PC4	✓	✓	✓	✓
			PE0	✓	✓	✓	×

Table 23.1 Functions Assigned to Each Multiplexed Pin (5/18)

Module/Function	Channel	Pin Functions	Allocation Port	Package 145-pin 144-pin	100-pin	64-pin	48-pin
Multi-function timer unit 3	MTU4	MTIOC4A (input/output)	P21	✓	✓	×	×
			P24	✓	✓	×	×
			P82	✓	×	×	×
			PA0	✓	✓	×	×
			PB3	✓	✓	×	×
			PE2	✓	✓	✓	×
		MTIOC4B (input/output)	P17	✓	✓	✓	✓
			P30	✓	✓	✓	✓
			P54	✓	✓	×	×
			PC2	✓	✓	×	×
			PD1	✓	✓	×	×
			PE3	✓	✓	×	×
		MTIOC4C (input/output)	P25	✓	✓	×	×
			P83	✓	×	×	×
			P87	✓	×	×	×
			PB1	✓	✓	×	×
			PE1	✓	✓	✓	×
			PE5	✓	✓	×	×
		MTIOC4D (input/output)	P31	✓	✓	✓	✓
			P55	✓	✓	×	×
			P86	✓	×	×	×
			PC3	✓	✓	×	×
			PD2	✓	✓	✓	✓
			PE4	✓	✓	×	×
	MTU5	MTIC5U (input)	P12	✓	✓	✓	✓
			PA4	✓	✓	✓	✓
			PD7	✓	✓	✓	×
		MTIC5V (input)	PA6	✓	✓	✓	✓
			PD6	✓	✓	✓	×
		MTIC5W (input)	PB0	✓	✓	×	×
			PD5	✓	✓	✓	✓
	MTU6	MTIOC6A (input/output)	PE7	✓	✓	✓	✓
		MTIOC6B (input/output)	PA5	✓	✓	×	×
		MTIOC6C (input/output)	PE6	✓	✓	✓	✓
		MTIOC6D (input/output)	PA0	✓	✓	×	×
	MTU7	MTIOC7A (input/output)	PA2	✓	✓	✓	✓
		MTIOC7B (input/output)	PA1	✓	✓	✓	✓
		MTIOC7C (input/output)	P67	✓	×	×	×
		MTIOC7D (input/output)	P66	✓	×	×	×
	MTU8	MTIOC8A (input/output)	PD6	✓	✓	✓	×
		MTIOC8B (input/output)	PD4	✓	✓	✓	✓
		MTIOC8C (input/output)	PD5	✓	✓	✓	✓
		MTIOC8D (input/output)	PD3	✓	✓	✓	✓

Table 23.1 Functions Assigned to Each Multiplexed Pin (6/18)

Module/Function	Channel	Pin Functions	Allocation Port	Package 145-pin 144-pin	100-pin	64-pin	48-pin
Multi-function timer unit 3	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 3	POE0	POE0# (input)	P32	✓	✓	×	×
			P93	✓	×	×	×
			PC4	✓	✓	✓	✓
			PD1	✓	✓	×	×
			PD7	✓	✓	✓	×
	POE4	POE4# (input)	P33	✓	✓	×	×
			P92	✓	×	×	×
			PB5	✓	✓	✓	✓
			PD0	✓	✓	×	×
			PD6	✓	✓	✓	×
	POE8	POE8# (input)	P17	✓	✓	✓	✓
			P30	✓	✓	✓	✓
			PD3	✓	✓	✓	✓
			PE3	✓	✓	×	×
			PJ5	✓	×	×	×
	POE10	POE10# (input)	P32	✓	✓	×	×
			P34	✓	✓	✓	✓
			PA6	✓	✓	✓	✓
			PD5	✓	✓	✓	✓
	POE11	POE11# (input)	P33	✓	✓	×	×
			PB3	✓	✓	×	×
			PD4	✓	✓	✓	✓

Table 23.1 Functions Assigned to Each Multiplexed Pin (7/18)

Module/Function	Channel	Pin Functions	Allocation Port	Package 145-pin 144-pin	100-pin	64-pin	48-pin
16-bit timer pulse unit	TPU0	TIOCA0 (input/output)	P86	✓	×	×	×
			PA0	✓	✓	×	×
		TIOCB0 (input/output)	P17	✓	✓	✓	✓
			PA1	✓	✓	✓	✓
		TIOCC0 (input/output)	P32	✓	✓	×	×
		TIOCD0 (input/output)	P33	✓	✓	×	×
			PA3	✓	✓	×	×
	TPU1	TIOCA1 (input/output)	P56	✓	×	×	×
			PA4	✓	✓	✓	✓
		TIOCB1 (input/output)	P16	✓	✓	✓	✓
			PA5	✓	✓	×	×
	TPU2	TIOCA2 (input/output)	P87	✓	×	×	×
			PA6	✓	✓	✓	✓
		TIOCB2 (input/output)	P15	✓	✓	×	×
			PA7	✓	✓	✓	×
	TPU3	TIOCA3 (input/output)	P21	✓	✓	×	×
			PB0	✓	✓	×	×
		TIOCB3 (input/output)	P20	✓	✓	×	×
			PB1	✓	✓	×	×
		TIOCC3 (input/output)	P22	✓	✓	×	×
			PB2	✓	✓	×	×
		TIOCD3 (input/output)	P23	✓	✓	×	×
			PB3	✓	✓	×	×
	TPU4	TIOCA4 (input/output)	P25	✓	✓	×	×
			PB4	✓	✓	×	×
		TIOCB4 (input/output)	P24	✓	✓	×	×
			PB5	✓	✓	✓	✓
	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	✓	✓	✓	×

Table 23.1 Functions Assigned to Each Multiplexed Pin (8/18)

Module/Function	Channel	Pin Functions	Allocation Port	Package 145-pin 144-pin	100-pin	64-pin	48-pin
Programmable pulse generator	PPG0	PO0 (output)	P20	✓	✓	×	×
		PO1 (output)	P21	✓	✓	×	×
		PO2 (output)	P22	✓	✓	×	×
		PO3 (output)	P23	✓	✓	×	×
		PO4 (output)	P24	✓	✓	×	×
		PO5 (output)	P25	✓	✓	×	×
		PO6 (output)	P26	✓	✓	×	×
		PO7 (output)	P27	✓	✓	×	×
		PO8 (output)	P30	✓	✓	×	×
		PO9 (output)	P31	✓	✓	×	×
		PO10 (output)	P32	✓	✓	×	×
		PO11 (output)	P33	✓	✓	×	×
		PO12 (output)	P34	✓	✓	×	×
		PO13 (output)	P13	✓	✓	×	×
			P15	✓	✓	×	×
		PO14 (output)	P16	✓	✓	×	×
		PO15 (output)	P14	✓	✓	×	×
			P17	✓	✓	×	×
	PPG1	PO16 (output)	P73	✓	×	×	×
			PA0	✓	✓	×	×
		PO17 (output)	PA1	✓	✓	×	×
			PC0	✓	✓	×	×
		PO18 (output)	PA2	✓	✓	×	×
			PC1	✓	✓	×	×
			PE1	✓	✓	×	×
		PO19 (output)	P74	✓	×	×	×
			PA3	✓	✓	×	×
		PO20 (output)	P75	✓	×	×	×
			PA4	✓	✓	×	×
		PO21 (output)	PA5	✓	✓	×	×
			PC2	✓	✓	×	×
		PO22 (output)	P76	✓	×	×	×
			PA6	✓	✓	×	×
		PO23 (output)	P77	✓	×	×	×
			PA7	✓	✓	×	×
			PE2	✓	✓	×	×
		PO24 (output)	PB0	✓	✓	×	×
			PC3	✓	✓	×	×
		PO25 (output)	PB1	✓	✓	×	×
			PC4	✓	✓	×	×
		PO26 (output)	P80	✓	×	×	×
			PB2	✓	✓	×	×
			PE3	✓	✓	×	×

Table 23.1 Functions Assigned to Each Multiplexed Pin (9/18)

Module/Function	Channel	Pin Functions	Allocation Port	Package 145-pin 144-pin	100-pin	64-pin	48-pin
Programmable pulse generator	PPG1	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	✓	✓	×	×
			PH1	✓	✓	✓	×
		TMCI0 (input)	P01	✓	×	×	×
			P21	✓	✓	×	×
			PB1	✓	✓	×	×
		TMRI0 (input)	P00	✓	×	×	×
			P20	✓	✓	×	×
			PA4	✓	✓	✓	✓
			PH2	✓	✓	✓	×
	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)	P27	✓	✓	✓	✓
			P34	✓	✓	✓	✓
			PA6	✓	✓	✓	✓
		TMRI3 (input)	P30	✓	✓	✓	✓
			P33	✓	✓	×	×

Table 23.1 Functions Assigned to Each Multiplexed Pin (10/18)

Module/Function	Channel	Pin Functions	Allocation Port	Package 145-pin 144-pin	100-pin	64-pin	48-pin
Compare match timer W	CMTW0	TOC0 (output)	PC7	✓	✓	✓	✓
		TIC0 (input)	PC6	✓	✓	✓	✓
		TOC1 (output)	PE7	✓	✓	✓	✓
		TIC1 (input)	PE6	✓	✓	✓	✓
	CMTW1	TOC2 (output)	PD3	✓	✓	✓	✓
		TIC2 (input)	PD2	✓	✓	✓	✓
		TOC3 (output)	PE3	✓	✓	×	×
		TIC3 (input)	PE2	✓	✓	✓	×
Realtime clock		RTCOUT (output)	P16	✓	✓	✓	×
			P32	✓	✓	×	×
		RTCIC0 (input)*1	P30	✓	✓	✓	×
		RTCIC1 (input)*1	P31	✓	✓	✓	×
		RTCIC2 (input)*1	P32	✓	✓	×	×
Battery backup		TAMPI0 (input)*1	P30	✓	✓	✓	×
		TAMPI1 (input)*1	P31	✓	✓	✓	×
		TAMPI2 (input)*1	P32	✓	✓	×	×
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	✓	✓	✓	✓
		TXD1 (output)/SMOSI1 (input/output)/ SSDA1 (input/output)	P16	✓	✓	✓	✓
			P26	✓	✓	✓	✓
		SCK1 (input/output)	P17	✓	✓	✓	✓
			P27	✓	✓	✓	✓
		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)	P51	✓	✓	×	×
		CTS2# (input)/RTS2# (output)/ SS2# (input)	P54	✓	✓	×	×
			PJ5	✓	×	×	×

Table 23.1 Functions Assigned to Each Multiplexed Pin (11/18)

Module/Function	Channel	Pin Functions	Allocation Port	Package 145-pin 144-pin	100-pin	64-pin	48-pin
Serial communications interface	SCI3	RXD3 (input)/SMISO3 (input/output)/ SSCL3 (input/output)	P16	✓	✓	✓	✓
			P25	✓	✓	×	×
		TXD3 (output)/SMOSI3 (input/output)/ SSDA3 (input/output)	P17	✓	✓	✓	✓
			P23	✓	✓	×	×
		SCK3 (input/output)	P15	✓	✓	×	×
			P24	✓	✓	×	×
		CTS3# (input)/RTS3# (output)/ SS3# (input)	P26	✓	✓	✓	✓
	SCI4	RXD4 (input)/SMISO4 (input/output)/ SSCL4 (input/output)	PB0	✓	×	×	×
		TXD4 (output)/SMOSI4 (input/output)/ SSDA4 (input/output)	PB1	✓	×	×	×
		SCK4 (input/output)	PB3	✓	×	×	×
		CTS4# (input)/RTS4# (output)/ SS4# (input)	PB2	✓	×	×	×
	SCI5	RXD5 (input)/SMISO5 (input/output)/ SSCL5 (input/output)	PA2	✓	✓	✓	✓
			PA3	✓	✓	×	×
			PC2	✓	✓	×	×
		TXD5 (output)/SMOSI5 (input/output)/ SSDA5 (input/output)	PA4	✓	✓	✓	✓
			PC3	✓	✓	×	×
		SCK5 (input/output)	PA1	✓	✓	✓	✓
			PC1	✓	✓	✓	×
			PC4	✓	✓	✓	✓
		CTS5# (input)/RTS5# (output)/ SS5# (input)	PA6	✓	✓	✓	✓
			PC0	✓	✓	✓	×
	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)	P55	✓	×	×	×
			P90	✓	×	×	×
		SCK7 (input/output)	P56	✓	×	×	×
			P91	✓	×	×	×
		CTS7# (input)/RTS7# (output)/ SS7# (input)	P93	✓	×	×	×

Table 23.1 Functions Assigned to Each Multiplexed Pin (12/18)

Module/Function	Channel	Pin Functions	Allocation Port	Package 145-pin 144-pin	100-pin	64-pin	48-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	✓	×	×	×
			P86	✓	×	×	×
			PC6	✓	✓	✓	✓
		TXD10 (output)/SMOSI10 (input/output)/ SSDA10 (input/output)	P82	✓	×	×	×
			P87	✓	×	×	×
			PC7	✓	✓	✓	✓
		SCK10 (input/output)	P80	✓	×	×	×
			P83	✓	×	×	×
			PC5	✓	✓	✓	✓
		RTS10# (output)	P80	✓	×	×	×
		CTS10# (input)/SS10#(input)	P83	✓	×	×	×
		CTS10# (input)/RTS10# (output)/ SS10# (input)	PC4	✓	✓	✓	✓
	SCI11	RXD11 (input)/SMISO11 (input/output)/ SSCL11 (input/output)	P76	✓	×	×	×
			PB6	✓	✓	✓	✓
		TXD11 (output)/SMOSI11 (input/output)/ SSDA11 (input/output)	P77	✓	×	×	×
			PB7	✓	✓	✓	✓
		SCK11 (input/output)	P75	✓	×	×	×
			PB5	✓	✓	✓	✓
		RTS11# (output)	P75	✓	×	×	×
		CTS11# (input)/SS11#(input)	P74	✓	×	×	×
		CTS11# (input)/RTS11# (output)/ SS11#(input)	PB4	✓	✓	×	×
	SCI12	RXD12 (input)/SMISO12 (input/output)/ SSCL12 (input/output)/RXDX12 (input)	PE2	✓	✓	✓	×
			PA2	✓	✓	✓	✓
		TXD12 (output)/SMOSI12 (input/output)/ SSDA12 (input/output)/TXDX12 (output)/ SIOX12 (input/output)	PE1	✓	✓	✓	×
			PA4	✓	✓	✓	✓
		SCK12 (input/output)	PE0	✓	✓	✓	×
			PA1	✓	✓	✓	✓
		CTS12# (input)/RTS12# (output)/ SS12# (input)	PE3	✓	✓	×	×
			PA6	✓	✓	✓	✓

Table 23.1 Functions Assigned to Each Multiplexed Pin (13/18)

Module/Function	Channel	Pin Functions	Allocation Port	Package 145-pin 144-pin	100-pin	64-pin	48-pin
Serial communications interface	RSCI10	RXD010 (input)/ SMISO010 (input/output)/ SSCL010 (input/output)	P81	✓	×	×	×
			P86	✓	×	×	×
			PC6	✓	✓	✓	✓
		TXD010 (output)/ SMOSI010 (input/output)/ SSDA010 (input/output)	P82	✓	×	×	×
			P87	✓	×	×	×
			PC7	✓	✓	✓	✓
		SCK010 (input/output)	P80	✓	×	×	×
			P83	✓	×	×	×
			PC5	✓	✓	✓	✓
		RTS010# (output)	P80	✓	×	×	×
		CTS010# (input)/SS010#(input)	P83	✓	×	×	×
		CTS010# (input)/RTS010# (output)/ SS010# (input)	PC4	✓	✓	✓	✓
		DE010 (output)	P80	✓	×	×	×
			PC4	✓	✓	✓	✓
	RSCI11	RXD011 (input)/ SMISO011 (input/output)/ SSCL011 (input/output)	P76	✓	×	×	×
			PB6	✓	✓	✓	✓
			PC0	✓	✓	✓	×
		TXD011 (output)/ SMOSI011 (input/output)/ SSDA011 (input/output)	P77	✓	×	×	×
			PB7	✓	✓	✓	✓
			PC1	✓	✓	✓	×
		SCK011 (input/output)	P75	✓	×	×	×
			PB5	✓	✓	✓	✓
		TXDA011 (output)	PC1	✓	✓	✓	×
		TXDB011 (output)	PC2	✓	✓	×	×
		RTS011# (output)	P75	✓	×	×	×
		CTS011# (input)/SS011#(input)	P74	✓	×	×	×
		CTS011# (input)/RTS011# (output)/ SS011#(input)	PB4	✓	✓	×	×
		DE011 (output)	P75	✓	×	×	×
			PB4	✓	✓	×	×
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 (input/output)	P17	✓	✓	✓	✓
Hi-speed I^2C bus interface	RIICHS0	SCLHS0[FM+/HS] (input/output)	P12	✓	✓	✓	✓
		SDAHS0[FM+/HS] (input/output)	P13	✓	✓	✓	✓

Table 23.1 Functions Assigned to Each Multiplexed Pin (14/18)

Module/Function	Channel	Pin Functions	Allocation Port	Package 145-pin 144-pin	100-pin	64-pin	48-pin
USB 2.0FS host/ function module	USB0	USB0_DP (input/output)*1	PH1	✓	✓	✓	×
		USB0_DM (input/output)*1	PH2	✓	✓	✓	×
		USB0_VBUS (input)	P16	✓	✓	✓	×
		USB0_EXICEN (output)	P21	✓	✓	×	×
		USB0_VBUSEN (output)	P16	✓	✓	×	×
			P24	✓	✓	×	×
			P32	✓	✓	×	×
		USB0_OVRCURA (input)	P14	✓	✓	×	×
		USB0_OVRCURB (input)	P16	✓	✓	×	×
			P22	✓	✓	×	×
		USB0_ID (input)	P20	✓	✓	×	×
	USB1	USB1_VBUS (input)	P73	✓*4	×	×	×
		USB1_EXICEN (output)	P80	✓*4	×	×	×
		USB1_VBUSEN (output)	P73	✓*4	×	×	×
			P74	✓*4	×	×	×
			P82	✓*4	×	×	×
		USB1_OVRCURA (input)	P75	✓*4	×	×	×
		USB1_OVRCURB (input)	P73	✓*4	×	×	×
			P81	✓*4	×	×	×
		USB1_ID (input)	P77	✓*4	×	×	×
CAN module	CAN0	CRX0 (input)	P33	✓	✓	×	×
			PD2	✓	✓	×	×
		CTX0 (output)	P32	✓	✓	×	×
			PD1	✓	✓	×	×
	CAN1	CRX1-DS (input)	P15	✓	✓	×	×
		CRX1 (input)	P55	✓	✓	×	×
		CTX1 (output)	P14	✓	✓	×	×
			P54	✓	✓	×	×
Serial peripheral interface	RSPI0	RSPCKA (input/output)	PA5	✓	✓	×	×
			PC5	✓	✓	✓	✓
		MOSIA (input/output)	PA6	✓	✓	✓	✓
			PC6	✓	✓	✓	✓
		MISOA (input/output)	PA7	✓	✓	✓	×
			PC7	✓	✓	✓	✓
		SSLA0 (input/output)	PA4	✓	✓	✓	✓
			PC4	✓	✓	✓	✓
		SSLA1 (output)	PA0	✓	✓	×	×
			PC0	✓	✓	✓	×
		SSLA2 (output)	PA1	✓	✓	✓	✓
			PC1	✓	✓	✓	×
		SSLA3 (output)	PA2	✓	✓	✓	✓
			PC2	✓	✓	×	×

Table 23.1 Functions Assigned to Each Multiplexed Pin (15/18)

Module/Function	Channel	Pin Functions	Allocation Port	Package 145-pin 144-pin	100-pin	64-pin	48-pin
Serial peripheral interface	RSPI1	RSPCKB (input/output)	P27	✓	✓	✓	✓
			PE5	✓	✓	×	×
		MOSIB (input/output)	P26	✓	✓	✓	✓
			PE6	✓	✓	✓	✓
		MISOB (input/output)	P30	✓	✓	✓	✓
			PE7	✓	✓	✓	✓
		SSLB0 (input/output)	P31	✓	✓	✓	✓
			PE4	✓	✓	×	×
		SSLB1 (output)	P50	✓	✓	×	×
			PE0	✓	✓	✓	×
		SSLB2 (output)	P51	✓	✓	×	×
			PE1	✓	✓	✓	×
		SSLB3 (output)	P52	✓	✓	×	×
			PE2	✓	✓	✓	×
	RSPI2	RSPCKC (input/output)	P56	✓	×	×	×
			PD3	✓	✓	×	×
		MOSIC (input/output)	P54	✓	✓	×	×
			PD1	✓	✓	×	×
		MISOC (input/output)	P55	✓	✓	×	×
			PD2	✓	✓	×	×
		SSLC0 (input/output)	PD4	✓	✓	×	×
		SSLC1 (output)	PD5	✓	✓	×	×
		SSLC2 (output)	PD6	✓	✓	×	×
		SSLC3 (output)	PD7	✓	✓	×	×
Serial peripheral interface	RSPIA0	RSPCK0 (input/output)	PA5	✓	✓	×	×
			PC5	✓	✓	✓	✓
		MOSI0 (input/output)	PA6	✓	✓	✓	✓
			PC6	✓	✓	✓	✓
		MISO0 (input/output)	PA7	✓	✓	✓	×
			PC7	✓	✓	✓	✓
		SSL00 (input/output)	PA4	✓	✓	✓	✓
			PC4	✓	✓	✓	✓
		SSL01 (output)	PA0	✓	✓	×	×
			PC0	✓	✓	✓	×
		SSL02 (output)	PA1	✓	✓	✓	✓
			PC1	✓	✓	✓	×
		SSL03 (output)	PA2	✓	✓	✓	✓
			PC2	✓	✓	×	×

Table 23.1 Functions Assigned to Each Multiplexed Pin (16/18)

Module/Function	Channel	Pin Functions	Allocation Port	Package 145-pin 144-pin	100-pin	64-pin	48-pin
Quad-SPI memory interface		QSPCLK (output)	P77	✓	×	×	×
			PD5	✓	✓	✓	✓
		QSSL (output)	P76	✓	×	×	×
			PD4	✓	✓	✓	✓
		QIO0 (input/output)	PC3	✓	✓	×	×
			PD6	✓	✓	✓	×
			PE6	✓	✓	✓	✓
		QIO1 (input/output)	PC4	✓	✓	✓	✓
			PD7	✓	✓	✓	×
			PE7	✓	✓	✓	✓
		QIO2 (input/output)	P80	✓	×	×	×
			PD2	✓	✓	✓	✓
		QIO3 (input/output)	P81	✓	×	×	×
			PD3	✓	✓	✓	✓
Serial sound interface	SSIE	AUDIO_CLK (input)	P22	✓	✓	×	×
			PC4	✓	✓	✓	✓
	SSIE0	SSIBCK0 (input/output)	P23	✓	✓	×	×
			PC5	✓	✓	✓	✓
		SSILRCK0 (input/output)	P21	✓	✓	×	×
			PC6	✓	✓	✓	✓
		SSIRXD0 (input)	P20	✓	✓	×	×
			P53	✓	✓	✓	✓
		SSITXD0 (output)	P17	✓	✓	✓	✓
			PC7	✓	✓	✓	✓
SD host interface		SDHI_CLK (output)	P21	✓	✓	×	×
			P77	✓	×	×	×
			PD5	✓	✓	✓	✓
		SDHI_CMD (input/output)	P20	✓	✓	×	×
			P76	✓	×	×	×
			PD4	✓	✓	✓	✓
		SDHI_CD (input)	P25	✓	✓	×	×
			P81	✓	×	×	×
			PA1	✓	✓	✓	✓
			PE6	✓	✓	✓	✓
		SDHI_WP (input)	P24	✓	✓	×	×
			P80	✓	×	×	×
			PA2	✓	✓	✓	✓
			PE7	✓	✓	✓	✓
		SDHI_D0 (input/output)	P22	✓	✓	×	×
			PC3	✓	✓	×	×
			PD6	✓	✓	✓	×
			PE6	✓	✓	✓	✓

Table 23.1 Functions Assigned to Each Multiplexed Pin (17/18)

Module/Function	Channel	Pin Functions	Allocation Port	Package 145-pin 144-pin	100-pin	64-pin	48-pin
SD host interface		SDHI_D1 (input/output)	P23	✓	✓	×	×
			PC4	✓	✓	✓	✓
			PD7	✓	✓	✓	×
			PE7	✓	✓	✓	✓
		SDHI_D2 (input/output)	P75	✓	×	×	×
			P87	✓	×	×	×
			PD2	✓	✓	✓	✓
		SDHI_D3 (input/output)	P17	✓	✓	✓	✓
			PC2	✓	✓	×	×
			PD3	✓	✓	✓	✓
12-bit A/D converter		AN000 (input)*1	P40	✓	✓	✓	✓
		AN001 (input)*1	P41	✓	✓	✓	✓
		AN002 (input)*1	P42	✓	✓	✓	✓
		AN003 (input)*1	P43	✓	✓	✓	✓
		AN004 (input)*1	P44	✓	✓	×	×
		AN005 (input)*1	P45	✓	✓	×	×
		AN006 (input)*1	P46	✓	✓	×	×
		AN007 (input)*1	P47	✓	✓	×	×
		ADTRG0# (input)	P07	✓	✓	×	×
			P16	✓	✓	✓	✓
			P25	✓	✓	×	×
		AN100 (input)*1	PD7	✓	✓	✓	×
		AN101 (input)*1	PD6	✓	✓	✓	×
		AN102 (input)*1	PD5	✓	✓	✓	✓
		AN103 (input)*1	PD4	✓	✓	✓	✓
		AN104 (input)*1	PD3	✓	✓	✓	✓
		AN105 (input)*1	PD2	✓	✓	✓	✓
		AN106 (input)*1	PD1	✓	✓	×	×
		AN107 (input)*1	PD0	✓	✓	×	×
		AN108 (input)*1	P90	✓	×	×	×
		AN109 (input)*1	P02	✓	×	×	×
		AN110 (input)*1	P01	✓	×	×	×
		AN111 (input)*1	P00	✓	×	×	×
		ANEX0 (output)*1	PE0	✓	✓	✓	×
		ANEX1 (input)*1	PE1	✓	✓	✓	×
		ADTRG1# (input)	P13	✓	✓	✓	✓
			P17	✓	✓	✓	✓
Clock generation circuit		CLKOUT (output)	P25	✓	✓	×	×
		EXCIN (input)*1	PJ3	✓	✓	×	×
		EXTAL (input)*1	P36	✓	✓	✓	✓
		XTAL (output)*1	P37	✓	✓	✓	✓
Clock frequency accuracy measurement circuit		CACREF (input)	PA0	✓	✓	×	×
			PC7	✓	✓	✓	✓

Table 23.1 Functions Assigned to Each Multiplexed Pin (18/18)

Module/Function	Channel	Pin Functions	Allocation Port	Package 145-pin 144-pin	100-pin	64-pin	48-pin
Capacitive touch sensing unit		TSCAP (—)	PC4	✓	✓	✓	✓
		TS0 (output)	P34	✓	✓	✓	✓
		TS1 (output)	P33	✓	✓	×	×
		TS2 (output)	P27	✓	✓	✓	✓
		TS3 (output)	P26	✓	✓	✓	✓
		TS4 (output)	P25	✓	✓	×	×
		TS5 (output)	P24	✓	✓	×	×
		TS6 (output)	P23	✓	✓	×	×
		TS7 (output)	P22	✓	✓	×	×
		TS8 (output)	P21	✓	✓	×	×
		TS9 (output)	P20	✓	✓	×	×
		TS10 (output)	P15	✓	✓	×	×
		TS11 (output)	P14	✓	✓	×	×
		TS12 (output)	P53	✓	✓	✓	✓
		TS13 (output)	PC6	✓	✓	✓	✓
		TS14 (output)	PC5	✓	✓	✓	✓
		TS15 (output)	PC1	✓	✓	✓	×
		TS16 (output)	PC0	✓	✓	✓	×
Remote control signal receiver	REMC0	PMC0-DS (input)*1	P53	✓	✓	✓	✓
			PB3	✓	✓	×	×
			PC3	✓	✓	×	×

Note 1. To use this pin function, set the corresponding pin as general input (set the PORTm.PDR.Bn and PORTm.PMR.Bn bits to 0).
Note 2. Not provided on the 145-pin TFLGA (0.65-mm pitch).
Note 3. Not provided on the 64-pin TFBGA.
Note 4. Only the product in a 145-pin TFLGA (0.65-mm pitch) has this function.

23.2 Register Descriptions

The registers and bits of unsupported pins, depending on the package, are reserved. The write value to the reserved bits is the value after a reset.

23.2.1 Write-Protect Register (PWPR)

Address(es): 0008 C11Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	B0WI	PFSWE	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	PFSWE	PFS Register Write Enable	0: Writing to the PFS register is disabled 1: Writing to the PFS register is enabled	R/W
b7	B0WI	PFSWE Bit Write Disable	0: Writing to the PFSWE bit is enabled 1: Writing to the PFSWE bit is disabled	R/W

PFSWE Bit (PFS Register Write Enable)

Writing to PmnPFS register is enabled only when the PFSWE bit is set to 1.
To set the PFSWE bit to 1, write 1 to the PFSWE bit after writing 0 to the B0WI bit.

B0WI Bit (PFSWE Bit Write Disable)

Writing to the PFSWE bit is enabled only when the B0WI bit is set to 0.

23.2.2 P0n Pin Function Control Register (P0nPFS) (n = 0 to 3, 5, 7)

Address(es): P00PFS 0008 C140h, P01PFS 0008 C141h, P02PFS 0008 C142h, P03PFS 0008 C143h, P05PFS 0008 C145h, P07PFS 0008 C147h

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 23.2 and Table 23.3.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P00: IRQ8 (145/144 pin) P01: IRQ9 (145/144 pin) P02: IRQ10 (145/144 pin) P03: IRQ11 (145/144 pin) P05: IRQ13 (145/144/100/64 pin) P07: IRQ15 (145/144/100 pin)	R/W
b7	ASEL	Analog Function Select	0: Used other than as analog pin. 1: Used as analog pin. P00: AN111 (145/144 pin) P01: AN110 (145/144 pin) P02: AN109 (145/144 pin)	R/W

The port mn pin function control register (PmnPFS) selects the pin function. Bits PSEL[5:0] select the peripheral function which is assigned to bits.
The ISEL bit is set when a pin is used as an IRQ input pin. This setting can be used with the combination of the peripheral function, though IRQn (external pin interrupt) of the same number should not be enabled by two or more pins.
The ASEL bit is set when a pin is used as an analog pin. When the pin is set as an analog pin by the ASEL bit, select the general I/O port by the port mode register (PORTm.PMR) and specify input by the port direction register (PORTm.PDR). The pin state cannot be read at this point, since the PmnPFS register is protected by the write-protect register (PWPR). Modify the register after releasing the protection.
The ISEL bit to which IRQn is not specified is reserved. The ASEL bit to which analog input/output is not specified is reserved.

Table 23.2 Register Settings for Input/Output Pin Function in 145-Pin TFLGA, 144-Pin LFQFP

	Pin			
PSEL[5:0] Settings	**P00**	**P01**	**P02**	**P07**
000000b (initial value)	Hi-Z			
000101b	TMRI0	TMCI0	TMCI1	—
001001b	—	—	—	ADTRG0#
001010b	TXD6 SMOSI6 SSDA6	RXD6 SMISO6 SSCL6	SCK6	—

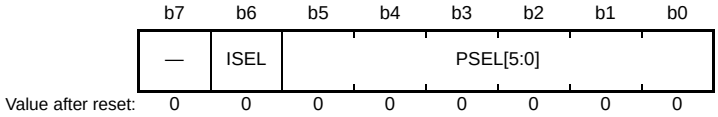
—: Do not specify this value.

Table 23.3 Register Settings for Input/Output Pin Function in 100-Pin TFLGA, 100-Pin LFQFP

	Pin
PSEL[5:0] Settings	**P07**
000000b (initial value)	Hi-Z
001001b	ADTRG0#

23.2.3 P1n Pin Function Control Register (P1nPFS) (n = 2 to 7)

Address(es): 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
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 23.4 and Table 23.5.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P12: IRQ2 (145/144/100/64/48 pin) P13: IRQ3 (145/144/100/64/48 pin) P14: IRQ4 (145/144/100 pin) P15: IRQ5 (145/144/100 pin) P16: IRQ6 (145/144/100/64/48 pin) P17: IRQ7 (145/144/100/64/48 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 23.4 Register Settings for Input/Output Pin Function in 145-/100-Pin TFLGA, 144-/100-Pin LFQFP

	Pin					
PSEL[5:0] Settings	**P12**	**P13**	**P14**	**P15**	**P16**	**P17**
000000b (initial value)	Hi-Z					
000001b	MTIC5U	MTIOC0B	MTIOC3A	MTIOC0B	MTIOC3C	MTIOC3A
000010b	—	—	MTCLKA	MTCLKB	MTIOC3D	MTIOC3B
000011b	—	TIOCA5	TIOCB5	TIOCB2	TIOCB1	TIOCB0
000100b	—	—	TCLKA	TCLKB	TCLKC	TCLKD
000101b	TMCI1	TMO3	TMRI2	TMCI2	TMO2	TMO1
000110b	—	PO13	PO15	PO13	PO14	PO15
000111b	—	—	—	—	RTCOUT	POE8#
001000b	—	—	—	—	—	MTIOC4B
001001b	—	ADTRG1#	—	—	ADTRG0#	ADTRG1#
001010b	RXD2 SMISO2 SSCL2	TXD2 SMOSI2 SSDA2	—	RXD1 SMISO1 SSCL1	TXD1 SMOSI1 SSDA1	SCK1
001011b	—	—	CTS1# RTS1# SS1#	SCK3	RXD3 SMISO3 SSCL3	TXD3 SMOSI3 SSDA3
001111b	SCL0[FM+]	SDA0[FM+]	—	—	SCL2-DS	SDA2-DS
010000b	—	—	CTX1	CRX1-DS	—	—
010001b	—	—	—	—	USB0_VBUS	—
010010b	—	—	USB0_OVRCURA	—	USB0_VBUSEN	—
010011b	—	—	—	—	USB0_OVRCURB	—
010111b	—	—	—	—	—	SSITXD0
011010b	—	—	—	—	—	SDHI_D3-C
101011b	—	—	TS11	TS10	—	—
101111b	SCLHS0[FM+/HS]	SDAHS0[FM+/HS]	—	—	—	—

—: Do not specify this value.

Table 23.5 Register Settings for Input/Output Pin Function in 64-Pin TFBGA, 64-Pin LFQFP, 48-pin HWQFN

	Pin			
PSEL[5:0] Settings	**P12**	**P13**	**P16**	**P17**
000000b (initial value)	Hi-Z			
000001b	MTIC5U	MTIOC0B	MTIOC3C	MTIOC3A
000010b	—	—	MTIOC3D	MTIOC3B
000011b	—	TIOCA5	TIOCB1	TIOCB0
000100b	—	—	TCLKC	TCLKD
000101b	TMCI1	TMO3	TMO2	TMO1
000111b	—	—	RTCOUT*1	POE8#
001000b	—	—	—	MTIOC4B
001001b	—	ADTRG1#	ADTRG0#	ADTRG1#
001010b	RXD2 SMISO2 SSCL2	TXD2 SMOSI2 SSDA2	TXD1 SMOSI1 SSDA1	SCK1
001011b	—	—	RXD3 SMISO3 SSCL3	TXD3 SMOSI3 SSDA3
001111b	SCL0[FM+]	SDA0[FM+]	SCL2-DS	SDA2-DS
010001b*1	—	—	USB0_VBUS	—
010111b	—	—	—	SSITXD0
011010b	—	—	—	SDHI_D3-C
101111b	SCLHS0[FM+/HS]	SDAHS0[FM+/HS]	—	—

—: Do not specify this value.

Note 1. This setting is not supported by 48-pin products.

23.2.4 P2n Pin Function Control Register (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

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 23.6 and Table 23.7.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P20: IRQ8 (145/144/100 pin) P21: IRQ9 (145/144/100 pin) P22: IRQ15 (145/144/100 pin) P23: IRQ3 (145/144/100 pin) P24: IRQ12 (145/144/100 pin) P25: IRQ5 (145/144/100 pin) P26: IRQ6 (145/144/100/64/48 pin) P27: IRQ7 (145/144/100/64/48 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 23.6 Register Settings for Input/Output Pin Function in 145-/100-Pin TFLGA, 144-/100-Pin LFQFP

	Pin							
PSEL[5:0] Settings	**P20**	**P21**	**P22**	**P23**	**P24**	**P25**	**P26**	**P27**
000000b (initial value)	Hi-Z							
000001b	MTIOC1A	MTIOC1B	MTIOC3B	MTIOC3D	MTIOC4A	MTIOC4C	MTIOC2A	MTIOC2B
000010b	—	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB	—	—
000011b	TIOCB3	TIOCA3	TIOCC3	TIOCD3	TIOCB4	TIOCA4	—	—
000101b	TMRI0	TMCI0	TMO0	—	TMRI1	—	TMO1	TMCI3
000110b	PO0	PO1	PO2	PO3	PO4	PO5	PO6	PO7
001000b	—	MTIOC4A	—	—	—	—	—	—
001001b	—	—	—	—	—	ADTRG0#	—	—
001010b	TXD0 SMOSI0 SSDA0	RXD0 SMISO0 SSCL0	SCK0	TXD3 SMOSI3 SSDA3	SCK3	RXD3 SMISO3 SSCL3	TXD1 SMOSI1 SSDA1	SCK1
001011b	—	—	—	CTS0# RTS0# SS0#	—	—	CTS3# RTS3# SS3#	—
001101b	—	—	—	—	—	—	MOSIB-A	RSPCKB-A
001111b	SDA1	SCL1	—	—	—	—	—	—
010011b	USB0_ID	USB0_EXICEN	USB0_OVRCURB	—	USB0_VBUSEN	—	—	—
010111b	SSIRXD0	SSILRCK0	AUDIO_CLK	SSIBCK0	—	—	—	—
011000b	—	—	EDREQ0	EDACK0	EDREQ1	EDACK1	—	—
011010b	SDHI_CMD-C	SDHI_CLK-C	SDHI_D0-C	SDHI_D1-C	SDHI_WP	SDHI_CD	—	—
101010b	—	—	—	—	—	CLKOUT	—	—
101011b	TS9	TS8	TS7	TS6	TS5	TS4	TS3	TS2

—: Do not specify this value.

Table 23.7 Register Settings for Input/Output Pin Function in 64-Pin TFBGA, 64-Pin LFQFP, 48-pin HWQFN

	Pin	
PSEL[5:0] Settings	**P26**	**P27**
000000b (initial value)	Hi-Z	
000001b	MTIOC2A	MTIOC2B
000101b	TMO1	TMCI3
001010b	TXD1 SMOSI1 SSDA1	SCK1
001011b	CTS3# RTS3# SS3#	—
001101b	MOSIB-A	RSPCKB-A
101011b	TS3	TS2

—: Do not specify this value.

23.2.5 P3n Pin Function Control Register (P3nPFS) (n = 0 to 4)

Address(es): P30PFS 0008 C158h, P31PFS 0008 C159h, P32PFS 0008 C15Ah, P33PFS 0008 C15Bh, P34PFS 0008 C15Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 23.8 and Table 23.9.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P30: IRQ0-DS (145/144/100/64/48 pin) P31: IRQ1-DS (145/144/100/64/48 pin) P32: IRQ2-DS (145/144/100 pin) P33: IRQ3-DS (145/144/100 pin) P34: IRQ4 (145/144/100/64/48 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 23.8 Register Settings for Input/Output Pin Function in 145-/100-Pin TFLGA, 144-/100-Pin LFQFP

	Pin				
PSEL[5:0] Settings	**P30**	**P31**	**P32**	**P33**	**P34**
000000b (initial value)	Hi-Z				
000001b	MTIOC4B	MTIOC4D	MTIOC0C	MTIOC0D	MTIOC0A
000011b	—	—	TIOCC0	TIOCD0	—
000101b	TMRI3	TMCI2	TMO3	TMRI3	TMCI3
000110b	PO8	PO9	PO10	PO11	PO12
000111b	POE8#	—	RTCOUT	—	POE10#
001000b	—	—	POE0#	POE4#	—
001010b	RXD1 SMISO1 SSCL1	—	TXD6 SMOSI6 SSDA6	RXD6 SMISO6 SSCL6	SCK6
001011b	—	CTS1# RTS1# SS1#	TXD0 SMOSI0 SSDA0	RXD0 SMISO0 SSCL0	SCK0
001101b	MISOB-A	SSLB0-A	—	—	—
010000b	—	—	CTX0	CRX0	—
010011b	—	—	USB0_VBUSEN	—	—
011000b	—	—	—	EDREQ1	—
100001b	—	—	POE10#	POE11#	—
101011b	—	—	—	TS1	TS0

—: Do not specify this value.

Table 23.9 Register Settings for Input/Output Pin Function in 64-Pin TFBGA, 64-Pin LFQFP, 48-pin HWQFN

	Pin		
PSEL[5:0] Settings	**P30**	**P31**	**P34**
000000b (initial value)	Hi-Z		
000001b	MTIOC4B	MTIOC4D	MTIOC0A
000101b	TMRI3	TMCI2	TMCI3
000111b	POE8#	—	POE10#
001010b	RXD1 SMISO1 SSCL1	—	—
001011b	—	CRS1# RTS1# SS1#	—
001101b	MISOB-A	SSLB0-A	—
101011b	—	—	TS0

—: Do not specify this value.

23.2.6 P4n Pin Function Control Register (P4nPFS) (n = 0 to 7)

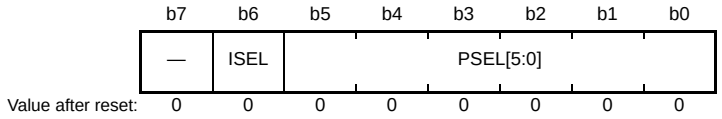
Address(es): P40PFS 0008 C160h, P41PFS 0008 C161h, P42PFS 0008 C162h, P43PFS 0008 C163h, P44PFS 0008 C164h, P45PFS 0008 C165h, P46PFS 0008 C166h, P47PFS 0008 C167h

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

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P40: IRQ8-DS (145/144/100/64/48 pin) P41: IRQ9-DS (145/144/100/64/48 pin) P42: IRQ10-DS (145/144/100/64/48 pin) P43: IRQ11-DS (145/144/100/64/48 pin) P44: IRQ12-DS (145/144/100 pin) P45: IRQ13-DS (145/144/100 pin) P46: IRQ14-DS (145/144/100 pin) P47: IRQ15-DS (145/144/100 pin)	R/W
b7	ASEL	Analog Function Select	0: Used other than as analog pin. 1: Used as analog pin. P40: AN000 (145/144/100/64/48 pin) P41: AN001 (145/144/100/64/48 pin) P42: AN002 (145/144/100/64/48 pin) P43: AN003 (145/144/100/64/48 pin) P44: AN004 (145/144/100 pin) P45: AN005 (145/144/100 pin) P46: AN006 (145/144/100 pin) P47: AN007 (145/144/100 pin)	R/W

23.2.7 P5n Pin Function Control Register (P5nPFS) (n = 0 to 6)

Address(es): P50PFS 0008 C168h, P51PFS 0008 C169h, P52PFS 0008 C16Ah, P53PFS 0008 C16Bh, P54PFS 0008 C16Ch, P55PFS 0008 C16Dh, P56PFS 0008 C16Eh



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 23.10, Table 23.11, and Table 23.12.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P50: IRQ0 (145/144/100 pin) P51: IRQ1 (145/144/100 pin) P52: IRQ2 (145/144/100 pin) P53: IRQ3 (145/144/100/64/48 pin) P54: IRQ4 (145/144/100 pin) P55: IRQ10 (145/144/100 pin) P56: IRQ6 (145/144 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 23.10 Register Settings for Input/Output Pin Function in 145-Pin TFLGA, 144-Pin LFQFP

	Pin						
PSEL[5:0] Settings	**P50**	**P51**	**P52**	**P53**	**P54**	**P55**	**P56**
000000b (initial value)	Hi-Z						
000001b	—	—	—	—	MTIOC4B	MTIOC4D	MTIOC3C
000011b	—	—	—	—	—	—	TIOCA1
000101b	—	—	—	—	TMCI1	TMO3	—
001010b	TXD2 SMOSI2 SSDA2	SCK2	RXD2 SMISO2 SSCL2	—	—	TXD7 SMOSI7 SSDA7	SCK7
001011b	—	—	—	—	CTS2# RTS2# SS2#	—	—
001101b	SSLB1-A	SSLB2-A	SSLB3-A	—	MOSIC-B	MISOC-B	RSPCKC-B
010000b	—	—	—	—	CTX1	CRX1	—
010111b	—	—	—	SSIRXD0	—	—	—
011000b	—	—	—	—	EDACK0	EDREQ0	EDACK1
101011b	—	—	—	TS12	—	—	—

—: Do not specify this value.

Table 23.11 Register Settings for Input/Output Pin Function in 100-Pin TFLGA, 100-Pin LFQFP

	Pin					
PSEL[5:0] Settings	**P50**	**P51**	**P52**	**P53**	**P54**	**P55**
000000b (initial value)	Hi-Z					
000001b	—	—	—	—	MTIOC4B	MTIOC4D
000101b	—	—	—	—	TMCI1	TMO3
001010b	TXD2 SMOSI2 SSDA2	SCK2	RXD2 SMISO2 SSCL2	—	—	—
001011b	—	—	—	—	CTS2# RTS2# SS2#	—
001101b	SSLB1-A	SSLB2-A	SSLB3-A	—	MOSIC-B	MISOC-B
010000b	—	—	—	—	CTX1	CRX1
010111b	—	—	—	SSIRXD0	—	—
011000b	—	—	—	—	EDACK0	EDREQ0
101011b	—	—	—	TS12	—	—

—: Do not specify this value.

Table 23.12 Register Settings for Input/Output Pin Function in 64-Pin TFBGA, 64-Pin LFQFP, 48-Pin HWQFN

	Pin
PSEL[5:0] Settings	**P53**
000000b (initial value)	Hi-Z
010111b	SSIRXD0
101011b	TS12

23.2.8 P6n Pin Function Control Register (P6nPFS) (n = 0 to 7)

Address(es): P60PFS 0008 C170h, P61PFS 0008 C171h, P62PFS 0008 C172h, P63PFS 0008 C173h, P64PFS 0008 C174h, P65PFS 0008 C175h, P66PFS 0008 C176h, P67PFS 0008 C177h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

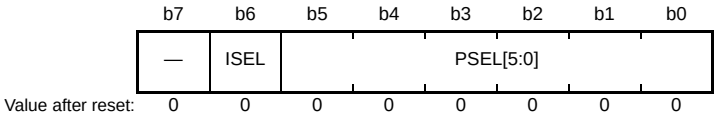
Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 23.13.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P60: IRQ0 (145/144 pin) P61: IRQ1 (145/144 pin) P62: IRQ2 (145/144 pin) P63: IRQ3 (145/144 pin) P64: IRQ4 (145/144 pin) P65: IRQ13 (145/144 pin) P66: IRQ14 (145/144 pin) P67: IRQ15 (145/144 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 23.13 Register Settings for Input/Output Pin Function in 145-Pin TFLGA, 144-Pin LFQFP

	Pin	
PSEL[5:0] Settings	**P66**	**P67**
000000b (initial value)	Hi-Z	
001000b	MTIOC7D	MTIOC7C

23.2.9 P7n Pin Function Control Register (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
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 23.14.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P70: IRQ0 (145/144 pin) P71: IRQ1 (145/144 pin)*1 P72: IRQ10 (145/144 pin)*1 P73: IRQ8 (145/144 pin) P74: IRQ12 (145/144 pin) P75: IRQ13 (145/144 pin) P76: IRQ14 (145/144 pin) P77: IRQ7 (145/144 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. The 145-pin TFLGA (0.65-mm pitch) product does not have the P71 and P72 pins.

Table 23.14 Register Settings for Input/Output Pin Function in 145-Pin TFLGA, 144-Pin LFQFP

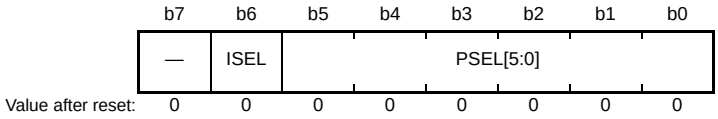
	Pin				
PSEL[5:0] Settings	**P73**	**P74**	**P75**	**P76**	**P77**
000000b (initial value)	Hi-Z				
000110b	PO16	PO19	PO20	PO22	PO23
001010b	—	—	SCK11	RXD11 SMISO11 SSCL11	TXD11 SMOSI11 SSDA11
001011b	—	CTS11# SS11#	RTS11#	—	—
010001b*1	USB1_VBUS	—	—	—	—
010010b*1	USB1_VBUSEN	—	USB1_OVRCURA	—	—
010011b*1	USB1_OVRCURB	USB1_VBUSEN	—	—	USB1_ID
011010b	—	—	SDHI_D2-A	SDHI_CMD-A	SDHI_CLK-A
011011b	—	—	—	QSSL-A	QSPCLK-A
101100b	—	—	SCK011	RXD011 SMISO011 SSCL011	TXD011 SMOSI011 SSDA011
101101b	—	CTS011# SS011#	RTS011#	—	—
101110b	—	—	DE011	—	—

—: Do not specify this value.

Note 1. Only the 145-pin TFLGA (0.65-mm pitch) product supports this setting.

23.2.10 P8n Pin Function Control Register (P8nPFS) (n = 0 to 3, 6, 7)

Address(es): P80PFS 0008 C180h, P81PFS 0008 C181h, P82PFS 0008 C182h, P83PFS 0008 C183h, P86PFS 0008 C186h, P87PFS 0008 C187h



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 23.15.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P80: IRQ8 (145/144 pin) P81: IRQ9 (145/144 pin) P82: IRQ2 (145/144 pin) P83: IRQ3 (145/144 pin) P86: IRQ14 (145/144 pin) P87: IRQ15 (145/144 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 23.15 Register Settings for Input/Output Pin Function in 145-Pin TFLGA, 144-Pin LFQFP

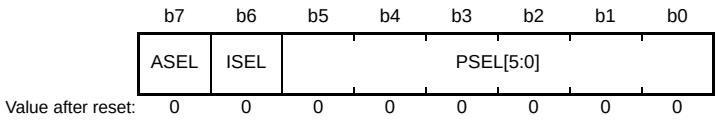
	Pin					
PSEL[5:0] Settings	**P80**	**P81**	**P82**	**P83**	**P86**	**P87**
000000b (initial value)	Hi-Z					
000001b	MTIOC3B	MTIOC3D	MTIOC4A	MTIOC4C	—	—
000011b	—	—	—	—	TIOCA0	TIOCA2
000110b	PO26	PO27	PO28	—	—	—
001000b	—	—	—	—	MTIOC4D	MTIOC4C
001010b	SCK10	RXD10 SMISO10 SSCL10	TXD10 SMOSI10 SSDA10	SCK10	RXD10 SMISO10 SSCL10	TXD10 SMOSI10 SSDA10
001011b	RTS10#	—	—	CTS10# SS10#	—	—
010011b*1	USB1_EXICEN	USB1_OVRCURB	USB1_VBUSEN	—	—	—
011000b	EDREQ0	EDACK0	EDREQ1	EDACK1	—	—
011010b	SDHI_WP	SDHI_CD	—	—	—	SDHI_D2-C
011011b	QIO2-A	QIO3-A	—	—	—	—
101100b	SCK010	RXD010 SMISO010 SSCL010	TXD010 SMOSI010 SSDA010	SCK010	RXD010 SMISO010 SSCL010	TXD010 SMOSI010 SSDA010
101101b	RTS010#	—	—	CTS010# SS010#	—	—
101110b	DE010	—	—	—	—	—

—: Do not specify this value.

Note 1. Only the 145-pin TFLGA (0.65-mm pitch) product supports this setting.

23.2.11 P9n Pin Function Control Register (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
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 23.16.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P90: IRQ0 (145/144 pin) P91: IRQ9 (145/144 pin) P92: IRQ10 (145/144 pin) P93: IRQ11 (145/144 pin)	R/W
b7	ASEL	Analog Function Select	0: Used other than as analog pin. 1: Used as analog pin. P90: AN108 (145/144 pin)	R/W

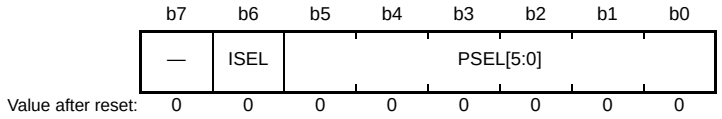
Table 23.16 Register Settings for Input/Output Pin Function in 145-Pin TFLGA, 144-Pin LFQFP

	Pin			
PSEL[5:0] Settings	**P90**	**P91**	**P92**	**P93**
000000b (initial value)	Hi-Z			
001000b	—	—	POE4#	POE0#
001010b	TXD7 SMOSI7 SSDA7	SCK7	RXD7 SMISO7 SSCL7	—
001011b	—	—	—	CTS7# RTS7# SS7#

—: Do not specify this value.

23.2.12 PAn Pin Function Control Register (PAnPFS) (n = 0 to 7)

Address(es): PA0PFS 0008 C190h, PA1PFS 0008 C191h, PA2PFS 0008 C192h, PA3PFS 0008 C193h, PA4PFS 0008 C194h, PA5PFS 0008 C195h, PA6PFS 0008 C196h, PA7PFS 0008 C197h



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 23.17, Table 23.18, and Table 23.19.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PA0: IRQ0 (145/144/100 pin) PA1: IRQ11 (145/144/100/64/48 pin) PA2: IRQ10 (145/144/100/64/48 pin) PA3: IRQ6-DS (145/144/100 pin) PA4: IRQ5-DS (145/144/100/64/48 pin) PA5: IRQ5 (145/144/100 pin) PA6: IRQ14 (145/144/100/64/48 pin) PA7: IRQ7 (145/144/100/64 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 23.17 Register Settings for Input/Output Pin Function in 145-/100-Pin TFLGA, 144-/100-Pin LFQFP

	Pin							
PSEL[5:0] Settings	**PA0**	**PA1**	**PA2**	**PA3**	**PA4**	**PA5**	**PA6**	**PA7**
000000b (initial value)	Hi-Z							
000001b	MTIOC4A	MTIOC0B	—	MTIOC0D	MTIC5U	—	MTIC5V	—
000010b	—	MTCLKC	—	MTCLKD	MTCLKA	—	MTCLKB	—
000011b	TIOCA0	TIOCB0	—	TIOCD0	TIOCA1	TIOCB1	TIOCA2	TIOCB2
000100b	—	—	—	TCLKB	—	—	—	—
000101b	—	—	—	—	TMRI0	—	TMCI3	—
000110b	PO16	PO17	PO18	PO19	PO20	PO21	PO22	PO23
000111b	CACREF	—	—	—	—	—	POE10#	—
001000b	MTIOC6D	MTIOC7B	MTIOC7A	—	—	MTIOC6B	—	—
001010b	—	SCK5	RXD5 SMISO5 SSCL5	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	—	—	—
001011b	—	—	—	—	—	—	CTS5# RTS5# SS5#	—
001100b	—	SCK12	RXD12 SMISO12 SSCL12 RXDX12	—	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	—	CTS12# RTS12# SS12#	—
001101b	SSLA1-B	SSLA2-B	SSLA3-B	—	SSLA0-B	RSPCKA-B	MOSIA-B	MISOA-B
001110b	SSL01-B	SSL02-B	SSL03-B	—	SSL00-B	RSPCK0-B	MOSI0-B	MISO0-B
110001b	—	SDHI_CD	SDHI_WP	—	—	—	—	—

—: Do not specify this value.

Table 23.18 Register Settings for Input/Output Pin Function in 64-Pin TFBGA, 64-Pin LFQFP

	Pin				
PSEL[5:0] Settings	**PA1**	**PA2**	**PA4**	**PA6**	**PA7**
000000b (initial value)	Hi-Z				
000001b	MTIOC0B	—	MTIC5U	MTIC5V	—
000010b	MTCLKC	—	MTCLKA	MTCLKB	—
000011b	TIOCB0	—	TIOCA1	TIOCA2	TIOCB2
000101b	—	—	TMRI0	TMCI3	—
000111b	—	—	—	POE10#	—
001000b	MTIOC7B	MTIOC7A	—	—	—
001010b	SCK5	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	—	—
001011b	—	—	—	CTS5# RTS5# SS5#	—
001100b	SCK12	RXD12 SMISO12 SSCL12 RXDX12	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	CTS12# RTS12# SS12#	—
001101b	SSLA2-B	SSLA3-B	SSLA0-B	MOSIA-B	MISOA-B
001110b	SSL02-B	SSL03-B	SSL00-B	MOSI0-B	MISO0-B
110001b	SDHI_CD	SDHI_WP	—	—	—

—: Do not specify this value.

Table 23.19 Register Settings for Input/Output Pin Function in 48-Pin HWQFN

	Pin			
PSEL[5:0] Settings	**PA1**	**PA2**	**PA4**	**PA6**
000000b (initial value)	Hi-Z			
000001b	MTIOC0B	—	MTIC5U	MTIC5V
000010b	MTCLKC	—	MTCLKA	MTCLKB
000011b	TIOCB0	—	TIOCA1	TIOCA2
000101b	—	—	TMRI0	TMCI3
000111b	—	—	—	POE10#
001000b	MTIOC7B	MTIOC7A	—	—
001010b	SCK5	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	—
001011b	—	—	—	CTS5# RTS5# SS5#
001100b	SCK12	RXD12 SMISO12 SSCL12 RXDX12	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	CTS12# RTS12# SS12#
001101b	SSLA2-B	SSLA3-B	SSLA0-B	MOSIA-B
001110b	SSL02-B	SSL03-B	SSL00-B	MOSI0-B
110001b	SDHI_CD	SDHI_WP	—	—

—: Do not specify this value.

23.2.13 PBn Pin Function Control Register (PBnPFS) (n = 0 to 7)

Address(es): PB0PFS 0008 C198h, PB1PFS 0008 C199h, PB2PFS 0008 C19Ah, PB3PFS 0008 C19Bh, PB4PFS 0008 C19Ch, PB5PFS 0008 C19Dh, PB6PFS 0008 C19Eh, PB7PFS 0008 C19Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 23.20 and Table 23.21.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PB0: IRQ12 (145/144/100 pin) PB1: IRQ4-DS (145/144/100 pin) PB2: IRQ2 (145/144/100 pin) PB3: IRQ3 (145/144/100 pin) PB4: IRQ4 (145/144/100 pin) PB5: IRQ13 (145/144/100/64/48 pin) PB6: IRQ6 (145/144/100/64/48 pin) PB7: IRQ15 (145/144/100/64/48 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 23.20 Register Settings for Input/Output Pin Function in 145-/100-Pin TFLGA, 144-/100-Pin LFQFP

	Pin							
PSEL[5:0] Settings	**PB0**	**PB1**	**PB2**	**PB3**	**PB4**	**PB5**	**PB6**	**PB7**
000000b (initial value)	Hi-Z							
000001b	MTIC5W	MTIOC0C	—	MTIOC0A	—	MTIOC2A	MTIOC3D	MTIOC3B
000010b	—	MTIOC4C	—	MTIOC4A	—	MTIOC1B	—	—
000011b	TIOCA3	TIOCB3	TIOCC3	TIOCD3	TIOCA4	TIOCB4	TIOCA5	TIOCB5
000100b	—	—	TCLKC	TCLKD	—	—	—	—
000101b	—	TMCI0	—	TMO0	—	TMRI1	—	—
000110b	PO24	PO25	PO26	PO27	PO28	PO29	PO30	PO31
000111b	—	—	—	POE11#	—	POE4#	—	—
001010b	RXD4*1 SMISO4*1 SSCL4*1	TXD4*1 SMOSI4*1 SSDA4*1	CTS4#*1 RTS4#*1 SS4#*1	SCK4*1	—	SCK9	RXD9 SMISO9 SSCL9	TXD9 SMOSI9 SSDA9
001011b	RXD6 SMISO6 SSCL6	TXD6 SMOSI6 SSDA6	CTS6# RTS6# SS6#	SCK6	CTS9# RTS9# SS9#	—	—	—
100100b	—	—	—	—	CTS11# RTS11# SS11#	SCK11	RXD11 SMISO11 SSCL11	TXD11 SMOSI11 SSDA11
101100b	—	—	—	—	CTS011#*2 RTS011#*2 SS011#	SCK011	RXD011 SMISO011 SSCL011	TXD011 SMOSI011 SSDA011
101110b	—	—	—	—	DE011	—	—	—

—: Do not specify this value.

Note 1. This setting is not supported by 100-pin products.

Note 2. When the setting of the SCR1.CRSEP bit is 1, the PB4 pin can be used as RTS011# but not as CTS011#.

Table 23.21 Register Settings for Input/Output Pin Function in 64-Pin TFBGA, 64-Pin LFQFP, 48-Pin HWQFN

	Pin		
PSEL[5:0] Settings	**PB5**	**PB6**	**PB7**
000000b (initial value)	Hi-Z		
000001b	MTIOC2A	MTIOC3D	MTIOC3B
000010b	MTIOC1B	—	—
000011b	TIOCB4	TIOCA5	TIOCB5
000101b	TMRI1	—	—
000111b	POE4#	—	—
001010b	SCK9	RXD9 SMISO9 SSCL9	TXD9 SMOSI9 SSDA9
100100b	SCK11	RXD11 SMISO11 SSCL11	TXD11 SMOSI11 SSDA11
101100b	SCK011	RXD011 SMISO011 SSCL011	TXD011 SMOSI011 SSDA011

—: Do not specify this value.

23.2.14 PCn Pin Function Control Register (PCnPFS) (n = 0 to 7)

Address(es): PC0PFS 0008 C1A0h, PC1PFS 0008 C1A1h, PC2PFS 0008 C1A2h, PC3PFS 0008 C1A3h, PC4PFS 0008 C1A4h, PC5PFS 0008 C1A5h, PC6PFS 0008 C1A6h, PC7PFS 0008 C1A7h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 23.22, Table 23.23, and Table 23.24.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PC0: IRQ14 (145/144/100/64 pin) PC1: IRQ12 (145/144/100/64 pin) PC2: IRQ10 (145/144/100 pin) PC3: IRQ11 (145/144/100 pin) PC4: IRQ12 (145/144/100/64/48 pin) PC5: IRQ5 (145/144/100/64/48 pin) PC6: IRQ13 (145/144/100/64/48 pin) PC7: IRQ14 (145/144/100/64/48 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 23.22 Register Settings for Input/Output Pin Function in 145-/100-Pin TFLGA, 144-/100-Pin LFQFP

	Pin							
PSEL[5:0] Settings	**PC0**	**PC1**	**PC2**	**PC3**	**PC4**	**PC5**	**PC6**	**PC7**
000000b (initial value)	Hi-Z							
000001b	MTIOC3C	MTIOC3A	MTIOC4B	MTIOC4D	MTIOC3D	MTIOC3B	MTIOC3C	MTIOC3A
000010b	—	—	—	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB
000011b	TCLKC	TCLKD	TCLKA	TCLKB	—	—	—	—
000101b	—	—	—	—	TMCI1	TMRI2	TMCI2	TMO2
000110b	PO17	PO18	PO21	PO24	PO25	PO29	PO30	PO31
000111b	—	—	—	—	POE0#	—	—	CACREF
001010b	—	SCK5	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	SCK5	SCK8	RXD8 SMISO8 SSCL8	TXD8 SMOSI8 SSDA8
001011b	CTS5# RTS5# SS5#	—	—	—	CTS8# RTS8# SS8#	—	—	—
001101b	SSLA1-A	SSLA2-A	SSLA3-A	—	SSLA0-A	RSPCKA-A	MOSIA-A	MISOA-A
001110b	SSL01-A	SSL02-A	SSL03-A	—	SSL00-A	RSPCK0-A	MOSI0-A	MISO0-A
010111b	—	—	—	—	AUDIO_CLK	SSIBCK0	SSILRCK0	SSITXD0
011010b	—	—	SDHI_D3-A	SDHI_D0-A	SDHI_D1-A	—	—	—
011011b	—	—	—	QIO0-A	QIO1-A	—	—	—
011101b	—	—	—	—	—	—	TIC0	TOC0
100100b	—	—	—	—	CTS10# RTS10# SS10#	SCK10	RXD10 SMISO10 SSCL10	TXD10 SMOSI10 SSDA10
101011b	TS16	TS15	—	—	TSCAP	TS14	TS13	—
101100b	RXD011 SMISO011 SSCL011	TXD011 SMOSI011 SSDA011 TXDA011	TXDB011	—	CTS010#*1 RTS010#*1 SS010#	SCK010	RXD010 SMISO010 SSCL010	TXD010 SMOSI010 SSDA010
101110b	—	—	—	—	DE010	—	—	—

—: Do not specify this value.

Note 1. When the setting of the SCR1.CRSEP bit is 1, the PC4 pin can be used as RTS010# but not as CTS010#.

Table 23.23 Register Settings for Input/Output Pin Function in 64-Pin TFBGA, 64-Pin LFQFP

	Pin					
PSEL[5:0] Settings	**PC0**	**PC1**	**PC4**	**PC5**	**PC6**	**PC7**
000000b (initial value)	Hi-Z					
000001b	MTIOC3C	MTIOC3A	MTIOC3D	MTIOC3B	MTIOC3C	MTIOC3A
000010b	—	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB
000011b	TCLKC	TCLKD	—	—	—	—
000101b	—	—	TMCI1	TMRI2	TMCI2	TMO2
000111b	—	—	POE0#	—	—	CACREF
001010b	—	SCK5	SCK5	SCK8	RXD8 SMISO8 SSCL8	TXD8 SMOSI8 SSDA8
001011b	CTS5# RTS5# SS5#	—	CTS8# RTS8# SS8#	—	—	—
001101b	SSLA1-A	SSLA2-A	SSLA0-A	RSPCKA-A	MOSIA-A	MISOA-A
001110b	SSL01-A	SSL02-A	SSL00-A	RSPCK0-A	MOSI0-A	MISO0-A
010111b	—	—	AUDIO_CLK	SSIBCK0	SSILRCK0	SSITXD0
011010b	—	—	SDHI_D1-A	—	—	—
011011b	—	—	QIO1-A	—	—	—
011101b	—	—	—	—	TIC0	TOC0
100100b	—	—	CTS10# RTS10# SS10#	SCK10	RXD10 SMISO10 SSCL10	TXD10 SMOSI10 SSDA10
101011b	TS16	TS15	TSCAP	TS14	TS13	—
101100b	RXD011 SMISO011 SSCL011	TXD011 SMOSI011 SSDA011 TXDA011	CTS010#*1 RTS010#*1 SS010#	SCK010	RXD010 SMISO010 SSCL010	TXD010 SMOSI010 SSDA010
101110b	—	—	DE010	—	—	—

—: Do not specify this value.

Note 1. When the setting of the SCR1.CRSEP bit is 1, the PC4 pin can be used as RTS010# but not as CTS010#.

Table 23.24 Register Settings for Input/Output Pin Function in 48-Pin HWQFN

	Pin			
PSEL[5:0] Settings	**PC4**	**PC5**	**PC6**	**PC7**
000000b (initial value)	Hi-Z			
000001b	MTIOC3D	MTIOC3B	MTIOC3C	MTIOC3A
000010b	MTCLKC	MTCLKD	MTCLKA	MTCLKB
000101b	TMCI1	TMRI2	TMCI2	TMO2
000111b	POE0#	—	—	CACREF
001010b	SCK5	SCK8	RXD8 SMISO8 SSCL8	TXD8 SMOSI8 SSDA8
001011b	CTS8# RTS8# SS8#	—	—	—
001101b	SSLA0-A	RSPCKA-A	MOSIA-A	MISOA-A
001110b	SSL00-A	RSPCK0-A	MOSI0-A	MISO0-A
010111b	AUDIO_CLK	SSIBCK0	SSILRCK0	SSITXD0
011010b	SDHI_D1-A	—	—	—
011011b	QIO1-A	—	—	—
011101b	—	—	TIC0	TOC0
100100b	CTS10# RTS10# SS10#	SCK10	RXD10 SMISO10 SSCL10	TXD10 SMOSI10 SSDA10
101011b	TSCAP	TS14	TS13	—
101100b	CTS010# RTS010# SS010#	SCK010	RXD010 SMISO010 SSCL010	TXD010 SMOSI010 SSDA010
101110b	DE010	—	—	—

—: Do not specify this value.

23.2.15 PDn Pin Function Control Register (PDnPFS) (n = 0 to 7)

Address(es): PD0PFS 0008 C1A8h, PD1PFS 0008 C1A9h, PD2PFS 0008 C1AAh, PD3PFS 0008 C1ABh, PD4PFS 0008 C1ACh, PD5PFS 0008 C1ADh, PD6PFS 0008 C1AEh, PD7PFS 0008 C1AFh

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 23.25, Table 23.26, and Table 23.27.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PD0: IRQ0 (145/144/100 pin) PD1: IRQ1 (145/144/100 pin) PD2: IRQ2 (145/144/100/64/48 pin) PD3: IRQ3 (145/144/100/64/48 pin) PD4: IRQ4 (145/144/100/64/48 pin) PD5: IRQ5 (145/144/100/64/48 pin) PD6: IRQ6 (145/144/100/64 pin) PD7: IRQ7 (145/144/100/64 pin)	R/W
b7	ASEL	Analog Function Select	0: Used other than as analog pin. 1: Used as analog pin. PD0: AN107 (145/144/100 pin) PD1: AN106 (145/144/100 pin) PD2: AN105 (145/144/100/64/48 pin) PD3: AN104 (145/144/100/64/48 pin) PD4: AN103 (145/144/100/64/48 pin) PD5: AN102 (145/144/100/64/48 pin) PD6: AN101 (145/144/100/64 pin) PD7: AN100 (145/144/100/64 pin)	R/W

Table 23.25 Register Settings for Input/Output Pin Function in 145-/100-Pin TFLGA, 144-/100-Pin LFQFP

	Pin							
PSEL[5:0] Settings	**PD0**	**PD1**	**PD2**	**PD3**	**PD4**	**PD5**	**PD6**	**PD7**
000000b (initial value)	Hi-Z							
000001b	—	MTIOC4B	MTIOC4D	—	—	MTIC5W	MTIC5V	MTIC5U
000111b	—	—	—	POE8#	POE11#	POE10#	POE4#	POE0#
001000b	POE4#	POE0#	—	MTIOC8D	MTIOC8B	MTIOC8C	MTIOC8A	—
001101b	—	MOSIC-A	MISOC-A	RSPCKC-A	SSLC0-A	SSLC1-A	SSLC2-A	SSLC3-A
010000b	—	CTX0	CRX0	—	—	—	—	—
011010b	—	—	SDHI_D2-B	SDHI_D3-B	SDHI_CMD-B	SDHI_CLK-B	SDHI_D0-B	SDHI_D1-B
011011b	—	—	QIO2-B	QIO3-B	QSSL-B	QSPCLK-B	QIO0-B	QIO1-B
011101b	—	—	TIC2	TOC2	—	—	—	—

—: Do not specify this value.

Table 23.26 Register Settings for Input/Output Pin Function in 64-Pin TFBGA, 64-Pin LFQFP

	Pin					
PSEL[5:0] Settings	**PD2**	**PD3**	**PD4**	**PD5**	**PD6**	**PD7**
000000b (initial value)	Hi-Z					
000001b	MTIOC4D	—	—	MTIC5W	MTIC5V	MTIC5U
000111b	—	POE8#	POE11#	POE10#	POE4#	POE0#
001000b	—	MTIOC8D	MTIOC8B	MTIOC8C	MTIOC8A	—
011010b	SDHI_D2-B	SDHI_D3-B	SDHI_CMD-B	SDHI_CLK-B	SDHI_D0-B	SDHI_D1-B
011011b	QIO2-B	QIO3-B	QSSL-B	QSPCLK-B	QIO0-B	QIO1-B
011101b	TIC2	TOC2	—	—	—	—

—: Do not specify this value.

Table 23.27 Register Settings for Input/Output Pin Function in 48-Pin HWQFN

	Pin			
PSEL[5:0] Settings	**PD2**	**PD3**	**PD4**	**PD5**
000000b (initial value)	Hi-Z			
000001b	MTIOC4D	—	—	MTIC5W
000111b	—	POE8#	POE11#	POE10#
001000b	—	MTIOC8D	MTIOC8B	MTIOC8C
011010b	SDHI_D2-B	SDHI_D3-B	SDHI_CMD-B	SDHI_CLK-B
011011b	QIO2-B	QIO3-B	QSSL-B	QSPCLK-B
011101b	TIC2	TOC2	—	—

—: Do not specify this value.

23.2.16 PEn Pin Function Control Register (PEnPFS) (n = 0 to 7)

Address(es): PE0PFS 0008 C1B0h, PE1PFS 0008 C1B1h, PE2PFS 0008 C1B2h, PE3PFS 0008 C1B3h, PE4PFS 0008 C1B4h, PE5PFS 0008 C1B5h, PE6PFS 0008 C1B6h, PE7PFS 0008 C1B7h

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 23.28, Table 23.29, and Table 23.30.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PE0: IRQ8 (145/144/100/64 pin) PE1: IRQ9 (145/144/100/64 pin) PE2: IRQ7-DS (145/144/100/64 pin) PE3: IRQ11 (145/144/100 pin) PE4: IRQ12 (145/144/100 pin) PE5: IRQ5 (145/144/100 pin) PE6: IRQ6 (145/144/100/64/48 pin) PE7: IRQ7 (145/144/100/64/48 pin)	R/W
b7	ASEL	Analog Function Select	0: Used other than as analog pin. 1: Used as analog pin. PE0: ANEX0 (145/144/100/64 pin) PE1: ANEX1 (145/144/100/64 pin)	R/W

Table 23.28 Register Settings for Input/Output Pin Function in 145-/100-Pin TFLGA, 144-/100-Pin LFQFP

	Pin							
PSEL[5:0] Settings	**PE0**	**PE1**	**PE2**	**PE3**	**PE4**	**PE5**	**PE6**	**PE7**
000000b (initial value)	Hi-Z							
000001b	—	MTIOC4C	MTIOC4A	MTIOC4B	MTIOC4D	MTIOC4C	—	—
000010b	—	—	—	—	MTIOC1A	MTIOC2B	—	—
000110b	—	PO18	PO23	PO26	PO28	—	—	—
000111b	—	—	—	POE8#	—	—	—	—
001000b	MTIOC3D	MTIOC3B	—	—	—	—	MTIOC6C	MTIOC6A
001100b	SCK12	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	RXD12 SMISO12 SSCL12 RXDX12	CTS12# RTS12# SS12#	—	—	—	—
001101b	SSLB1-B	SSLB2-B	SSLB3-B	—	SSLB0-B	RSPCKB-B	MOSIB-B	MISOB-B
011010b	—	—	—	—	—	—	SDHI_CD	SDHI_WP
011011b	—	—	—	—	—	—	QIO0-B	QIO1-B
011101b	—	—	TIC3	TOC3	—	—	TIC1	TOC1
110001b	—	—	—	—	—	—	SDHI_D0-B	SDHI_D1-B

—: Do not specify this value.

Table 23.29 Register Settings for Input/Output Pin Function in 64-Pin TFBGA, 64-Pin LFQFP

	Pin				
PSEL[5:0] Settings	**PE0**	**PE1**	**PE2**	**PE6**	**PE7**
000000b (initial value)	Hi-Z				
000001b	—	MTIOC4C	MTIOC4A	—	—
001000b	MTIOC3D	MTIOC3B	—	MTIOC6C	MTIOC6A
001100b	SCK12	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	RXD12 SMISO12 SSCL12 RXDX12	—	—
001101b	SSLB1-B	SSLB2-B	SSLB3-B	MOSIB-B	MISOB-B
011010b	—	—	—	SDHI_CD	SDHI_WP
011011b	—	—	—	QIO0-B	QIO1-B
011101b	—	—	TIC3	TIC1	TOC1
110001b	—	—	—	SDHI_D0-B	SDHI_D1-B

—: Do not specify this value.

Table 23.30 Register Settings for Input/Output Pin Function in 48-Pin HWQFN

	Pin	
PSEL[5:0] Settings	**PE6**	**PE7**
000000b (initial value)	Hi-Z	
001000b	MTIOC6C	MTIOC6A
001101b	MOSIB-B	MISOB-B
011010b	SDHI_CD	SDHI_WP
011011b	QIO0-B	QIO1-B
011101b	TIC1	TOC1
110001b	SDHI_D0-B	SDHI_D1-B

23.2.17 PFn Pin Function Control Register (PFnPFS) (n = 5)

Address(es): PF5PFS 0008 C1BDh

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

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PF5: IRQ4 (145/144 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

23.2.18 PHn Pin Function Control Register (PHnPFS) (n = 1, 2)

Address(es): PH1PFS 0008 C1C9h, PH2PFS 0008 C1CAh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 23.31.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PH1: IRQ0 (145/144/100/64 pin) PH2: IRQ1 (145/144/100/64 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 23.31 Register Settings for Input/Output Pin Function in 145-/100-Pin TFLGA, 144-/100-/64-Pin LFQFP, 64-Pin TFBGA

	Pin	
PSEL[5:0] Settings	**PH1**	**PH2**
000000b (initial value)	Hi-Z	
000101b	TMO0	TMRI0

23.2.19 PJn Pin Function Control Register (PJnPFS) (n = 3, 5)

Address(es): PJ3PFS 0008 C1D3h, PJ5PFS 0008 C1D5h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	PSEL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	PSEL[5:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 23.32 and Table 23.33.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PJ3: IRQ11 (145/144/100 pin) PJ5: IRQ13 (145/144 pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 23.32 Register Settings for Input/Output Pin Function in 145-Pin TFLGA, 144-Pin LFQFP

	Pin	
PSEL[5:0] Settings	**PJ3**	**PJ5**
000000b (initial value)	Hi-Z	
000001b	MTIOC3C	—
001010b	CTS6# RTS6# SS6#	—
001011b	CTS0# RTS0# SS0#	CTS2# RTS2# SS2#
011000b	EDACK1	—
100001b	—	POE8#

—: Do not specify this value.

Table 23.33 Register Settings for Input/Output Pin Function in 100-Pin TFLGA, 100-Pin LFQFP

	Pin
PSEL[5:0] Settings	**PJ3**
000000b (initial value)	Hi-Z
000001b	MTIOC3C
001010b	CTS6# RTS6# SS6#
001011b	CTS0# RTS0# SS0#
011000b	EDACK1

23.2.20 CS Output Enable Register (PFCSE)

Address(es): 0008 C100h

	b7	b6	b5	b4	b3	b2	b1	b0
	CS7E	CS6E	CS5E	CS4E	CS3E	CS2E	CS1E	CS0E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CS0E	CS0 Enable	0: Disables CSn# output. 1: Enables CSn# output. (n = 0 to 7)	R/W
b1	CS1E	CS1 Enable		R/W
b2	CS2E	CS2 Enable		R/W
b3	CS3E	CS3 Enable		R/W
b4	CS4E	CS4 Enable		R/W
b5	CS5E	CS5 Enable		R/W
b6	CS6E	CS6 Enable		R/W
b7	CS7E	CS7 Enable		R/W

CSnE Bit (CSn Enable) (n = 0 to 7)

These bits enable or disable the corresponding pins to output the CSn# signal.

To enable output of the CSn# signal, set the corresponding CSnE bit in PFCSE to 1.

23.2.21 CS Output Pin Select Register 0 (PFCSS0)

Address(es): 0008 C102h

	b7	b6	b5	b4	b3	b2	b1	b0
	CS3S[1:0]		CS2S[1:0]		CS1S[1:0]		—	CS0S
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CS0S	CS0# Output Pin Select*1	0: Set P60 as CS0# output pin 1: Set PC7 as CS0# output pin	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3, b2	CS1S[1:0]	CS1# Output Pin Select*2	b3 b2 0 0: Set P61 as CS1# output pin 0 1: Set P71 as CS1# output pin 1 X: Set PC6 as CS1# output pin	R/W
b5, b4	CS2S[1:0]	CS2# Output Pin Select*3	b5 b4 0 0: Set P62 as CS2# output pin 0 1: Set P72 as CS2# output pin 1 X: Set PC5 as CS2# output pin	R/W
b7, b6	CS3S[1:0]	CS3# Output Pin Select*4	b7 b6 0 0: Set P63 as CS3# output pin 0 1: Set P73 as CS3# output pin 1 X: Set PC4 as CS3# output pin	R/W

X: Don't care

Note 1. P60 is not present in 100-pin products. When CS0# output is used, set this bit to 1.

Note 2. P61 and P71 are not present in 100-pin products. When CS1# output is used, set these bits to 1xb. P71 is not present in 145-pin TFLGA (0.65-mm pitch) product.

Note 3. P62 and P72 are not present in 100-pin products. When CS2# output is used, set these bits to 1xb. P72 is not present in 145-pin TFLGA (0.65-mm pitch) product.

Note 4. P63 and P73 are not present in 100-pin products. When CS3# output is used, set these bits to 1xb.

CS0S Bit (CS0# Output Pin Select)

CSnS[1:0] Bits (CSn# Output Pin Select) (n = 1 to 3)

When CSn# output is enabled (the PFCSE.CSnE bit = 1), the CSn# output pin is selected.

23.2.22 CS Output Pin Select Register 1 (PFCSS1)

Address(es): 0008 C103h

	b7	b6	b5	b4	b3	b2	b1	b0
	CS7S[1:0]		CS6S[1:0]		CS5S[1:0]		CS4S[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CS4S[1:0]	CS4# Output Pin Select*1	b1 b0 0 0: Set P64 as CS4# output pin 0 1: Set P74 as CS4# output pin 1 X: Set P24 as CS4# output pin	R/W
b3, b2	CS5S[1:0]	CS5# Output Pin Select*2	b3 b2 0 0: Set P65 as CS5# output pin 0 1: Set P75 as CS5# output pin 1 X: Set P25 as CS5# output pin	R/W
b5, b4	CS6S[1:0]	CS6# Output Pin Select*3	b5 b4 0 0: Set P66 as CS6# output pin 0 1: Set P76 as CS6# output pin 1 X: Set P26 as CS6# output pin	R/W
b7, b6	CS7S[1:0]	CS7# Output Pin Select*4	b7 b6 0 0: Set P67 as CS7# output pin 0 1: Set P77 as CS7# output pin 1 X: Set P27 as CS7# output pin	R/W

X: Don't care

Note 1. P64 and P74 are not present in 100-pin products. When CS4# output is used, set these bits to 1xb.
Note 2. P65 and P75 are not present in 100-pin products. When CS5# output is used, set these bits to 1xb.
Note 3. P66 and P76 are not present in 100-pin products. When CS6# output is used, set these bits to 1xb.
Note 4. P67 and P77 are not present in 100-pin products. When CS7# output is used, set these bits to 1xb.

CSnS[1:0] Bits (CSn# Output Pin Select) (n = 4 to 7)

When CSn# output is enabled (the PFCSE.CSnE bit = 1), the CSn# output pin is selected.

23.2.23 Address Output Enable Register 0 (PFAOE0)

Address(es): 0008 C104h

	b7	b6	b5	b4	b3	b2	b1	b0
	A15E	A14E	A13E	A12E	A11E	A10E	A9E	A8E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	A8E	Address A8 Output Enable	0: Disables A8 output. 1: Enables A8 output.	R/W
b1	A9E	Address A9 Output Enable	0: Disables A9 output. 1: Enables A9 output.	R/W
b2	A10E	Address A10 Output Enable	0: Disables A10 output. 1: Enables A10 output.	R/W
b3	A11E	Address A11 Output Enable	0: Disables A11 output. 1: Enables A11 output.	R/W
b4	A12E	Address A12 Output Enable	0: Disables A12 output. 1: Enables A12 output.	R/W
b5	A13E	Address A13 Output Enable	0: Disables A13 output. 1: Enables A13 output.	R/W
b6	A14E	Address A14 Output Enable	0: Disables A14 output. 1: Enables A14 output.	R/W
b7	A15E	Address A15 Output Enable	0: Disables A15 output. 1: Enables A15 output.	R/W

23.2.24 Address Output Enable Register 1 (PFAOE1)

Address(es): 0008 C105h

	b7	b6	b5	b4	b3	b2	b1	b0
	A23E	A22E	A21E	A20E	A19E	A18E	A17E	A16E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	A16E	Address A16 Output Enable	0: Disables A16 output. 1: Enables A16 output.	R/W
b1	A17E	Address A17 Output Enable	0: Disables A17 output. 1: Enables A17 output.	R/W
b2	A18E	Address A18 Output Enable	0: Disables A18 output. 1: Enables A18 output.	R/W
b3	A19E	Address A19 Output Enable	0: Disables A19 output. 1: Enables A19 output.	R/W
b4	A20E	Address A20 Output Enable	0: Disables A20 output. 1: Enables A20 output.	R/W
b5	A21E	Address A21 Output Enable	0: Disables A21 output. 1: Enables A21 output.	R/W
b6	A22E	Address A22 Output Enable	0: Disables A22 output. 1: Enables A22 output.	R/W
b7	A23E	Address A23 Output Enable	0: Disables A23 output. 1: Enables A23 output.	R/W

23.2.25 External Bus Control Register 0 (PFBCR0)

Address(es): 0008 C106h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	WR1BC1E	—	DHE	BCLKO	ADRHMS2	ADRHMS	ADRLE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADRLE	A0 to A7 Output Enable	0: Configures PA0 to PA7 as the I/O port pins. 1: Configures PA0 to PA7 as the external address buses A0 to A7.	R/W
b1	ADRHMS	A16 to A23 Output Enable	See Table 23.34.	R/W
b2	ADRHMS2	A16 to A23 Output Enable 2		R/W
b3	BCLKO	BCLK Forced Output	0: BCLK is output when EXBE = 1 and not output when EXBE = 0. 1: BCLK is output regardless of the setting of EXBE.	R/W
b4	DHE	D8 to D15 Output Enable	0: Configures PE0 to PE7 as the I/O port pins. 1: Configures PE0 to PE7 as the external data bus D8 to D15.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	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	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

BCLKO Bit (BCLK Forced Output)

This bit enables or disables forced output on the BCLK pin.

When this bit is set to 0, BCLK output is enabled or disabled according to the setting of the EXBE bit; when this bit is 1, BCLK is output regardless of the setting of the EXBE bit.

Note that if the bit is set to 1, BCLK is output regardless of the PMR register.

[Setting procedure]

Output enabled: PSTOP1 (stopped) → BCLKO = 1 → PSTOP1 (operating)

Output disabled: PSTOP1 (operating) → PSTOP1 (stopped) → BCLKO = 0

Table 23.34 Settings for External Address Buses A16 to A23

ADRHMS Bit	ADRHMS2 Bit	Settings for External Address Buses A16 to A23
0	0	Set PC0 to PC7.
0	1	Set PC0, PC1, P71, P72, P74, and PC5 to PC7.*1
1	0	Set P90 to P93. (No allocation of A20 to A23)
1	1	Setting prohibited

Note 1. Since 145-pin TFLGA (0.65-mm pitch) product does not have the P71 and P72 pins, set the ADRHMS and ADRHMS2 bits to 00b or 10b for use of the external address buses A16 to A23.

23.2.26 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: Setting invalid*1 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: Disables ALE pin output. 1: Enables ALE pin output.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	MDSDE	SDRAM Pin Enable	0: Disables SDRAM pin output. (CKE, SDCS#, RAS#, CAS#, WE#, and DQM0) 1: Enables SDRAM pin output. (CKE, SDCS#, RAS#, CAS#, WE#, and DQM0)	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	DQM1E	DQM1 Enable	0: Disables DQM1 output 1: Enables DQM1 output	R/W
b7	SDCLKE	SDCLK Enable	0: Disables SDCLK output 1: Enables SDCLK output	R/W

Note 1. Even if these bits are set to “00b” in 145-, 144-, and 100-pin products, P55 is set as WAIT# input pin.

WAITS[1:0] Bits (WAIT Select)

When using an external wait when the external bus is enabled, set the external wait enable bit (CSnMOD.EWENB) of the CSn mode register to 1 (external wait enable), and set the WAITS[1:0] bits to something other than 00 (select the WAIT# pin) and set 0 to each bit of the PMR and PDR registers of the corresponding pin.
When using a function other than the WAIT# pin when the external bus is enabled, set the external wait permission bit (CSnMOD.EWENB) of the CSn mode register to 0 (external wait prohibition) and set 00 to the WAITS[1:0] bits.

ALEOE Bit (ALE Output Enable)

This bit enables or disables output of the ALE pin.

MDSDE Bit (SDRAM Pin Enable)

This bit enables or disables output of the SDRAM pin (CKE, SDCS#, RAS#, CAS#, WE#, and DQM0).
The DQM1 pin is enabled or disabled individually by the DQM1E bit, while the MDSDE bit is 1. The SDCLK pin is enabled or disabled independently by the SDCLKE bit, regardless of the MDSDE setting.

DQM1E Bit (DQM1 Enable)

This bit enables or disables output of the DQM1 pin.
When the MDSDE bit is set to 1, setting of the DQM1E bit is enabled. When the MDSDE bit is set to 0, setting of the DQM1E bit has no effect.

SDCLKE Bit (SDCLK Enable)

This bit enables or disables output of the SDCLK pin.
The SCKCR.PSTOP0 bit should be set to 1 before changing the setting of the SDCLKE bit.

23.2.27 External Bus Control Register 2 (PFBCR2)

Address(es): 0008 C108h

	b7	b6	b5	b4	b3	b2	b1	b0
	D3S[1:0]		D2S[1:0]		D1S[1:0]		D0S[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	D0S[1:0]	D0 Selection	b1 b0 0 0: PD0 is set as D0 pin. 0 1: PE0 is set as D0 pin.*1 1 0: P55 is set as D0 pin. 1 1: P61 is set as D0 pin.*2	R/W
b3, b2	D1S[1:0]	D1 Selection	b3 b2 0 0: PD1 is set as D1 pin. 0 1: PE1 is set as D1 pin.*1 1 0: P54 is set as D1 pin. 1 1: P62 is set as D1 pin.*2	R/W
b5, b4	D2S[1:0]	D2 Selection	b5 b4 0 0: PD2 is set as D2 pin. 0 1: PE2 is set as D2 pin.*1 1 0: PC6 is set as D2 pin. 1 1: P63 is set as D2 pin.*2	R/W
b7, b6	D3S[1:0]	D3 Selection	b7 b6 0 0: PD3 is set as D3 pin. 0 1: PE3 is set as D3 pin.*1 1 0: PC5 is set as D3 pin. 1 1: P64 is set as D3 pin.*2	R/W

Note 1. Do not make this setting if PE0 to PE7 are to be set to operate as external data bus pins D8 to D15 and the setting of PFBCR0.DHE is 1.

Note 2. Since 100-pin products does not have PORT6, set the bits to a value other than 11b when the given Dx input and output pin (x = 0 to 3) is to be used.

23.2.28 External Bus Control Register 3 (PFBCR3)

Address(es): 0008 C109h

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

Bit	Symbol	Bit Name	Description	R/W
b0	DLHS	D4 to D7 Selection	0: PD4 to PD7 are set as pins D4 to D7 1: PE4 to PE7 are set as pins D4 to D7.*1	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Do not make this setting if PE0 to PE7 are to be set to operate as external data bus pins D8 to D15 and the setting of PFBCR0.DHE is 1.

23.3 How to Set the External Bus Interface

If the external bus interface is to be used, set the MPC registers according to Table 23.35 and then set the external bus enable bit (EXBE) in system control register 0 (SYSCR0) to 1.
Table 23.35 lists how to set up port pins to act as the external bus interface. For details on the relevant registers of the MPC, refer to section 23.2, Register Descriptions.

Table 23.35 How to Set the External Bus Interface (1/3)

Port	Output Signal	Settings of MPC Registers	
		145-Pin, 144-Pin	100-Pin
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	
	D1[A1/D1]	PFBCR2.D1S[1:0] = 10	
P55	WAIT#	PFBCR1.WAITS[1:0] = 01	
	D0[A0/D0]	PFBCR2.D0S[1:0] = 10	
P60	CS0#	PFCSE.CS0E = 1, PFCSS0.CS0S = 0	(not provided)
P61	CS1#	PFCSE.CS1E = 1, PFCSS0.CS1S[1:0] = 00	(not provided)
	SDCS#	PFBCR1.MDSDE = 1	(not provided)
	D0[A0/D0]	PFBCR2.D0S[1:0] = 11	(not provided)
P62	CS2#	PFCSE.CS2E = 1, PFCSS0.CS2S[1:0] = 00	(not provided)
	RAS#	PFBCR1.MDSDE = 1	(not provided)
	D1[A1/D1]	PFBCR2.D1S[1:0] = 11	(not provided)
P63	CS3#	PFCSE.CS3E = 1, PFCSS0.CS3S[1:0] = 00	(not provided)
	CAS#	PFBCR1.MDSDE = 1	(not provided)
	D2[A2/D2]	PFBCR2.D2S[1:0] = 11	(not provided)
P64	CS4#	PFCSE.CS4E = 1, PFCSS1.CS4S[1:0] = 00	(not provided)
	WE#	PFBCR1.MDSDE = 1	(not provided)
	D3[A3/D3]	PFBCR2.D3S[1:0] = 11	(not provided)
P65	CS5#	PFCSE.CS5E = 1, PFCSS1.CS5S[1:0] = 00	(not provided)
	CKE	PFBCR1.MDSDE = 1	(not provided)
P66	CS6#	PFCSE.CS6E = 1, PFCSS1.CS6S[1:0] = 00	(not provided)
	DQM0	PFBCR1.MDSDE = 1	(not provided)
P67	CS7#	PFCSE.CS7E = 1, PFCSS1.CS7S[1:0] = 00	(not provided)
	DQM1	PFBCR1.MDSDE = 1, PFBCR1.DQM1E = 1	(not provided)
P70	SDCLK	PFBCR1.SDCLKE = 1	(not provided)
P71*1	CS1#	PFCSE.CS1E = 1, PFCSS0.CS1S[1:0] = 01	(not provided)
	A18	PFAOE1.A18E = 1, PFBCR0.ADRHMS = 0, PFBCR0.ADRHMS2 = 1	(not provided)
P72*1	CS2#	PFCSE.CS2E = 1, PFCSS0.CS2S[1:0] = 01	(not provided)
	A19	PFAOE1.A19E = 1, PFBCR0.ADRHMS = 0, PFBCR0.ADRHMS2 = 1	(not provided)

Table 23.35 How to Set the External Bus Interface (2/3)

Port	Output Signal	Settings of MPC Registers	
		145-Pin, 144-Pin	100-Pin
P73	CS3#	PFCSE.CS3E = 1, PFCSS0.CS3S[1:0] = 01	(not provided)
P74	CS4#	PFCSE.CS4E = 1, PFCSS1.CS4S[1:0] = 01	(not provided)
	A20	PFAOE1.A20E = 1, PFBCR0.ADRHMS = 0, PFBCR0.ADRHMS2 = 1	(not provided)
P75	CS5#	PFCSE.CS5E = 1, PFCSS1.CS5S[1:0] = 01	(not provided)
P76	CS6#	PFCSE.CS6E = 1, PFCSS1.CS6S[1:0] = 01	(not provided)
P77	CS7#	PFCSE.CS7E = 1, PFCSS1.CS7S[1:0] = 01	(not provided)
P90	A16	PFAOE1.A16E = 1, PFBCR0.ADRHMS = 1, PFBCR0.ADRHMS2 = 0	(not provided)
P91	A17	PFAOE1.A17E = 1, PFBCR0.ADRHMS = 1, PFBCR0.ADRHMS2 = 0	(not provided)
P92	A18	PFAOE1.A18E = 1, PFBCR0.ADRHMS = 1, PFBCR0.ADRHMS2 = 0	(not provided)
P93	A19	PFAOE1.A19E = 1, PFBCR0.ADRHMS = 1, PFBCR0.ADRHMS2 = 0	(not provided)
PA0	A0	PFBCR0.ADRLE = 1, CSnMOD.WRMOD = 0	
	BC0#	PFBCR0.ADRLE = 1, CSnMOD.WRMOD = 1	
PA1	A1	PFBCR0.ADRLE = 1	
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, PFBCR0.ADRHMS2 = 0	
PC3	A19	PFAOE1.A19E = 1, PFBCR0.ADRHMS = 0, PFBCR0.ADRHMS2 = 0	
PC4	A20	PFAOE1.A20E = 1, PFBCR0.ADRHMS = 0, PFBCR0.ADRHMS2 = 0	
	CS3#	PFCSE.CS3E = 1, PFCSS0.CS3S[1:0] = 10/11	
PC5	A21	PFAOE1.A21E = 1, PFBCR0.ADRHMS = 0	
	CS2#	PFCSE.CS2E = 1, PFCSS0.CS2S[1:0] = 10/11	
	WAIT#	PFBCR1.WAITS[1:0] = 10	
	D3[A3/D3]	PFBCR2.D3S[1:0] = 10	
PC6	A22	PFAOE1.A22E = 1, PFBCR0.ADRHMS = 0	
	CS1#	PFCSE.CS1E = 1, PFCSS0.CS1S[1:0] = 10/11	
	D2[A2/D2]	PFBCR2.D2S[1:0] = 10	
PC7	A23	PFAOE1.A23E = 1, PFBCR0.ADRHMS = 0	
	CS0#	PFCSE.CS0E = 1, PFCSS0.CS0S = 1	
PD0	D0[A0/D0]	PFBCR2.D0S[1:0] = 00	
PD1	D1[A1/D1]	PFBCR2.D1S[1:0] = 00	

Table 23.35 How to Set the External Bus Interface (3/3)

Port	Output Signal	Settings of MPC Registers	
		145-Pin, 144-Pin	100-Pin
PD2	D2[A2/D2]	PFBCR2.D2S[1:0] = 00	
PD3	D3[A3/D3]	PFBCR2.D3S[1:0] = 00	
PD4	D4[A4/D4]	PFBCR3.DLHS = 0	
PD5	D5[A5/D5]	PFBCR3.DLHS = 0	
PD6	D6[A6/D6]	PFBCR3.DLHS = 0	
PD7	D7[A7/D7]	PFBCR3.DLHS = 0	
PE0	D8[A8/D8]	PFBCR0.DHE = 1	
	D0[A0/D0]	PFBCR2.D0S[1:0] = 01	
PE1	D9[A9/D9]	PFBCR0.DHE = 1	
	D1[A1/D1]	PFBCR2.D1S[1:0] = 01	
PE2	D10[A10/D10]	PFBCR0.DHE = 1	
	D2[A2/D2]	PFBCR2.D2S[1:0] = 01	
PE3	D11[A11/D11]	PFBCR0.DHE = 1	
	D3[A3/D3]	PFBCR2.D3S[1:0] = 01	
PE4	D12[A12/D12]	PFBCR0.DHE = 1	
	D4[A4/D4]	PFBCR3.DLHS = 1	
PE5	D13[A13/D13]	PFBCR0.DHE = 1	
	D5[A5/D5]	PFBCR3.DLHS = 1	
PE6	D14[A14/D14]	PFBCR0.DHE = 1	
	D6[A6/D6]	PFBCR3.DLHS = 1	
PE7	D15[A15/D15]	PFBCR0.DHE = 1	
	D7[A7/D7]	PFBCR3.DLHS = 1	

Note 1. The 145-pin TFLGA (0.65-mm pitch) product does not have the P71 and P72 pins.

23.4 Usage Notes

23.4.1 Procedure for Specifying Input/Output Pin Function

Use the following procedure to specify the input/output pin functions.

(1) Clear the port mode register (PMR) for the target pin to 0 to select the general I/O port.
(2) Specify the assignments of input/output signals for peripheral modules to the desired pins.
(3) Enable writing to the Pmn pin function control register (PmnPFS) through the write-protect register (PWPR) setting. (m = 0 to 9, A to F, H, and J, n = 0 to 7)
(4) Specify the input/output function for the pin through the PSEL[5:0] bit settings in the PmnPFS register.
(5) Clear the PFSWE bit in the PWPR register to 0 to disable writing to the PmnPFS register.
(6) Set the PMR to 1 as necessary to switch to the selected input/output function for the pin.

23.4.2 Notes on MPC Register Setting

(1) Settings of the Pmn pin function control register (PmnPFS) should be made only while the PMR register for the target pin is set to 0. If the PmnPFS register is set while the PMR register is 1, unexpected edges may be input through the input pin or unexpected pulses are output through the output pin.
(2) Only the allowed values (functions) should be specified in the Pmn pin function control registers. If a value that is not allowed for the register is specified, correct operation is not guaranteed.
(3) Do not assign a single function to multiple pins through the MPC settings.
(4) Ports 0, 4, 9, D, and E also function as analog Input/output pins for the A/D converter. When using these ports as analog Input/output pins, set them to general input pins by clearing the corresponding bits in the port mode register (PMR) and port direction register (PDR) to 0 and set the PmnPFS.ASEL bit to 1, to avoid degradation of accuracy.
(5) The initial value of the time capture event input pin enable bit (TCEN) of the time capture control register y (RTCCRy, y = 0 to 2) is undefined after a reset. Therefore, set this bit to 0 to avoid unnecessary input.
(6) Points to note regarding the port mode register (PMR), port direction register (PDR), and the PmnPFS register settings for pins that have multiplexed pin functions are listed in Table 23.36. The pin states are readable if the value of the ASEL bit is 0. Ensure that the PMR.Bj bit is 0 when changes to the PSEL[5:0] bits are made.

Table 23.36 Register Settings

Item	PMR.Bn	PDR.Bn	PmnPFS			Point to Note
			ASEL	ISEL	PSEL[5:0]	
After a reset	0	0	0	0	000000b	Pins function as general input port pins after release from the reset state.
General input ports	0	0	0	0/1	x	Set the PmnPFS.ISEL bit to 1 if these are multiplexed with interrupt inputs.
General output ports	0	1	0	0	x	
Peripheral functions	1	x	0	0/1	Peripheral functions (see Table 23.2 to Table 23.33)	Set the PmnPFS.ISEL bit to 1 if these are multiplexed with interrupt inputs.
Interrupt inputs	0	0	0	1	x	
NMI	x	x	x	x*1	x	Register settings are not required.
Analog inputs and outputs	0	0	1	x*1	x	Set these as general input port pins so that the output buffers are turned off.
Time-capture event-input pins	0	0	x	0/1	x	Set these as general input port pins so that the output buffers are turned off.
External bus	0	x*2	0	0	x	Set the PMR.Bn bit to 0 and do not select the peripheral function.
JTAG interface	0	x	x	0	x	Set the PMR.Bn bit and the PmnPFS.ISEL bit to 0 and switch the input buffers off.
FINE interface	0	x	x	0	x	Set the PMR.Bn bit and the PmnPFS.ISEL bit to 0 and switch the input buffers off.
EXTAL/XTAL	0	0	x	x*1	x	Set these as general input port pins so that the output buffers are turned off.

x: Setting not required.
0/1: Setting the PmnPFS.ISEL bit to 0 makes the pin incapable of functioning as an IRQ pin.
Setting the PmnPFS.ISEL bit to 1 makes the pin capable of functioning as an IRQ pin (When a IRQ is assigned).
Note 1. Even if the PmnPFS.ISEL bit is set to 1, the pin will not function as an IRQn input pin.
Note 2. To use the WAIT# input pin, set the corresponding bit in the PORTm.PDR register to 0.

Note:
- The pin state is readable when the PmnPFS.ASEL bit is 0.
- If the value of the PmnPFS.PSEL[5:0] bits is to be changed, do so while the PMR.Bn bit is 0.
- If an RIIC function is assigned to a port pin, clear the PMR.Bn (to 0); pulling up is automatically turned off for outputs from peripheral modules other than the RIIC.
- If an input pin for time-capture events is not in use, clear the time capture event input pin enable bit (TCEN) in time capture control register y (RTCCRy) (y = 0 to 2) to 0 (disabled). The value of the RTCCRy.TCEN bit after a reset is undefined.
- Do not make settings to assign multiple external bus signals to a single pin.

23.4.3 Notes on the Use of Analog Functions

To use an analog function, set the corresponding bits in both the port mode register (PMR) and port direction register (PDR) to 0 so that the pin acts as a general input port. After that, set the pin function select bit in the Pmn pin function control register (PmnPFS.ASEL) to 1.

23.4.4 Notes on Using the Capacitive Touch Sensing Unit (CTSU)

When using the pins (TSn (n = 0 to 16) and TSCAP) of the capacitive touch sensing unit (CTSU), set the given bits of the port mode register (PMR), the port direction register (PDR), and the pull-up control register (PCR) to 0. Then, use the PmnPFS.PSEL[5:0] bits to select the CTSU pins and set the PMR register to 1. When a pin function of the capacitive touch sensing unit is to be used, do not use the pin as the IRQ input pin regardless of the ISEL setting of the corresponding bit.

24. Multi-Function Timer Pulse Unit 3 (MTU3a)

24.1 Overview

This MCU has an on-chip multi-function timer pulse unit 3 (MTU3a), consisting of eight 16-bit timer channels and one 32-bit timer channel.

Table 24.1 shows the specifications of the MTU and Table 24.2 lists the functions of the MTU. Figure 24.1 and Figure 24.2 show block diagrams of the MTU.

Table 24.1 MTU Specifications

Item	Description
Pulse input/output	28 lines max.
Pulse input	3 lines
Count clock	11 clocks for each channel (14 clocks for MTU0, 12 clocks for MTU2, 10 clocks for MTU5, and four clocks for MTU1 & MTU2 (LWA = 1))
Available operations	[MTU0 to MTU4, MTU6, MTU7, MTU8] • Waveform output on compare match • Input capture function (noise filter setting available) • Counter-clearing operation • Simultaneous writing to multiple timer counters (TCNT) (excluding MTU8) • Simultaneous clearing on compare match or input capture (excluding MTU8) • Simultaneous input and output to registers in synchronization with counter operations (excluding MTU8) • Up to 12-phase PWM output in combination with synchronous operation (excluding MTU8)
	[MTU0, MTU3, MTU4, MTU6, MTU7, MTU8] • Buffer operation specifiable
	[MTU1, MTU2] • Phase counting mode can be specified independently • 32-bit phase counting mode can be specified for interlocked operation of MTU1 and MTU2 (when TMDR3.LWA = 1) • Cascade connection operation available
	[MTU3, MTU4, MTU6, MTU7] • Through interlocked operation of MTU3/4 and MTU6/7, the positive and negative signals in six phases (12 phases in total) can be output in complementary PWM and reset-synchronized PWM operation. • In complementary PWM mode, transfer of values from buffer registers to temporary registers on crests or troughs of the timer-counter values or writing to the buffer registers (MTU4.TGRD and MTU7.TGRD) • Double-buffering selectable in complementary PWM mode
	[MTU3, MTU4] • Through interlocking with MTU0, a mode for driving AC synchronous motors (brushless DC motors) by using complementary PWM output and reset-synchronized PWM output is settable and allows the selection of two types of waveform output (chopping or level)
	[MTU5] • Capable of operation as a dead-time compensation counter
	[MTU0/MTU5, MTU1, MTU2, MTU8] 32-bit phase counting mode specifiable by combining MTU1 and MTU2 and through interlocked operation with MTU0/MTU5 and MTU8
Interrupt skipping function	• In complementary PWM mode, interrupts on crests and troughs of counter values and triggers to start conversion by the A/D converter can be skipped
Interrupt sources	43 sources
Buffer operation	Automatic transfer of register data (transfer from the buffer register to the timer register)
Trigger generation	A/D conversion start triggers can be generated
	A/D conversion start request delaying function enables A/D converter to be started with any desired timing and to be synchronized with PWM output
Low power consumption function	Module stop mode can be set

Table 24.2 MTU Functions (1/2)

Item		MTU0	MTU1	MTU2	MTU1 & MTU2 (LWA = 1)	MTU3	MTU4	MTU5	MTU6	MTU7	MTU8
Count clock		PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB MTCLKC MTCLKD MTIOC1A	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB MTCLKC	MTCLKA MTCLKB MTCLKC MTCLKD	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTIOC1A	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB	PCLKA/1 PCLKA/2 PCLKA/4 PCLKA/8 PCLKA/16 PCLKA/32 PCLKA/64 PCLKA/256 PCLKA/1024 MTCLKA MTCLKB
External clock for phase counting mode		—	MTCLKA MTCLKB	MTCLKA MTCLKB MTCLKC MTCLKD	MTCLKA MTCLKB MTCLKC MTCLKD	—	—	—	—	—	—
General registers (TGR)		TGRA TGRB TGRE	TGRA TGRB	TGRA TGRB	TGRALW TGRBLW	TGRA TGRB	TGRA TGRB	TGRU TGRV TGRW	TGRA TGRB	TGRA TGRB	TGRA TGRB
General registers/ buffer registers		TGRC TGRD TGRF	—	—	—	TGRC TGRD TGRE	TGRC TGRD TGRE TGRF	—	TGRC TGRD TGRE	TGRC TGRD TGRE TGRF	TGRC TGRD
I/O pins		MTIOC0A MTIOC0B MTIOC0C MTIOC0D	MTIOC1A MTIOC1B	MTIOC2A MTIOC2B	MTIOC1A MTIOC1B	MTIOC3A MTIOC3B MTIOC3C MTIOC3D	MTIOC4A MTIOC4B MTIOC4C MTIOC4D	MTIC5U MTIC5V MTIC5W	MTIOC6A MTIOC6B MTIOC6C MTIOC6D	MTIOC7A MTIOC7B MTIOC7C MTIOC7D	MTIOC8A MTIOC8B MTIOC8C MTIOC8D
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGRALW/ TGRBLW input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	✓	✓	✓	—	✓	✓	—	✓	✓	✓
	1 output	✓	✓	✓	—	✓	✓	—	✓	✓	✓
	Toggle output	✓	✓	✓	—	✓	✓	—	✓	✓	✓
Input capture function		✓	✓	✓	✓*1	✓	✓	✓	✓	✓	✓*2
Synchronous operation		✓	✓	✓	—	✓	✓	—	✓	✓	—
PWM mode 1		✓	✓	✓	—	✓	✓	—	✓	✓	—
PWM mode 2		✓	✓	✓	—	—	—	—	—	—	—
Complementary PWM mode		—	—	—	—	✓	✓	—	✓	✓	—
Reset-synchronized PWM mode		—	—	—	—	✓	✓	—	✓	✓	—
AC synchronous motor drive mode		✓	—	—	—	✓	✓	—	—	—	—
Phase counting mode		—	✓	✓	✓	—	—	—	—	—	—
Buffer operation		✓	—	—	—	✓	✓	—	✓	✓	✓
Dead time compensation counter function		—	—	—	—	—	—	✓	—	—	—
DMAC/DTC trigger sources		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGRALW/ TGRBLW input capture	TGR compare match or input capture	TGR compare match or input capture and TCNT overflow/ underflow*3	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture and TCNT overflow/ underflow*3	TGR compare match or input capture
A/D conversion start trigger		TGRA compare match or input capture TGRE compare match	TGRA compare match or input capture	TGRA compare match or input capture	TGRALW input capture	TGRA compare match or input capture	TGRA compare match or input capture, or TCNT underflow (trough) in complementary PWM mode	—	TGRA compare match or input capture	TGRA compare match or input capture, or TCNT underflow (trough) in complementary PWM mode	—

Table 24.2 MTU Functions (2/2)

Item	MTU0	MTU1	MTU2	MTU1 & MTU2 (LWA = 1)	MTU3	MTU4	MTU5	MTU6	MTU7	MTU8
Interrupt sources	Seven sources • Compare match or input capture 0A • Compare match or input capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Compare match 0E • Compare match 0F • Overflow	Four sources • Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	Four sources • Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow	Four sources • Input capture 1A • Input capture 1B • Overflow • Underflow	Five sources • Compare match or input capture 3A • Compare match or input capture 3B • Compare match or input capture 3C • Compare match or input capture 3D • Overflow	Five sources • Compare match or input capture 4A • Compare match or input capture 4B • Compare match or input capture 4C • Compare match or input capture 4D • Overflow or underflow*3	Three sources • Compare match or input capture 5U • Compare match or input capture 5V • Compare match or input capture 5W	Five sources • Compare match or input capture 6A • Compare match or input capture 6B • Compare match or input capture 6C • Compare match or input capture 6D • Overflow	Five sources • Compare match or input capture 7A • Compare match or input capture 7B • Compare match or input capture 7C • Compare match or input capture 7D • Overflow or underflow*3	Five sources • Compare match or input capture 8A • Compare match or input capture 8B • Compare match or input capture 8C • Compare match or input capture 8D • Overflow
Event link function (output)	Seven sources • Compare match 0A • Compare match 0B • Compare match 0C • Compare match 0D • Compare match 0E • Compare match 0F • Overflow	—	—	—	Five sources • Compare match 3A • Compare match 3B • Compare match 3C • Compare match 3D • Overflow	Six sources • Compare match 4A • Compare match 4B • Compare match 4C • Compare match 4D • Overflow • Underflow*3	—	—	—	—
Event link function (input)	• Start counting • Input capture (to be captured in the TGRA) • Restart counting (clear counter)	—	—	—	• Start counting • Input capture (to be captured in the TGRA) • Restart counting (clear counter)	• Start counting • Input capture (to be captured in the TGRA) • Restart counting (clear counter)	—	—	—	—
A/D conversion start request delaying function	—	—	—	—	—	A/D conversion start request at a match between TADCORA and TCNT or A/D conversion start request at a match between TADCORB and TCNT	—	—	A/D conversion start request at a match between TADCORA and TCNT or A/D conversion start request at a match between TADCORB and TCNT	—
Interrupt skipping 1	—	—	—	—	Skips TGRA compare match interrupts	Skips TCIV interrupts	—	Skips TGRA compare match interrupts	Skips TCIV interrupts	—
Interrupt skipping 2	—	—	—	—	—	Skipping in compare count between TADCORA and TCNT, and TADCORB and TCNT	—	—	Skipping in compare count between TADCORA and TCNT, and TADCORB and TCNT	—
Module stop function	MSTPCRA.MSTPA9*4									

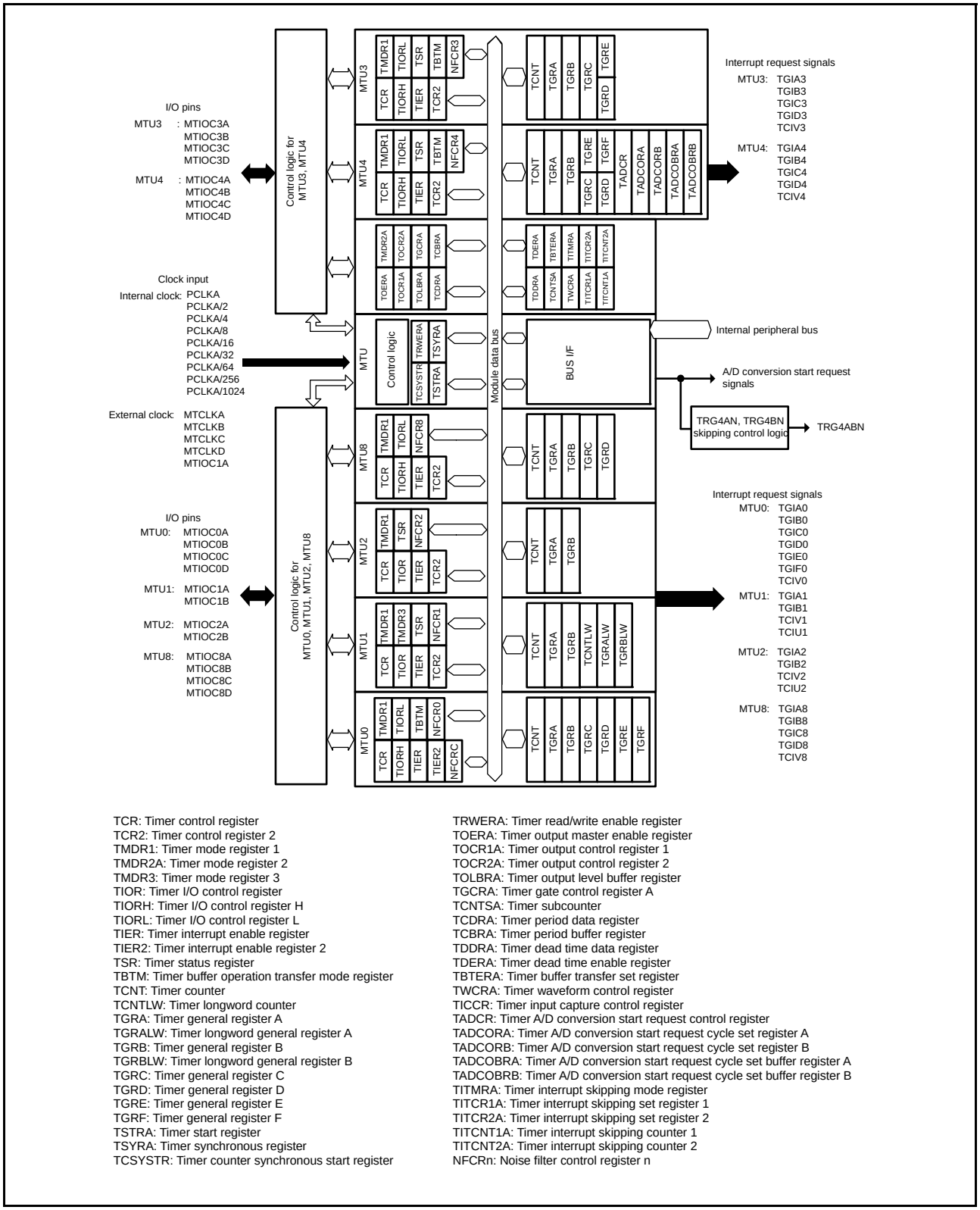
✓: Possible —: Not possible

Note 1. When LWA is 1, the TGRALW capture source can be selected from either of the following: an input from MTIOC1A or MTU0.TGRA compare match/input capture event.
The TGRBLW capture source can be selected from any of the following: an input from MTIOC1B, MTU0.TGRC compare match/input capture event, or MTU8.TGRC compare match event.

Note 2. Capture in MTU8 is supported only in normal mode.

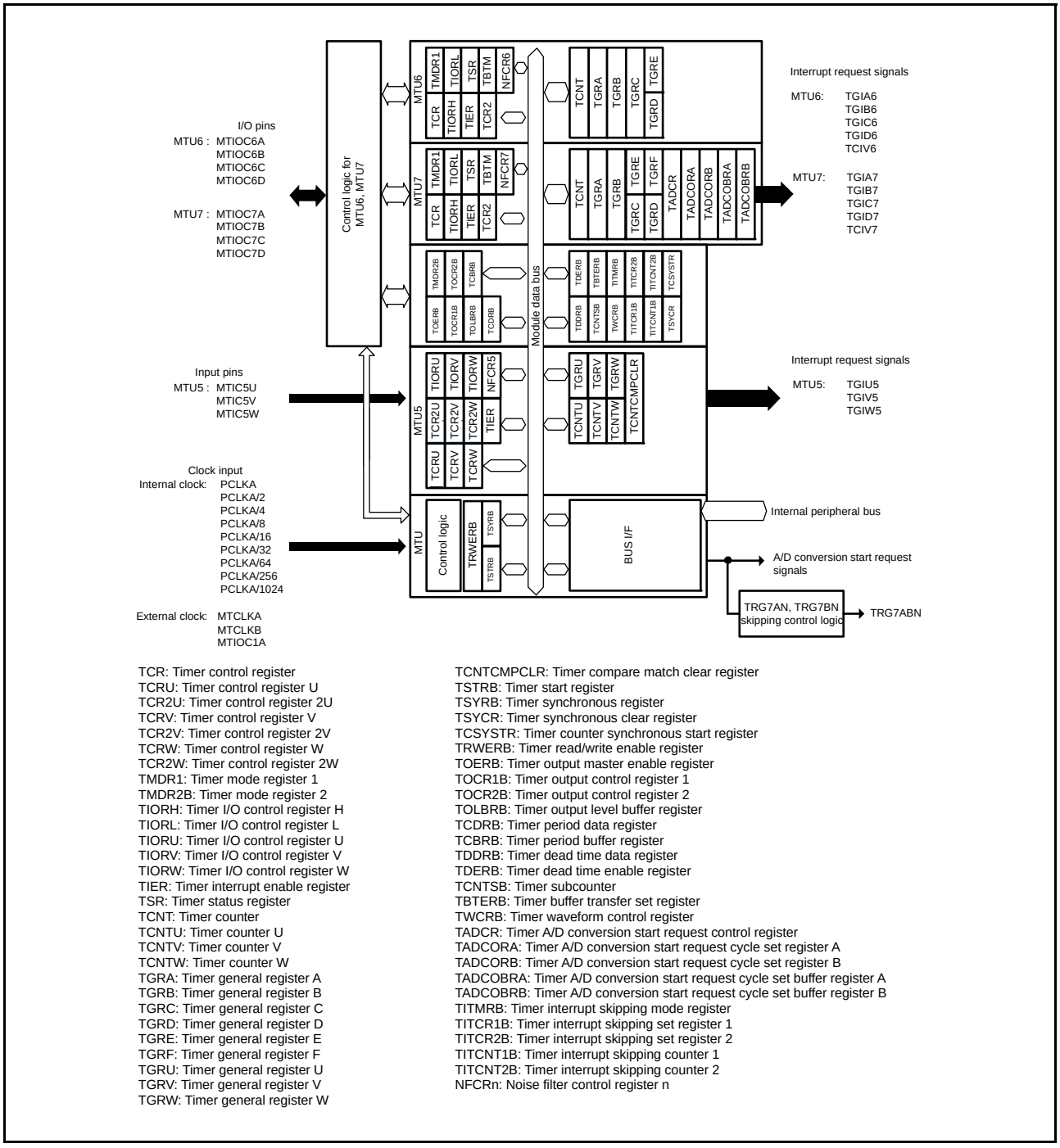
Note 3. Underflow is available only in complementary PWM mode.

Note 4. For details on the module stop function, refer to section 11, Low Power Consumption.



TCR: Timer control register
TCR2: Timer control register 2
TMDR1: Timer mode register 1
TMDR2A: Timer mode register 2
TMDR3: Timer mode register 3
TIOR: Timer I/O control register
TIORH: Timer I/O control register H
TIORL: Timer I/O control register L
TIER: Timer interrupt enable register
TIER2: Timer interrupt enable register 2
TSR: Timer status register
TBTM: Timer buffer operation transfer mode register
TCNT: Timer counter
TCNTLW: Timer longword counter
TGRA: Timer general register A
TGRALW: Timer longword general register A
TGRB: Timer general register B
TGRBLW: Timer longword general register B
TGRC: Timer general register C
TGRD: Timer general register D
TGRE: Timer general register E
TGRF: Timer general register F
TSTRA: Timer start register
TSYRA: Timer synchronous register
TCSYSTR: Timer counter synchronous start register
TRWERA: Timer read/write enable register
TOERA: Timer output master enable register
TOCR1A: Timer output control register 1
TOCR2A: Timer output control register 2
TOLBRA: Timer output level buffer register
TGCRA: Timer gate control register A
TCNTSA: Timer subcounter
TCDRA: Timer period data register
TCBRA: Timer period buffer register
TDDRA: Timer dead time data register
TDERA: Timer dead time enable register
TBTERA: Timer buffer transfer set register
TWCRA: Timer waveform control register
TICCR: Timer input capture control register
TADCR: Timer A/D conversion start request control register
TADCORA: Timer A/D conversion start request cycle set register A
TADCORB: Timer A/D conversion start request cycle set register B
TADCOBRA: Timer A/D conversion start request cycle set buffer register A
TADCOBRB: Timer A/D conversion start request cycle set buffer register B
TITMRA: Timer interrupt skipping mode register
TITCR1A: Timer interrupt skipping set register 1
TITCR2A: Timer interrupt skipping set register 2
TITCNT1A: Timer interrupt skipping counter 1
TITCNT2A: Timer interrupt skipping counter 2
NFCRn: Noise filter control register n

Figure 24.1 Block Diagram of MTU (MTU0 to MTU4, MTU8)



TCR: Timer control register
TCRU: Timer control register U
TCR2U: Timer control register 2U
TCRV: Timer control register V
TCR2V: Timer control register 2V
TCRW: Timer control register W
TCR2W: Timer control register 2W
TMDR1: Timer mode register 1
TMDR2B: Timer mode register 2
TIORH: Timer I/O control register H
TIORL: Timer I/O control register L
TIORU: Timer I/O control register U
TIORV: Timer I/O control register V
TIORW: Timer I/O control register W
TIER: Timer interrupt enable register
TSR: Timer status register
TCNT: Timer counter
TCNTU: Timer counter U
TCNTV: Timer counter V
TCNTW: Timer counter W
TGRA: Timer general register A
TGRB: Timer general register B
TGRC: Timer general register C
TGRD: Timer general register D
TGRE: Timer general register E
TGRF: Timer general register F
TGRU: Timer general register U
TGRV: Timer general register V
TGRW: Timer general register W

TCNTCMPCLR: Timer compare match clear register
TSTRB: Timer start register
TSYRB: Timer synchronous register
TSYCR: Timer synchronous clear register
TCSYSTR: Timer counter synchronous start register
TRWERB: Timer read/write enable register
TOERB: Timer output master enable register
TOCR1B: Timer output control register 1
TOCR2B: Timer output control register 2
TOLBRB: Timer output level buffer register
TCDRB: Timer period data register
TCBRB: Timer period buffer register
TDDRB: Timer dead time data register
TDERB: Timer dead time enable register
TCNTSB: Timer subcounter
TBTERB: Timer buffer transfer set register
TWCRB: Timer waveform control register
TADCR: Timer A/D conversion start request control register
TADCORA: Timer A/D conversion start request cycle set register A
TADCORB: Timer A/D conversion start request cycle set register B
TADCOBRA: Timer A/D conversion start request cycle set buffer register A
TADCOBRB: Timer A/D conversion start request cycle set buffer register B
TITMRB: Timer interrupt skipping mode register
TITCR1B: Timer interrupt skipping set register 1
TITCR2B: Timer interrupt skipping set register 2
TITCNT1B: Timer interrupt skipping counter 1
TITCNT2B: Timer interrupt skipping counter 2
NFCRn: Noise filter control register n

Figure 24.2 Block Diagram of MTU (MTU5 to MTU7)

Table 24.3 shows the configuration of pins for the MTU.

Table 24.3 Pin Configuration of the MTU

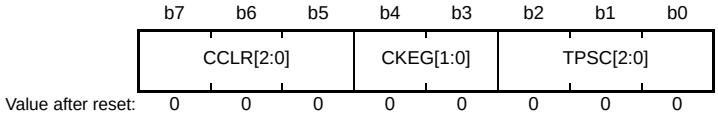
Channel	Pin Name	I/O	Function
MTU	MTCLKA	Input	External clock A input pin (MTU1 and MTU2 phase counting mode A phase input)
	MTCLKB	Input	External clock B input pin (MTU1 and MTU2 phase counting mode B phase input)
	MTCLKC	Input	External clock C input pin (MTU2 phase counting mode A phase input)
	MTCLKD	Input	External clock D input pin (MTU2 phase counting mode B phase input)
MTU0	MTIOC0A	I/O	MTU0 TGRA input capture input/output compare output/PWM output pin
	MTIOC0B	I/O	MTU0 TGRB input capture input/output compare output/PWM output pin
	MTIOC0C	I/O	MTU0 TGRC input capture input/output compare output/PWM output pin
	MTIOC0D	I/O	MTU0 TGRD input capture input/output compare output/PWM output pin
MTU1	MTIOC1A	I/O	MTU1 TGRA input capture input/output compare output/PWM output pin
	MTIOC1B	I/O	MTU1 TGRB input capture input/output compare output/PWM output pin
MTU2	MTIOC2A	I/O	MTU2 TGRA input capture input/output compare output/PWM output pin
	MTIOC2B	I/O	MTU2 TGRB input capture input/output compare output/PWM output pin
MTU3	MTIOC3A	I/O	MTU3 TGRA input capture input/output compare output/PWM output pin
	MTIOC3B	I/O	MTU3 TGRB input capture input/output compare output/PWM output pin
	MTIOC3C	I/O	MTU3 TGRC input capture input/output compare output/PWM output pin
	MTIOC3D	I/O	MTU3 TGRD input capture input/output compare output/PWM output pin
MTU4	MTIOC4A	I/O	MTU4 TGRA input capture input/output compare output/PWM output pin
	MTIOC4B	I/O	MTU4 TGRB input capture input/output compare output/PWM output pin
	MTIOC4C	I/O	MTU4 TGRC input capture input/output compare output/PWM output pin
	MTIOC4D	I/O	MTU4 TGRD input capture input/output compare output/PWM output pin
MTU5	MTIC5U	Input	MTU5 TGRU input capture input/external pulse input pin
	MTIC5V	Input	MTU5 TGRV input capture input/external pulse input pin
	MTIC5W	Input	MTU5 TGRW input capture input/external pulse input pin
MTU6	MTIOC6A	I/O	MTU6 TGRA input capture input/output compare output/PWM output pin
	MTIOC6B	I/O	MTU6 TGRB input capture input/output compare output/PWM output pin
	MTIOC6C	I/O	MTU6 TGRC input capture input/output compare output/PWM output pin
	MTIOC6D	I/O	MTU6 TGRD input capture input/output compare output/PWM output pin
MTU7	MTIOC7A	I/O	MTU7 TGRA input capture input/output compare output/PWM output pin
	MTIOC7B	I/O	MTU7 TGRB input capture input/output compare output/PWM output pin
	MTIOC7C	I/O	MTU7 TGRC input capture input/output compare output/PWM output pin
	MTIOC7D	I/O	MTU7 TGRD input capture input/output compare output/PWM output pin
MTU8	MTIOC8A	I/O	MTU8.TGRA input capture input/output compare output pin
	MTIOC8B	I/O	MTU8.TGRB input capture input/output compare output pin
	MTIOC8C	I/O	MTU8.TGRC input capture input/output compare output pin
	MTIOC8D	I/O	MTU8.TGRD input capture input/output compare output pin

24.2 Register Descriptions

24.2.1 Timer Control Register (TCR)

- MTU0.TCR, MTU1.TCR, MTU2.TCR, MTU3.TCR, MTU4.TCR, MTU6.TCR, MTU7.TCR, MTU8.TCR

Address(es): MTU0.TCR 000C 1300h, MTU1.TCR 000C 1380h, MTU2.TCR 000C 1400h, MTU3.TCR 000C 1200h, MTU4.TCR 000C 1201h, MTU6.TCR 000C 1A00h, MTU7.TCR 000C 1A01h, MTU8.TCR 000C 1600h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC[2:0]	Time Prescaler Select	Refer to Table 24.6 to Table 24.9.	R/W
b4, b3	CKEG[1:0]	Clock Edge Select	b4 b3 0 0: Count at rising edge 0 1: Count at falling edge 1 x: Count at both edges	R/W
b7 to b5	CCLR[2:0]	Counter Clear Source Select	Refer to Table 24.4 and Table 24.5.	R/W

x: Don’t care

The TCR register controls the TCNT operation for each channel in combination with the TCR2 register. The MTU has a total of 11 TCR registers, one each for MTU0 to MTU4, MTU6, MTU7, and MTU8 and three (TCRU, TCRV, and TCRW) for MTU5. TCR values should be specified only while TCNT operation is stopped.

TPSC[2:0] Bits (Time Prescaler Select)

These bits select the TCNT count clock source. The count clock source can be selected independently for each channel. Refer to Table 24.6 to Table 24.9 for details.

CKEG[1:0] Bits (Clock Edge Select)

These bits select the clock edge, including the MTIOC1A pin. When the internal clock is counted at both edges, the count clock period is halved (e.g. PCLKA/4 at both edges = PCLKA/2 at rising edge). If phase counting mode is used on MTU1 and MTU2, the setting of these bits is ignored and the phase counting mode setting has priority. Internal clock edge selection is valid when the count clock source is PCLKA/2 or slower. When PCLKA/1 or the overflow/underflow in another channel is selected for the count clock source, a value can be written to these bits but counter operation compiles with the initial value.

CCLR[2:0] Bits (Counter Clear Source Select)

These bits select the TCNT counter clearing source. Refer to Table 24.4 and Table 24.5 for details.

Table 24.4 CCLR[2:0] (MTU0, MTU3, MTU4, MTU6, MTU7, MTU8)

Channel	Bit 7 CCLR[2]	Bit 6 CCLR[1]	Bit 5 CCLR[0]	Description
MTU0 MTU3 MTU4 MTU6 MTU7 MTU8	0	0	0	TCNT clearing disabled
	0	0	1	TCNT cleared by TGRA compare match/input capture
	0	1	0	TCNT cleared by TGRB compare match/input capture
	0	1	1	TCNT cleared by counter clearing in another channel performing synchronous clearing/ synchronous operation*1
	1	0	0	TCNT clearing disabled
	1	0	1	TCNT cleared by TGRC compare match/input capture*2
	1	1	0	TCNT cleared by TGRD compare match/input capture*2
	1	1	1	TCNT cleared by counter clearing in another channel performing synchronous clearing/ synchronous operation*1

Note 1. Synchronous operation is selected by setting the TSYRA.SYNC or TSYRB.SYNC bit to 1 except for MTU8.
Note 2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority and compare match/input capture does not occur.

Table 24.5 CCLR[2:0] (MTU1 and MTU2)

Channel	Bit 7 Reserved*2	Bit 6 CCLR[1]	Bit 5 CCLR[0]	Description
MTU1 MTU2	0	0	0	TCNT clearing disabled
	0	0	1	TCNT cleared by TGRA compare match/input capture (when LWA = 0) TCNTLW cleared by TGRALW input capture (when LWA = 1)
	0	1	0	TCNT cleared by TGRB compare match/input capture (when LWA = 0) TCNTLW cleared by TGRBLW input capture (when LWA = 1)
	0	1	1	TCNT cleared by counter clearing in another channel performing synchronous clearing/synchronous operation*1

Note 1. Synchronous operation is selected by setting the TSYRA.SYNC and TSYRB.SYNC bits to 1.
Note 2. Bit 7 is reserved in MTU1 and MTU2. It is read as 0. The write value is ignored.

- MTU5.TCRU, MTU5.TCRV, MTU5.TCRW

Address(es): MTU5.TCRU 000C 1C84h, MTU5.TCRV 000C 1C94h, MTU5.TCRW 000C 1CA4h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TPSC[1:0]	Time Prescaler Select	Refer to Table 24.10.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TPSC[1:0] Bits (Time Prescaler Select)

These bits select the TCNT count clock source. Refer to Table 24.10 for details.

24.2.2 Timer Control Register 2 (TCR2)

- MTU0.TCR2, MTU3.TCR2, MTU4.TCR2, MTU6.TCR2, MTU7.TCR2, MTU8.TCR2

Address(es): MTU0.TCR2 000C 1328h, MTU3.TCR2 000C 124Ch, MTU4.TCR2 000C 124Dh, MTU6.TCR2 000C 1A4Ch, MTU7.TCR2 000C 1A4Dh, MTU8.TCR2 000C 1606h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TPSC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

- MTU1.TCR2, MTU2.TCR2

Address(es): MTU1.TCR2 000C 1394h, MTU2.TCR2 000C 140Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	PCB[1:0]		TPSC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC2[2:0]	Time Prescaler Select	Refer to Table 24.6 to Table 24.9.	R/W
b4, b3	PCB[1:0]	Phase Counting Mode Function Expansion Control	Functional Expansion Control for Phase Counting Modes 2, 3, and 5	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TCR2 register controls the TCNT operation for each channel in combination with the TCR register. The MTU has a total of 11 TCR2 registers, one each for MTU0 to MTU4, MTU6, MTU7, and MTU8 and three (TCR2U, TCR2V, and TCR2W) for MTU5. TCR2 values should be specified only while TCNT operation is stopped.

TPSC2[2:0] Bits (Time Prescaler Select)

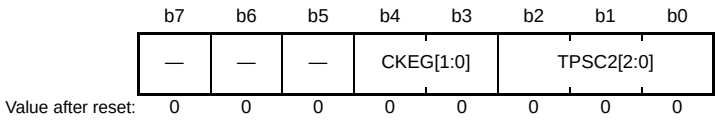
These bits select the TCNT count clock source. The count clock source can be selected independently for each channel. Refer to Table 24.6 to Table 24.9 for details.

PCB[1:0] Bits (Phase Counting Mode Function Expansion Control)

These bits control extended functions for phase counting mode 2, 3, and 5 in MTU1 and MTU2. Refer to section 24.3.6, Phase Counting Mode.

- MTU5.TCR2U, MTU5.TCR2V, MTU5.TCR2W

Address(es): MTU5.TCR2U 000C 1C85h, MTU5.TCR2V 000C 1C95h, MTU5.TCR2W 000C 1CA5h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC2[2:0]	Time Prescaler Select	Refer to Table 24.10.	R/W
b4, b3	CKEG[1:0]	Clock Edge Select	b4 b3 0 0: Counts at the rising edge. 0 1: Counts at the falling edge. 1 x: Counts at both edges.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

x: Don't care

TPSC2[2:0] Bits (Time Prescaler Select)

These bits select the TCNT count clock source. Refer to Table 24.10 for details.

CKEG[1:0] Bits (Clock Edge Select)

These bits select the edge of the count clock signal input from the MTIOC1A pin.

Table 24.6 TPSC[2:0], TPSC2[2:0] (MTU0)

	TCR2 register			TCR register			
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	Description
MTU0	0	0	0	0	0	0	Internal clock: counts on PCLKA/1
	0	0	0	0	0	1	Internal clock: counts on PCLKA/4
	0	0	0	0	1	0	Internal clock: counts on PCLKA/16
	0	0	0	0	1	1	Internal clock: counts on PCLKA/64
	0	0	0	1	0	0	External clock: counts on MTCLKA pin input
	0	0	0	1	0	1	External clock: counts on MTCLKB pin input
	0	0	0	1	1	0	External clock: counts on MTCLKC pin input
	0	0	0	1	1	1	External clock: counts on MTCLKD pin input
	0	0	1	x	x	x	Internal clock: counts on PCLKA/2
	0	1	0	x	x	x	Internal clock: counts on PCLKA/8
	0	1	1	x	x	x	Internal clock: counts on PCLKA/32
	1	0	0	x	x	x	Internal clock: counts on PCLKA/256
	1	0	1	x	x	x	Internal clock: counts on PCLKA/1024
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	External clock: counts on MTIOC1A pin input

x: Don't care

Table 24.7 TPSC[2:0], TPSC2[2:0] (MTU1)

	TCR2 register			TCR register			
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	Description
MTU1	0	0	0	0	0	0	Internal clock: counts on PCLKA/1
	0	0	0	0	0	1	Internal clock: counts on PCLKA/4
	0	0	0	0	1	0	Internal clock: counts on PCLKA/16
	0	0	0	0	1	1	Internal clock: counts on PCLKA/64
	0	0	0	1	0	0	External clock: counts on MTCLKA pin input
	0	0	0	1	0	1	External clock: counts on MTCLKB pin input
	0	0	0	1	1	0	Internal clock: counts on PCLKA/256
	0	0	0	1	1	1	Overflow/underflow of MTU2.TCNT
	0	0	1	x	x	x	Internal clock: counts on PCLKA/2
	0	1	0	x	x	x	Internal clock: counts on PCLKA/8
	0	1	1	x	x	x	Internal clock: counts on PCLKA/32
	1	0	0	x	x	x	Internal clock: counts on PCLKA/1024
	1	0	1	x	x	x	Setting prohibited
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	Setting prohibited

x: Don't care

Note: This setting has no effect when MTU1 is in phase counting mode.

Table 24.8 TPSC[2:0], TPSC2[2:0] (MTU2)

	TCR2 register			TCR register			
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	Description
MTU2	0	0	0	0	0	0	Internal clock: counts on PCLKA/1
	0	0	0	0	0	1	Internal clock: counts on PCLKA/4
	0	0	0	0	1	0	Internal clock: counts on PCLKA/16
	0	0	0	0	1	1	Internal clock: counts on PCLKA/64
	0	0	0	1	0	0	External clock: counts on MTCLKA pin input
	0	0	0	1	0	1	External clock: counts on MTCLKB pin input
	0	0	0	1	1	0	External clock: counts on MTCLKC pin input
	0	0	0	1	1	1	Internal clock: counts on PCLKA/1024
	0	0	1	x	x	x	Internal clock: counts on PCLKA/2
	0	1	0	x	x	x	Internal clock: counts on PCLKA/8
	0	1	1	x	x	x	Internal clock: counts on PCLKA/32
	1	0	0	x	x	x	Internal clock: counts on PCLKA/256
	1	0	1	x	x	x	Setting prohibited
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	Setting prohibited

x: Don't care

Note: This setting has no effect when the MTU2 is in phase counting mode.

Table 24.9 TPSC[2:0], TPSC2[2:0] (MTU3, MTU4, MTU6, MTU7, MTU8)

	TCR2 register			TCR register			
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	Description
MTU3 MTU4 MTU6 MTU7 MTU8	0	0	0	0	0	0	Internal clock: counts on PCLKA/1
	0	0	0	0	0	1	Internal clock: counts on PCLKA/4
	0	0	0	0	1	0	Internal clock: counts on PCLKA/16
	0	0	0	0	1	1	Internal clock: counts on PCLKA/64
	0	0	0	1	0	0	Internal clock: counts on PCLKA/256
	0	0	0	1	0	1	Internal clock: counts on PCLKA/1024
	0	0	0	1	1	0	External clock: counts on MTCLKA pin input
	0	0	0	1	1	1	External clock: counts on MTCLKB pin input
	0	0	1	x	x	x	Internal clock: counts on PCLKA/2
	0	1	0	x	x	x	Internal clock: counts on PCLKA/8
	0	1	1	x	x	x	Internal clock: counts on PCLKA/32
	1	0	0	x	x	x	Setting prohibited
	1	0	1	x	x	x	Setting prohibited
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	Setting prohibited

x: Don't care

Table 24.10 TPSC[1:0], TPSC2[2:0] (MTU5)

	TCR2 register			TCR register		
	Bit 2	Bit 1	Bit 0	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[1]	TPSC[0]	Description
MTU5	0	0	0	0	0	Internal clock: counts on PCLKA/1
	0	0	0	0	1	Internal clock: counts on PCLKA/4
	0	0	0	1	0	Internal clock: counts on PCLKA/16
	0	0	0	1	1	Internal clock: counts on PCLKA/64
	0	0	1	x	x	Internal clock: counts on PCLKA/2
	0	1	0	x	x	Internal clock: counts on PCLKA/8
	0	1	1	x	x	Internal clock: counts on PCLKA/32
	1	0	0	x	x	Internal clock: counts on PCLKA/256
	1	0	1	x	x	Internal clock: counts on PCLKA/1024
	1	1	0	x	x	Setting prohibited
	1	1	1	x	x	External clock: counts on MTIOC1A pin input

x: Don't care

24.2.3 Timer Mode Register 1 (TMDR1)

- MTU0.TMDR1

Address(es): MTU0.TMDR1 000C 1301h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	BFE	BFB	BFA	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

- MTU1.TMDR1, MTU2.TMDR1

Address(es): MTU1.TMDR1 000C 1381h, MTU2.TMDR1 000C 1401h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

- MTU3.TMDR1, MTU4.TMDR1, MTU6.TMDR1, MTU7.TMDR1, MTU8.TMDR1

Address(es): MTU3.TMDR1 000C 1202h, MTU4.TMDR1 000C 1203h, MTU6.TMDR1 000C 1A02h, MTU7.TMDR1 000C 1A03h, MTU8.TMDR1 000C 1601h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	BFB	BFA	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MD[3:0]	Mode Select	These bits specify the timer operating mode. Refer to Table 24.11 for details.	R/W
b4	BFA	Buffer Operation A	0: TGRA and TGRC operate normally 1: TGRA and TGRC used together for buffer operation	R/W
b5	BFB	Buffer Operation B	0: TGRB and TGRD operate normally 1: TGRB and TGRD used together for buffer operation	R/W
b6	BFE	Buffer Operation E	0: MTU0.TGRE and MTU0.TGRF operate normally 1: MTU0.TGRE and MTU0.TGRF used together for buffer operation	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The TMDR1 register specifies the operating mode of each channel. The MTU has a total of eight TMDR1 registers, one each for MTU0 to MTU4, MTU6, MTU7, and MTU8. TMDR1 register values should be specified only while TCNT operation is stopped.

Table 24.11 Operating Mode Setting by MD[3:0] Bits (MTU0 to MTU4 and MTU6 to MTU8)

Bit 3	Bit 2	Bit 1	Bit 0		MTU0	MTU1	MTU2	MTU1 & MTU2 (LWA = 1)	MTU3	MTU4	MTU6	MTU7	MTU8
MD[3]	MD[2]	MD[1]	MD[0]	Description									
0	0	0	0	Normal mode	✓	✓	✓		✓	✓	✓	✓	✓
0	0	0	1	Setting prohibited									
0	0	1	0	PWM mode 1	✓	✓	✓		✓	✓	✓	✓	
0	0	1	1	PWM mode 2	✓	✓	✓						
0	1	0	0	Phase counting mode 1		✓	✓	✓					
0	1	0	1	Phase counting mode 2		✓	✓	✓					
0	1	1	0	Phase counting mode 3		✓	✓	✓					
0	1	1	1	Phase counting mode 4		✓	✓	✓					
1	0	0	0	Reset-synchronized PWM mode*1					✓		✓		
1	0	0	1	Phase counting mode 5		✓	✓	✓					
1	0	1	x	Setting prohibited									
1	1	0	0	Setting prohibited									
1	1	0	1	Complementary PWM mode 1 (transfer at crest)*1					✓		✓		
1	1	1	0	Complementary PWM mode 2 (transfer at trough)*1					✓		✓		
1	1	1	1	Complementary PWM mode 3 (transfer at crest and trough)*1					✓		✓		

x: Don't care

Note: Only set the corresponding operating mode listed above for each channel.

Note 1. Reset-synchronized PWM mode and complementary PWM mode can only be set for MTU3 and MTU6. When MTU3 or MTU6 is set to reset-synchronized PWM mode or complementary PWM mode, the MTU4 or MTU7 settings become ineffective and automatically conform to the MTU3 or MTU6 setting, respectively. MTU4 and MTU7 should be set to the initial values (normal mode).

BFA Bit (Buffer Operation A)

This bit specifies whether to operate TGRA in the normal way or to use TGRA and TGRC together for buffer operation. When TGRC is used as a buffer register, TGRC input capture/output compare does not take place in modes other than complementary PWM mode, but compare match with TGRC occurs in complementary PWM mode.

If a compare match occurs on MTU4 in the Tb interval in complementary PWM mode, the TGIEC bit in timer interrupt enable register (MTU4.TIER) should be set to 0.

In reset-synchronized PWM mode or complementary PWM mode, buffer operation in MTU3 and MTU4 (or MTU6 and MTU7) depends on the settings in the BFA bit of MTU3.TMDR1 (MTU6.TMDR1). The BFA bit of MTU4.TMDR1 (MTU7.TMDR1) should be set to 0.

In MTU1 and MTU2, which have no TGRC, this bit is reserved. It is read as 0. The write value should be 0.

Refer to Figure 24.50 for an illustration of the Tb interval in complementary PWM mode.

BFB Bit (Buffer Operation B)

This bit specifies whether to operate TGRB in the normal way or to use TGRB and TGRD together for buffer operation. When TGRD is used as a buffer register, TGRD input capture/output compare does not take place in modes other than complementary PWM mode, but compare match with TGRD occurs in complementary PWM mode.

If a compare match occurs on MTU4 in the Tb interval in complementary PWM mode, the TGIED bit in timer interrupt enable register (MTU4.TIER) should be set to 0.

In reset-synchronized PWM mode or complementary PWM mode, buffer operation in MTU3 and MTU4 (or MTU6 and MTU7) depends on the settings in the BFB bit of MTU3.TMDR1 (MTU6.TMDR1). The BFB bit of MTU4.TMDR1

(MTU7.TMDR1) should be set to 0.
In MTU1 and MTU2, which have no TGRD, this bit is reserved. It is read as 0. The write value should be 0.
Refer to Figure 24.50 for an illustration of the Tb interval in complementary PWM mode.

BFE Bit (Buffer Operation E)

This bit specifies whether to operate MTU0.TGRE and MTU0.TGRF in the normal way or to use them together for buffer operation. Compare match with TGRF occurs even when TGRF is used as a buffer register.
In MTU0 to MTU4, MTU6, MTU7, and MTU8, this bit is reserved. It is read as 0. The write value should be 0.

24.2.4 Timer Mode Register 2m (TMDR2m) (m = A, B)

Address(es): MTU.TMDR2A 000C 1270h, MTU.TMDR2B 000C 1A70h

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

Bit	Symbol	Bit Name	Description	R/W
b0	DRS	Double Buffer Select	0: Double buffer function is disabled 1: Double buffer function is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TMDR2A and TMDR2B specify the double buffer function in complementary PWM mode 3 (transfer at the crest and trough of the counter value). The MTU has two TMDR2 registers, and one each for MTU3 (TMDR2A) and MTU6 (TMDR2B). TMDR2A and TMDR2B values should be specified only while TCNT operation is stopped.

DRS Bit (Double Buffer Select)

This bit enables or disables the double buffer function in complementary PWM mode.

24.2.5 Timer Mode Register 3 (TMDR3)

Address(es): MTU1.TMDR3 000C 1391h

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

Bit	Symbol	Bit Name	Description	R/W
b0	LWA	MTU1/MTU2 Combination Longword Access Control	0: 16-bit access is enabled. 1: 32-bit access is enabled.	R/W
b1	PHCKSEL	External Input Phase Clock Select	0: MTCLKA and MTCLKB are selected for the external phase clock. 1: MTCLKC and MTCLKD are selected for the external phase clock.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TMDR3 register controls longword access to a 32-bit register or counter in a combination of MTU1 and MTU2. There is only one TMDR3 register in MTU1. The counter (TCNTLW), general register A (TGRALW), and general register B (TGRBLW) of MTU1 and MTU2 are accessed in the combinations listed in Table 24.12.

LWA Bit (MTU1/MTU2 Combination Longword Access Control)

This bit selects a 32-bit access in a combination of MTU1 and MTU2.

When LWA is set to 0, the MTU1 and MTU2 independently operate as a 16-bit timer. therefore registers TCNTLW, TGRALW, and TGRBLW cannot be accessed.

When LWA is set to 1, MTU1 and MTU2 operate as a 32-bit cascaded timer and the timer is controlled by registers MTU1.TCR, MTU1.TCR2, MTU1.TIOR, and MTU1.TMDR1. The settings of registers MTU2.TCR, MTU2.TCR2, MTU2.TIOR, and MTU2.TMDR1 are disabled and the 16-bit registers (TCNT, TGRA, and TGRB) in MTU1 and MTU2 cannot be accessed. Furthermore, MTU2 input capture and compare match are also disabled, which in turn disables any linked operation with the ELC.

The cascaded connection of MTU1 and MTU2 with the LWA bit set to 1 can only be used in phase counting mode, but not in normal mode, PWM1 mode, or PWM2 mode. Select phase counting mode when setting the LWA bit to 1.

Initialize the registers TCNT, TGRA, and TGRB in MTU1 and MTU2 in advance before setting the LWA bit to 1.

PHCKSEL Bit (External Input Phase Clock Select)

When the MTU1 and MTU2 registers are combined for 32-bit phase counting mode or MTU2 phase counting mode, this bit selects either the A- or B-phase signal from the external clock. Refer to Table 24.65, Clock Input Pins in Phase Counting Mode for details.

Table 24.12 Setting and Combination of the TMDR3 Register

	TMDR3.LWA = 0		TMDR3.LWA = 1	
Register	**Symbol**	**Access mode**	**Symbol**	**Access mode**
Counter in MTU1*1	MTU1.TCNT	Word	MTU1.TCNTLW	Longword
Counter in MTU2	MTU2.TCNT	Word		
General register A in MTU1	MTU1.TGRA	Word	MTU1.TGRALW	Longword
General register A in MTU2	MTU2.TGRA	Word		
General register B in MTU1	MTU1.TGRB	Word	MTU1.TGRBLW	Longword
General register B in MTU2	MTU2.TGRB	Word		

Note 1. When the LWA bit is set to 1, setting the count clock for MTU1 as MTU2.TCNT overflow/underflow is not required.

24.2.6 Timer I/O Control Register (TIOR)

- MTU0.TIORH, MTU1.TIOR, MTU2.TIOR, MTU3.TIORH, MTU4.TIORH, MTU6.TIORH, MTU7.TIORH, MTU8.TIORH

Address(es): MTU0.TIORH 000C 1302h, MTU1.TIOR 000C 1382h, MTU2.TIOR 000C 1402h, MTU3.TIORH 000C 1204h, MTU4.TIORH 000C 1206h, MTU6.TIORH 000C 1A04h, MTU7.TIORH 000C 1A06h, MTU8.TIORH 000C 1602h

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOA[3:0]	I/O Control A*1	Refer to the following tables. MTU0.TIORH: Table 24.27 MTU1.TIOR: Table 24.29 MTU2.TIOR: Table 24.30 MTU3.TIORH: Table 24.31 MTU4.TIORH: Table 24.33 MTU6.TIORH: Table 24.35 MTU7.TIORH: Table 24.37 MTU8.TIORH: Table 24.39	R/W
b7 to b4	IOB[3:0]	I/O Control B*1	Refer to the following tables. MTU0.TIORH: Table 24.13 MTU1.TIOR: Table 24.15 MTU2.TIOR: Table 24.16 MTU3.TIORH: Table 24.17 MTU4.TIORH: Table 24.19 MTU6.TIORH: Table 24.21 MTU7.TIORH: Table 24.23 MTU8.TIORH: Table 24.25	R/W

Note 1. When the value of IOm[3:0] (m = A, B) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

- MTU0.TIORL, MTU3.TIORL, MTU4.TIORL, MTU6.TIORL, MTU7.TIORL, MTU8.TIORL

Address(es): MTU0.TIORL 000C 1303h, MTU3.TIORL 000C 1205h, MTU4.TIORL 000C 1207h, MTU6.TIORL 000C 1A05h, MTU7.TIORL 000C 1A07h, MTU8.TIORL 000C 1603h

	b7	b6	b5	b4	b3	b2	b1	b0
	IOD[3:0]				IOC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOC[3:0]	I/O Control C*1	Refer to the following tables. MTU0.TIORL: Table 24.28 MTU3.TIORL: Table 24.32 MTU4.TIORL: Table 24.34 MTU6.TIORL: Table 24.36 MTU7.TIORL: Table 24.38 MTU8.TIORL: Table 24.40	R/W
b7 to b4	IOD[3:0]	I/O Control D*1	Refer to the following tables. MTU0.TIORL: Table 24.14 MTU3.TIORL: Table 24.18 MTU4.TIORL: Table 24.20 MTU6.TIORL: Table 24.22 MTU7.TIORL: Table 24.24 MTU8.TIORL: Table 24.26	R/W

Note 1. When the value of IOm[3:0] (m = C, D) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

- MTU5.TIORU, MTU5.TIORV, MTU5.TIORW

Address(es): MTU5.TIORU 000C 1C86h, MTU5.TIORV 000C 1C96h, MTU5.TIORW 000C 1CA6h

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	IOC[4:0]	I/O Control C	Refer to the following table. MTU5.TIORU, MTU5.TIORV, MTU5.TIORW: Table 24.41	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TIOR register controls the TGR register. The MTU has a total of 17 TIOR registers, two each for MTU0, MTU3, MTU4, MTU6, MTU7, and MTU8, one each for MTU1 and MTU2, and three (MTU5.TIORU/TIORV/TIORW) for MTU5. The TIOR register should be set when the TMDR register setting is normal mode, PWM mode, or phase counting mode.

Note that TIOR is affected by the TMDR1 setting.

The initial output specified by TIOR is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). Note also that, in PWM mode 2, the output at the point at which the counter becomes 0000h is specified.

When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

Table 24.13 TIORH (MTU0)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB Register Function	MTIOC0B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1).*1

x: Don't care

Note 1. When PCLKA/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLKA/1 as the count clock for MTU1.

Table 24.14 TIORL (MTU0)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	TGRD Register Function	MTIOC0D Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register*1	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1).*2

x: Don't care

Note 1. When the MTU0.TMDR1.BFB bit is set to 1 and MTU0.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note 2. When PCLKA/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLKA/1 as the count clock for MTU1.

Table 24.15 TIOR (MTU1)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB/TGRBLW Register Function	MTIOC1B Pin Function
0	0	0	0	Output compare register (only available when LWA = 0)	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	0	0		Input capture at occurrence of compare match or input capture in the MTU0.TGRC register
1	1	1	x		Input capture at occurrence of compare match in the MTU8.TGRC register

x: Don't care

Table 24.16 TIOR (MTU2)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB Register Function	MTIOC2B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 24.17 TIORH (MTU3)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB Register Function	MTIOC3B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 24.18 TIORL (MTU3)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	TGRD Register Function	MTIOC3D Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU3.TMDR1.BFB bit is set to 1 and MTU3.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 24.19 TIORH (MTU4)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB Register Function	MTIOC4B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 24.20 TIORL (MTU4)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	TGRD Register Function	MTIOC4D Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU4.TMDR1.BFB bit is set to 1 and MTU4.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 24.21 TIORH (MTU6)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB Register Function	MTIOC6B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 24.22 TIORL (MTU6)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	TGRD Register Function	MTIOC6D Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU6.TMDR1.BFB bit is set to 1 and MTU6.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 24.23 TIORH (MTU7)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB Register Function	MTIOC7B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 24.24 TIORL (MTU7)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	TGRD Register Function	MTIOC7D Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU7.TMDR1.BFB bit is set to 1 and MTU7.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 24.25 TIORH (MTU8)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB Register Function	MTIOC8B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1).

x: Don't care

Table 24.26 TIORL (MTU8)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	TGRD Register Function	MTIOC8D Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU8.TMDR1.BFB bit is set to 1 and the MTU8.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 24.27 TIORH (MTU0)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA Register Function	MTIOC0A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	0	0		Capture input source is the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1).*1
1	1	1	x		Input capture on generation of compare match with MTU8.TGRC

x: Don't care

Note 1. When PCLKA/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLKA/1 as the count clock for MTU1.

Table 24.28 TIORL (MTU0)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRC Register Function	MTIOC0C Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register*1	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1).*2

x: Don't care

Note 1. When the MTU0.TMDR1.BFB bit is set to 1 and MTU0.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note 2. When PCLKA/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLKA/1 as the count clock for MTU1.

Table 24.29 TIOR (MTU1)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA/TGRALW Register Function	MTIOC1A Pin Function
0	0	0	0	Output compare register (only available when LWA = 0)	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Input capture at generation of MTU0.TGRA compare match/input capture.

x: Don't care

Table 24.30 TIOR (MTU2)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA Register Function	MTIOC2A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 24.31 TIORH (MTU3)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA Register Function	MTIOC3A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 24.32 TIORL (MTU3)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRC Register Function	MTIOC3C Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU3.TMDR1.BFB bit is set to 1 and MTU3.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 24.33 TIORH (MTU4)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA Register Function	MTIOC4A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 24.34 TIORL (MTU4)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRC Register Function	MTIOC4C Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU4.TMDR1.BFB bit is set to 1 and MTU4.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 24.35 TIORH (MTU6)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA Register Function	MTIOC6A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 24.36 TIORL (MTU6)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRC Register Function	MTIOC6C Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU6.TMDR1.BFB bit is set to 1 and MTU6.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 24.37 TIORH (MTU7)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA Register Function	MTIOC7A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 24.38 TIORL (MTU7)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRC Register Function	MTIOC7C Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU7.TMDR1.BFB bit is set to 1 and MTU7.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 24.39 TIORH (MTU8)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA Register Function	MTIOC8A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Table 24.40 TIORL (MTU8)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRC Register Function	MTIOC8C Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

x: Don't care

Note 1. When the MTU8.TMDR1.BFA bit is set to 1 and the MTU8.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 24.41 TIORU, TIORV, and TIORW (MTU5)

Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[4]	IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRU, TGRV, TGRW Registers Function	MTIC5U, MTIC5V, MTIC5W Pin Function
0	0	0	0	0	Output compare register	No function
0	0	0	0	1		Setting prohibited
0	0	0	1	x		Setting prohibited
0	0	1	x	x		Setting prohibited
0	1	x	x	x		Setting prohibited
1	0	0	0	0	Input capture register*1	Setting prohibited
1	0	0	0	1		Input capture at rising edge.
1	0	0	1	0		Input capture at falling edge.
1	0	0	1	1		Input capture at both edges.
1	0	1	x	x		Input capture on generation of compare match with MTU8.TGRC
1	1	0	0	0		Setting prohibited
1	1	0	0	1		Measurement of low pulse width of external input signal. Capture at trough in complementary PWM mode.
1	1	0	1	0		Measurement of low pulse width of external input signal. Capture at crest of complementary PWM mode.
1	1	0	1	1		Measurement of low pulse width of external input signal. Capture at crest and trough of complementary PWM mode.
1	1	1	0	0		Setting prohibited
1	1	1	0	1		Measurement of high pulse width of external input signal. Capture at trough in complementary PWM mode.
1	1	1	1	0		Measurement of high pulse width of external input signal. Capture at crest of complementary PWM mode.
1	1	1	1	1		Measurement of high pulse width of external input signal. Capture at crest and trough of complementary PWM mode.

x: Don't care

Note 1. Set the IOC[4:0] bits to 19h, 1Ah, 1Bh, 1Dh, 1Eh, or 1Fh only when using external pulse width measurement or only when using dead time compensation linked with MTU6 and MTU7. For details, refer to section 24.3.11, External Pulse Width Measurement and section 24.3.12, Dead Time Compensation.

24.2.7 Timer Compare Match Clear Register (TCNTCMPCLR)

Address(es): MTU5.TCNTCMPCLR 000C 1CB6h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CMPCLR5U	CMPCLR5V	CMPCLR5W
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPCLR5W	TCNT Compare Clear 5W	0: Disables MTU5.TCNTW to be cleared to 0000h at MTU5.TCNTW and MTU5.TGRW compare match or input capture 1: Enables MTU5.TCNTW to be cleared to 0000h at MTU5.TCNTW and MTU5.TGRW compare match or input capture	R/W
b1	CMPCLR5V	TCNT Compare Clear 5V	0: Disables MTU5.TCNTV to be cleared to 0000h at MTU5.TCNTV and MTU5.TGRV compare match or input capture 1: Enables MTU5.TCNTV to be cleared to 0000h at MTU5.TCNTV and MTU5.TGRV compare match or input capture	R/W
b2	CMPCLR5U	TCNT Compare Clear 5U	0: Disables MTU5.TCNTU to be cleared to 0000h at MTU5.TCNTU and MTU5.TGRU compare match or input capture 1: Enables MTU5.TCNTU to be cleared to 0000h at MTU5.TCNTU and MTU5.TGRU compare match or input capture	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TCNTCMPCLR specifies requests to clear MTU5.TCNTU, MTU5.TCNTV, and MTU5.TCNTW. The MTU has one TCNTCMPCLR (on MTU5).

24.2.8 Timer Interrupt Enable Register (TIER)

- MTU1.TIER, MTU2.TIER

Address(es): MTU1.TIER 000C 1384h, MTU2.TIER 000C 1404h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

- MTU0.TIER, MTU3.TIER, MTU6.TIER

Address(es): MTU0.TIER 000C 1304h, MTU3.TIER 000C 1208h, MTU6.TIER 000C 1A08h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

- MTU4.TIER, MTU7.TIER

Address(es): MTU4.TIER 000C 1209h, MTU7.TIER 000C 1A09h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

- MTU8.TIER

Address(es): MTU8.TIER 000C 1604h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TGIEA	TGR Interrupt Enable A	0: Interrupt requests (TGIA) disabled 1: Interrupt requests (TGIA) enabled	R/W
b1	TGIEB	TGR Interrupt Enable B	0: Interrupt requests (TGIB) disabled 1: Interrupt requests (TGIB) enabled	R/W
b2	TGIEC	TGR Interrupt Enable C	0: Interrupt requests (TGIC) disabled 1: Interrupt requests (TGIC) enabled	R/W
b3	TGIED	TGR Interrupt Enable D	0: Interrupt requests (TGID) disabled 1: Interrupt requests (TGID) enabled	R/W
b4	TCIEV	Overflow Interrupt Enable	0: Interrupt requests (TCIV) disabled 1: Interrupt requests (TCIV) enabled	R/W
b5	TCIEU	Underflow Interrupt Enable	0: Interrupt requests (TCIU) disabled 1: Interrupt requests (TCIU) enabled	R/W
b6	TTGE2	A/D Conversion Start Request Enable 2	0: A/D conversion start request generation by MTUn.TCNT underflow (trough) disabled 1: A/D conversion start request generation by MTUn.TCNT underflow (trough) enabled	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

n = 4, 7

The TIER register enables or disables interrupt requests from each channel. The MTU has a total of ten TIER registers, two for MTU0 and one each for MTU1 to MTU8.

TGIEA and TGIEB Bits (TGR Interrupt Enable A and B)

Each bit enables or disables interrupt requests (TGIm) (m = A, B).

TGIEC and TGIED Bits (TGR Interrupt Enable C and D)

Each bit enables or disables an interrupt request (TGIm) (m = C, D).
In MTU1 and MTU2, these bits are reserved. They are read as 0. The write value should be 0.

TCIEV Bit (Overflow Interrupt Enable)

This bit enables or disables interrupt requests (TCIV).

TCIEU Bit (Underflow Interrupt Enable)

This bit enables or disables interrupt requests (TCIU).
In MTU0, MTU3, MTU4, MTU6, MTU7, and MTU8, this bit is reserved. It is read as 0. The write value should be 0.

TTGE2 Bit (A/D Conversion Start Request Enable 2)

This bit enables or disables generation of A/D conversion start requests by MTUn.TCNT underflow (trough) in complementary PWM mode (n = 4, 7).
In MTU0 to MTU3, MTU6, and MTU8, this bit is reserved. It is read as 0. The write value should be 0.

TTGE Bit (A/D Conversion Start Request Enable)

This bit enables or disables generation of A/D conversion start requests by TGRA input capture/compare match.
MTU8 is a reserved bit. It is read as 0. The write value should be 0.

- MTU0.TIER2

Address(es): MTU0.TIER2 000C 1324h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE2	—	—	—	—	—	TGIEF	TGIEE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TGIEE	TGR Interrupt Enable E	0: Interrupt requests (TGIE) disabled 1: Interrupt requests (TGIE) enabled	R/W
b1	TGIEF	TGR Interrupt Enable F	0: Interrupt requests (TGIF) disabled 1: Interrupt requests (TGIF) enabled	R/W
b6 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	TTGE2	A/D Conversion Start Request Enable 2	0: A/D conversion start request generation by compare match between MTU0.TCNT and MTU0.TGRE disabled 1: A/D conversion start request generation by compare match between MTU0.TCNT and MTU0.TGRE enabled	R/W

TGIEE and TGIEF Bits (TGR Interrupt Enable E and F)

Each bit enables or disables interrupt requests by compare match between MTU0.TCNT and MTU0.TGRm (m = E, F).

TTGE2 Bit (A/D Conversion Start Request Enable 2)

Each bit enables or disables A/D conversion start requests by compare match between MTU0.TCNT and MTU0.TGRE.

- MTU5.TIER

Address(es): MTU5.TIER 000C 1CB2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TGIE5U	TGIE5V	TGIE5W
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TGIE5W	TGR Interrupt Enable 5W	0: Interrupt requests TGIW5 disabled 1: Interrupt requests TGIW5 enabled	R/W
b1	TGIE5V	TGR Interrupt Enable 5V	0: Interrupt requests TGIV5 disabled 1: Interrupt requests TGIV5 enabled	R/W
b2	TGIE5U	TGR Interrupt Enable 5U	0: Interrupt requests TGIU5 disabled 1: Interrupt requests TGIU5 enabled	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TGIE5m Bits (TGR Interrupt Enable 5m)

Each bit enables or disables interrupt requests (TGIm5) (m = U, V, W).

24.2.9 Timer Status Register (TSR)

- MTU1.TSR, MTU2.TSR

Address(es): MTU1.TSR 000C 1385h, MTU2.TSR 000C 1405h

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

- MTU3.TSR, MTU4.TSR, MTU6.TSR, MTU7.TSR

Address(es): MTU3.TSR 000C 122Ch, MTU4.TSR 000C 122Dh, MTU6.TSR 000C 1A2Ch, MTU7.TSR 000C 1A2Dh

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

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	The read value is undefined.The write value should be 1.	R/W
b6	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7	TCFD	Count Direction Flag	0: TCNT counts down 1: TCNT counts up	R

TSR indicates the states of each of the channels. The MTU has a total of six TSR registers, one each for MTU1 to MTU4, MTU6, and MTU7.

TCFD Flag (Count Direction Flag)

Status flag that indicates the direction in which TCNT is counting in MTU1 to MTU4, MTU6, and MTU7.

24.2.10 Timer Buffer Operation Transfer Mode Register (TBTM)

- MTU0.TBTM

Address(es): MTU0.TBTM 000C 1326h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TTSE	TTSB	TTSA
Value after reset:	0	0	0	0	0	0	0	0

- MTU3.TBTM, MTU4.TBTM, MTU6.TBTM, MTU7.TBTM

Address(es): MTU3.TBTM 000C 1238h, MTU4.TBTM 000C 1239h, MTU6.TBTM 000C 1A38h, MTU7.TBTM 000C 1A39h

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

Bit	Symbol	Bit Name	Description	R/W
b0	TTSA	Timing Select A	0: When compare match A occurs in each channel, data is transferred from TGRC to TGRA 1: When TCNT is cleared in each channel, data is transferred from TGRC to TGRA	R/W
b1	TTSB	Timing Select B	0: When compare match B occurs in each channel, data is transferred from TGRD to TGRB 1: When TCNT is cleared in each channel, data is transferred from TGRD to TGRB	R/W
b2	TTSE	Timing Select E	0: When compare match E occurs in MTU0, data is transferred from MTU0.TGRF to MTU0.TGRE 1: When MTU0.TCNT is cleared, data is transferred from MTU0.TGRF to MTU0.TGRE	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TBTM specifies the timing for transferring data from the buffer register to the timer general register in PWM mode. The MTU has a total of five TBTM registers, one each for MTU0, MTU3, MTU4, MTU6, and MTU7.

TTSA Bit (Timing Select A)

This bit specifies the timing for transferring data from TGRC to TGRA in each channel when they are used together for buffer operation. When a channel is not set to PWM mode, do not set the TTSA bit in the channel to 1.

TTSB Bit (Timing Select B)

This bit specifies the timing for transferring data from TGRD to TGRB in each channel when they are used together for buffer operation. When a channel is not set to PWM mode, do not set the TTSB bit in the channel to 1.

TTSE Bit (Timing Select E)

This bit specifies the timing for transferring data from MTU0.TGRF to MTU0.TGRE when they are used together for buffer operation.

In MTU3, MTU4, MTU6, and MTU7, this bit is reserved. It is read as 0 and the write value should be 0. When a channel is not set to PWM mode, do not set the TTSE bit in the channel to 1.

24.2.11 Timer Input Capture Control Register (TICCR)

Address(es): MTU1.TICCR 000C 1390h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	I2BE	I2AE	I1BE	I1AE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	I1AE	Input Capture Enable	0: Does not include the MTIOC1A pin in the MTU2.TGRA input capture conditions 1: Includes the MTIOC1A pin in the MTU2.TGRA input capture conditions	R/W
b1	I1BE	Input Capture Enable	0: Does not include the MTIOC1B pin in the MTU2.TGRB input capture conditions 1: Includes the MTIOC1B pin in the MTU2.TGRB input capture conditions	R/W
b2	I2AE	Input Capture Enable	0: Does not include the MTIOC2A pin in the MTU1.TGRA input capture conditions 1: Includes the MTIOC2A pin in the MTU1.TGRA input capture conditions	R/W
b3	I2BE	Input Capture Enable	0: Does not include the MTIOC2B pin in the MTU1.TGRB input capture conditions 1: Includes the MTIOC2B pin in the MTU1.TGRB input capture conditions	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TICCR specifies input capture conditions when MTU1.TCNT and MTU2.TCNT are cascaded. The MTU has one TICCR for MTU1.

24.2.12 Timer Synchronous Clear Register (TSYCR)

Address(es): MTU6.TSYCR 000C 1A50h

	b7	b6	b5	b4	b3	b2	b1	b0
	CE0A	CE0B	CE0C	CE0D	CE1A	CE1B	CE2A	CE2B
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CE2B	Clear Enable 2B	0: Disables counter clearing by the MTU2.TGIB2 interrupt generation timing. 1: Enables counter clearing by the MTU2.TGIB2 interrupt generation timing.	R/W
b1	CE2A	Clear Enable 2A	0: Disables counter clearing by the MTU2.TGIA2 interrupt generation timing*1. 1: Enables counter clearing by the MTU2.TGIA2 interrupt generation timing*1.	R/W
b2	CE1B	Clear Enable 1B	0: Disables counter clearing by the MTU1.TGIB1 interrupt generation timing*1. 1: Enables counter clearing by the MTU1.TGIB1 interrupt generation timing*1.	R/W
b3	CE1A	Clear Enable 1A	0: Disables counter clearing by the MTU1.TGIA1 interrupt generation timing*1. 1: Enables counter clearing by the MTU1.TGIA1 interrupt generation timing*1.	R/W
b4	CE0D	Clear Enable 0D	0: Disables counter clearing by the MTU0.TGID0 interrupt generation timing*1. 1: Enables counter clearing by the MTU0.TGID0 interrupt generation timing*1.	R/W
b5	CE0C	Clear Enable 0C	0: Disables counter clearing by the MTU0.TGIC0 interrupt generation timing*1. 1: Enables counter clearing by the MTU0.TGIC0 interrupt generation timing*1.	R/W
b6	CE0B	Clear Enable 0B	0: Disables counter clearing by the MTU0.TGIB0 interrupt generation timing*1. 1: Enables counter clearing by the MTU0.TGIB0 interrupt generation timing*1.	R/W
b7	CE0A	Clear Enable 0A	0: Disables counter clearing by the MTU0.TGIA0 interrupt generation timing*1. 1: Enables counter clearing by the MTU0.TGIA0 interrupt generation timing*1.	R/W

Note 1. This does not depend on the TIERn.TGIEm bit setting. (n = 0, 1, 2; m = A, B, C, D)

TSYCR specifies synchronous clear conditions for MTU6.TCNT and MTU7.TCNT. The MTU has one TSYCR for MTU1.

CEnm Bits (Clear Enable nm; n = 0, 1, 2; m = A, B, C, D)

These bits enable or disable counter clearing by the MTUn.TGImn interrupt generation timing.

24.2.13 Timer Counter (TCNT)

- MTU0.TCNT to MTU7.TCNT

Address(es): MTU0.TCNT 000C 1306h, MTU1.TCNT 000C 1386h, MTU2.TCNT 000C 1406h, MTU3.TCNT 000C 1210h, MTU4.TCNT 000C 1212h, MTU5.TCNTU 000C 1C80h, MTU5.TCNTV 000C 1C90h, MTU5.TCNTW 000C 1CA0h, MTU6.TCNT 000C 1A10h, MTU7.TCNT 000C 1A12h

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

Note: TCNT must not be accessed in 8 bits; it should be accessed in 16 bits.

- MTU8.TCNT

Address(es): MTU8.TCNT 000C 1608h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Note: MTU8.TCNT must not be accessed in 8 or 16 bits; it should be accessed in 32 bits.

MTU0.TCNT to MTU7.TCNT are 16-bit readable/writable counters and MTU8.TCNT is a 32-bit readable/writable counter. The MTU has a total of 11 TCNT counters, one each for MTU0 to MTU4 and MTU6 to MTU8 and three (MTU5.TCNTU, TCNTV, and TCNTW) for MTU5. The TCNT counters in MTU0 to MTU4, MTU6, and MTU7 are initialized to 0000h by a reset, and the MTU8.TCNT counter is initialized to 00000000h by a reset. MTU5.TCNTU, MTU5.TCNTV, and MTU5.TCNTW are initialized to 0000h by a reset.
In MTU0 to MTU4, MTU6, and MTU7, the TCNT counters must not be accessed in 8-bit units; they should be accessed in 16-bit units. The MTU8.TCNT counter must not be accessed in 8- or 16-bit units; it should be accessed in 32-bit units.
The MTU1.TCNT and MTU2.TCNT counters are read as 0000h when TMDR3.LWA is 1. Refer to section 24.2.5, Timer Mode Register 3 (TMDR3) for details.

24.2.14 Timer Longword Counter (TCNTLW)

Address(es): MTU1.TCNTLW 000C 13A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Note: TCNTLW must not be accessed in 8 or 16 bits; it should be accessed in 32 bits.

The TCNTLW counter is a 32-bit readable/writable counter. Only one counter of this type is provided, and is formed by combining MTU1.TCNT and MTU2.TCNT. Such operation is only effective when TMDR3.LWA is 1. The TCNTLW counter is initialized to 0000 0000h by a reset. This counter is read as 0000 0000h when TMDR3.LWA is 0. Refer to section 24.2.5, Timer Mode Register 3 (TMDR3) for details. This register can only be used in 32-bit phase counting mode.

24.2.15 Timer General Register m (TGRm) (m = A, B, C, D, E, F, U, V, W)

- MTU0.TGR to MTU7.TGR

Address(es): MTU0.TGRA 000C 1308h, MTU0.TGRB 000C 130Ah, MTU0.TGRC 000C 130Ch, MTU0.TGRD 000C 130Eh, MTU0.TGRE 000C 1320h, MTU0.TGRF 000C 1322h,
MTU1.TGRA 000C 1388h, MTU1.TGRB 000C 138Ah,
MTU2.TGRA 000C 1408h, MTU2.TGRB 000C 140Ah,
MTU3.TGRA 000C 1218h, MTU3.TGRB 000C 121Ah, MTU3.TGRC 000C 1224h, MTU3.TGRD 000C 1226h, MTU3.TGRE 000C 1272h,
MTU4.TGRA 000C 121Ch, MTU4.TGRB 000C 121Eh, MTU4.TGRC 000C 1228h, MTU4.TGRD 000C 122Ah, MTU4.TGRE 000C 1274h,
MTU4.TGRF 000C 1276h,
MTU5.TGRU 000C 1C82h, MTU5.TGRV 000C 1C92h, MTU5.TGRW 000C 1CA2h,
MTU6.TGRA 000C 1A18h, MTU6.TGRB 000C 1A1Ah, MTU6.TGRC 000C 1A24h, MTU6.TGRD 000C 1A26h, MTU6.TGRE 000C 1A72h,
MTU7.TGRA 000C 1A1Ch, MTU7.TGRB 000C 1A1Eh, MTU7.TGRC 000C 1A28h, MTU7.TGRD 000C 1A2Ah, MTU7.TGRE 000C 1A74h,
MTU7.TGRF 000C 1A76h

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

Note: TGR must not be accessed in 8 bits; it should be accessed in 16 bits. The initial value of TGR is FFFFh.

- MTU8.TGR

Address(es): MTU8.TGRA 000C 160Ch, MTU8.TGRB 000C 1610h, MTU8.TGRC 000C 1614h, MTU8.TGRD 000C 1618h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

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

Note: MTU8.TGR must not be accessed in 8 or 16 bits; it should be accessed in 32 bits.

The MTU0.TGR to MTU7.TGR registers are 16-bit readable/writable registers; the MTU8.TGR register is a 32-bit readable/writable register. The MTU has a total of 39 TGR registers, six for MTU0, two each for MTU1 and MTU2, five each for MTU3 and MTU6, six each for MTU4 and MTU7, and three for MTU5, and four for MTU8.
The TGRA, TGRB, TGRC, and TGRD registers function as either output compare or input capture registers. The TGRC and TGRD registers for MTU0, MTU3, MTU4, MTU6, MTU7, and MTU8 can also be designated for operation as buffer registers. TGR buffer register combinations are TGRA and TGRC, and TGRB and TGRD.
MTU0.TGRE and MTU0.TGRF function as compare registers. When the MTU0.TCNT count matches the MTU0.TGRE value, an A/D conversion start request can be issued. The TGRF register can also be designated for operation as a buffer register. TGR buffer register combination is TGRE and TGRF. MTU5.TGRU, MTU5.TGRV, and MTU5.TGRW function as compare match, input capture, or external pulse width measurement registers.
The MTU1.TGRA, MTU2.TGRA, MTU1.TGRB, and MTU2.TGRB registers are read as 0000h when TMDR3.LWA is 1. Refer to section 24.2.5, Timer Mode Register 3 (TMDR3) for details.

24.2.16 Timer Longword General Register m (TGRmLW) (m = A, B)

Address(es): MTU1.TGRALW 000C 13A4h, MTU1.TGRBLW 000C 13A8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

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

Note: TGRALW and TGRBLW must not be accessed in 8 or 16 bits; it should be accessed in 32 bits.

The TGRmLW register is a 32-bit readable/writable register. Two general registers of this type are provided, and are formed by combining MTU1.TGRm and MTU2.TGRm. Such operation is only effective when TMDR3.LWA is 1.
The TGRmLW register is initialized to FFFF FFFFh by a reset, but it is read as 0000 0000h when TMDR3.LWA is 0. Refer to section 24.2.5, Timer Mode Register 3 (TMDR3) for details.
The TGRALW and TGRBLW registers function as input capture registers which can only be used in 32-bit phase counting mode.

24.2.17 Timer Start Registers (TSTRA, TSTRB, TSTR)

- MTU.TSTRA (for MTU0, MTU1, MTU2, MTU3, MTU4, and MTU8)

Address(es): MTU.TSTRA 000C 1280h

	b7	b6	b5	b4	b3	b2	b1	b0
	CST4	CST3	—	—	CST8	CST2	CST1	CST0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CST0	Counter Start 0	0: MTU0.TCNT counting is stopped 1: MTU0.TCNT performs count operation	R/W
b1	CST1	Counter Start 1	0: MTU1.TCNT counting is stopped 1: MTU1.TCNT performs count operation	R/W
b2	CST2	Counter Start 2	0: MTU2.TCNT counting is stopped 1: MTU2.TCNT performs count operation	R/W
b3	CST8	Counter start 8	0: MTU8.TCNT counting is stopped 1: MTU8.TCNT performs count operation.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CST3	Counter Start 3	0: MTU3.TCNT counting is stopped 1: MTU3.TCNT performs count operation	R/W
b7	CST4	Counter Start 4	0: MTU4.TCNT counting is stopped 1: MTU4.TCNT performs count operation	R/W

Note: When 1 is written to a bit in TCSYSTR, the corresponding bit in TSTRA is also set to 1 automatically.

The TSTRA register starts or stops TCNT operation in MTU0 to MTU4 and MTU8.
TSTRB starts or stops TCNT operation in MTU6 and MTU7.
TSTR starts or stops TCNT operation in MTU5.
Before setting the operating mode in TMDR1 or setting the TCNT count clock in TCR, be sure to stop the TCNT counter.

CSTn Bits (Counter Start n) (n = 0, 1, 2, 3, 4, 8)

Each bit starts or stops TCNT in the corresponding channel.
If 0 is written to the CSTn bit during operation with the MTIOC pin designated for output, the counter stops. At this time, initial output level specified in the TOCR1A or TOCR2A register is output from the MTIOC pin in complementary PWM mode or reset-synchronized PWM mode. In any mode other than complementary PWM mode and reset synchronous PWM mode, the output compare signal level from the MTIOC pin is retained. If TIOR is written to while the CSTn bit is 0, the pin output level will be changed to the specified initial output value.

- MTU.TSTRB (for MTU6 and MTU7)

Address(es): MTU.TSTRB 000C 1A80h

	b7	b6	b5	b4	b3	b2	b1	b0
	CST7	CST6	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CST6	Counter Start 6	0: MTU6.TCNT counting is stopped 1: MTU6.TCNT performs count operation	R/W
b7	CST7	Counter Start 7	0: MTU7.TCNT counting is stopped 1: MTU7.TCNT performs count operation	R/W

Note: When 1 is written to a bit in TCSYSTR, the corresponding bit in TSTRB is also set to 1 automatically.

CSTn Bits (Counter Start n) (n = 6, 7)

Each bit starts or stops TCNT in the corresponding channel.
If 0 is written to the CSTn bit during operation with the MTIOC pin designated for output, the counter stops. At this time the MTIOC pin output the initial output level set in the TOCR1B or TOCR2B register in complementary PWM mode or reset-synchronized PWM mode, but the output compare signal level from the MTIOC pin is retained in the other modes.
If TIOR is written to while the CSTn bit is 0, the pin output level will be changed to the specified initial output value.

- MTU5.TSTR

Address(es): MTU5.TSTR 000C 1CB4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CSTU5	CSTV5	CSTW5
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CSTW5	Counter Start W5	0: MTU5.TCNTW counting is stopped 1: MTU5.TCNTW performs count operation	R/W
b1	CSTV5	Counter Start V5	0: MTU5.TCNTV counting is stopped 1: MTU5.TCNTV performs count operation	R/W
b2	CSTU5	Counter Start U5	0: MTU5.TCNTU counting is stopped 1: MTU5.TCNTU performs count operation	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

24.2.18 Timer Synchronous Register m (TSYRm) (m = A, B)

- MTU.TSYRA (for MTU0 to MTU4)

Address(es): MTU.TSYRA 000C 1281h

	b7	b6	b5	b4	b3	b2	b1	b0
	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SYNC0	Timer Synchronous Operation 0	0: MTU0.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU0.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W
b1	SYNC1	Timer Synchronous Operation 1	0: MTU1.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU1.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W
b2	SYNC2	Timer Synchronous Operation 2	0: MTU2.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU2.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W
b5 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	SYNC3	Timer Synchronous Operation 3	0: MTU3.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU3.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W
b7	SYNC4	Timer Synchronous Operation 4	0: MTU4.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU4.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W

TSYRA selects independent operation or synchronous operation of TCNT in MTU0 to MTU4.
TSYRB selects independent operation or synchronous operation of TCNT in MTU6 and MTU7.
A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

SYNCn Bits (Timer Synchronous Operation n) (n = 0, 1, 2, 3, 4)

Each bit selects whether operation is independent of or synchronized with other channels.
When synchronous operation is selected, the TCNT synchronous setting of multiple channels and synchronous clearing by counter clearing on another channel are possible.
To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of the TCR.CCLR[2:0] bits.

- MTU.TSYRB (for MTU6 and MTU7)

Address(es): MTU.TSYRB 000C 1A81h

	b7	b6	b5	b4	b3	b2	b1	b0
	SYNC7	SYNC6	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	SYNC6	Timer Synchronous Operation 6	0: MTU6.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU6.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W
b7	SYNC7	Timer Synchronous Operation 7	0: MTU7.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU7.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W

SYNCn Bits (Timer Synchronous Operation n) (n = 6, 7)

Each bit selects whether operation is independent of or synchronized with other channels.

When synchronous operation is selected, the TCNT synchronous setting of multiple channels and synchronous clearing by counter clearing on another channel are possible.

To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of bits TCR.CCLR[2:0].

24.2.19 Timer Counter Synchronous Start Register (TCSYSTR)

Address(es): MTU.TCSYSTR 000C 1282h

	b7	b6	b5	b4	b3	b2	b1	b0
	SCH0	SCH1	SCH2	SCH3	SCH4	—	SCH6	SCH7
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SCH7	Synchronous Start 7	0: Does not specify synchronous start for MTU7.TCNT 1: Specifies synchronous start for MTU7.TCNT	R/(W)*1
b1	SCH6	Synchronous Start 6	0: Does not specify synchronous start for MTU6.TCNT 1: Specifies synchronous start for MTU6.TCNT	R/(W)*1
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R
b3	SCH4	Synchronous Start 4	0: Does not specify synchronous start for MTU4.TCNT 1: Specifies synchronous start for MTU4.TCNT	R/(W)*1
b4	SCH3	Synchronous Start 3	0: Does not specify synchronous start for MTU3.TCNT 1: Specifies synchronous start for MTU3.TCNT	R/(W)*1
b5	SCH2	Synchronous Start 2	0: Does not specify synchronous start for MTU2.TCNT 1: Specifies synchronous start for MTU2.TCNT	R/(W)*1
b6	SCH1	Synchronous Start 1	0: Does not specify synchronous start for MTU1.TCNT 1: Specifies synchronous start for MTU1.TCNT	R/(W)*1
b7	SCH0	Synchronous Start 0	0: Does not specify synchronous start for MTU0.TCNT 1: Specifies synchronous start for MTU0.TCNT	R/(W)*1

Note 1. Only 1 can be written to this bit. This bit is automatically cleared when the corresponding counter starts.

TCSYSTR specifies synchronous start of the counters.

SCH7 Bit (Synchronous Start 7)

This bit controls synchronous start of MTU7.TCNT.

[Clearing condition]

- When 1 is set to the TSTRB.CST7 bit while SCH7 = 1

SCH6 Bit (Synchronous Start 6)

This bit controls synchronous start of MTU6.TCNT.

[Clearing condition]

- When 1 is set to the TSTRB.CST6 bit while SCH6 = 1

SCH4 Bit (Synchronous Start 4)

This bit controls synchronous start of MTU4.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST4 bit while SCH4 = 1

SCH3 Bit (Synchronous Start 3)

This bit controls synchronous start of MTU3.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST3 bit while SCH3 = 1

SCH2 Bit (Synchronous Start 2)

This bit controls synchronous start of MTU2.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST2 bit while SCH2 = 1

SCH1 Bit (Synchronous Start 1)

This bit controls synchronous start of MTU1.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST1 bit while SCH1 = 1

SCH0 Bit (Synchronous Start 0)

This bit controls synchronous start of MTU0.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST0 bit while SCH0 = 1

24.2.20 Timer Read/Write Enable Register m (TRWERm) (m = A, B)

Address(es): MTU.TRWERA 000C 1284h, MTU.TRWERB 000C 1A84h

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

Bit	Symbol	Bit Name	Description	R/W
b0	RWE	Read/Write Enable	0: Read/write access to the registers is disabled 1: Read/write access to the registers is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TRWERA enables or disables access to the registers and counters that have write-protection capability against accidental modification in MTU3 and MTU4.
TRWERB enables or disables access to the registers and counters that have write-protection capability against accidental modification in MTU6 and MTU7.

RWE Bit (Read/Write Enable)

This bit enables or disables access to the registers that have write-protection capability against accidental modification.
[Clearing condition]

- When 0 is written to the RWE bit after reading RWE = 1

- Registers and Counters having Write-Protection Capability against Accidental Modification (TRWERA)

24 registers: MTUn.TCR, MTUn.TCR2, MTUn.TMDR1, MTUn.TIORH, MTUn.TIORL, MTUn.TIER, MTUn.TGRA, MTUn.TGRB, MTU.TOERA, MTU.TOCR1A, MTU.TOCR2A, MTU.TGCRA, MTU.TCDRA, MTU.TDDRA, and MTUn.TCNT (n = 3, 4)

- Registers and Counters having Write-Protection Capability against Accidental Modification (TRWERB)

23 registers: MTUn.TCR, MTUn.TCR2, MTUn.TMDR1, MTUn.TIORH, MTUn.TIORL, MTUn.TIER, MTUn.TGRA, MTUn.TGRB, MTU.TOERB, MTU.TOCR1B, MTU.TOCR2B, MTU.TCDRB, MTU.TDDRB, and MTUn.TCNT (n = 6, 7)

24.2.21 Timer Output Master Enable Register m (TOERm) (m = A, B)

- MTU.TOERA

Address(es): MTU.TOERA 000C 120Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
Value after reset:	1	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OE3B	Master Enable MTIOC3B	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b1	OE4A	Master Enable MTIOC4A	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b2	OE4B	Master Enable MTIOC4B	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b3	OE3D	Master Enable MTIOC3D	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b4	OE4C	Master Enable MTIOC4C	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b5	OE4D	Master Enable MTIOC4D	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b7, b6	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. To output the inactive level from each pin when the MTU output is set to disabled, first set the data direction register (PDR) and port output data register (PODR) of I/O ports to output the inactive level from general I/O ports, and then set the port mode register (PMR) to use general I/O ports. For details, refer to section 22, I/O Ports.

TOERA enables or disables output settings for output pins MTIOC4D, MTIOC4C, MTIOC3D, MTIOC4B, MTIOC4A, and MTIOC3B.

These pins do not output correctly if the bits in the TOERA register have not been set. In MTU3 and MTU4, set TOERA prior to setting TIOR.

Set MTU.TOERA after setting the CST3 and CST4 bits in MTU.TSTRA to 0 (refer to Figure 24.44 and Figure 24.48).

- MTU.TOERB

Address(es): MTU.TOERB 000C 1A0Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	OE7D	OE7C	OE6D	OE7B	OE7A	OE6B
Value after reset:	1	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OE6B	Master Enable MTIOC6B	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b1	OE7A	Master Enable MTIOC7A	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b2	OE7B	Master Enable MTIOC7B	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b3	OE6D	Master Enable MTIOC6D	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b4	OE7C	Master Enable MTIOC7C	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b5	OE7D	Master Enable MTIOC7D	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b7, b6	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. To output the inactive level from each pin when the MTU output is set to disabled, first set the data direction register (PDR) and port output data register (PODR) of I/O ports to output the inactive level from general I/O ports, and then set the port mode register (PMR) to use general I/O ports. For details, refer to section 22, I/O Ports.

TOERB enables or disables output settings for output pins MTIOC7D, MTIOC7C, MTIOC6D, MTIOC7B, MTIOC7A, and MTIOC6B.

These pins do not output correctly if the bits in the TOERB register have not been set. In MTU6, and MTU7, set TOERB prior to setting TIOR.

Set MTU.TOERB after setting the CST6 and CST7 bits in MTU.TSTRB to 0 (refer to Figure 24.44 and Figure 24.48).

24.2.22 Timer Output Control Register 1m (TOCR1m) (m = A, B)

Address(es): MTU.TOCR1A 000C 120Eh, MTU.TOCR1B 000C 1A0Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OLSP	Output Level Select P*1, *3	Refer to Table 24.42.	R/W
b1	OLSN	Output Level Select N*1, *3	Refer to Table 24.43.	R/W
b2	TOCS	TOC Select	0: TOCR1m setting is selected (m = A, B) 1: TOCR2m setting is selected	R/W
b3	TOCL	TOC Register Write Protection*2, *4	0: Write access to the TOCS, OLSN, and OLSP bits is enabled 1: Write access to the TOCS, OLSN, and OLSP bits is disabled	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	PSYE	PWM Synchronous Output Enable	0: Toggle output is disabled 1: Toggle output is enabled	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Setting the TOCR1m.TOCS bit to 0 makes this bit setting valid.
Note 2. Setting the TOCR1m.TOCL bit to 1 prevents accidental modification when the CPU goes out of control.
Note 3. If dead-time is not generated, the negative-phase output is the exact inverse of the positive-phase output. In this case, only the OLSP bit is valid.
Note 4. This bit can be set to 1 only once after a reset. After 1 is written, 0 cannot be written to the bit.

TOCR1A and TOCR1B enable or disable PWM-synchronized toggle output in complementary PWM mode and reset-synchronized PWM mode, and control inversion of PWM output level.

OLSP Bit (Output Level Select P)

This bit selects the positive-phase output level in reset-synchronized PWM mode and complementary PWM mode. The initial output is selected while the counter is stopped.

OLSN Bit (Output Level Select N)

This bit selects the negative-phase output level in reset-synchronized PWM mode and complementary PWM mode. The initial output is selected while the counter is stopped.

TOCS Bit (TOC Select)

This bit selects either the TOCR1m or TOCR2m (m = A, B) setting to be used for the output level in complementary PWM mode and reset-synchronized PWM mode.

TOCL Bit (TOC Register Write Protection)

This bit enables or disables write access to the TOCS, OLSN, and OLSP bits in TOCR1m (m = A, B).

PSYE Bit (PWM Synchronous Output Enable)

This bit enables or disables toggle output synchronized with the PWM period from the MTIOC3A or MTIOC6A pin.

Table 24.42 Output Level Select Function

Bit 0	Function			
			Compare Match Output	
OLSP	Initial Output	Active Level	Up-Counting	Down-Counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Table 24.43 Output Level Select Function

Bit 1	Function			
			Compare Match Output	
OLSN	Initial Output	Active Level	Up-Counting	Down-Counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Figure 24.3 shows an example of output in complementary PWM mode (one phase) when OLSN = 1 and OLSP = 1.

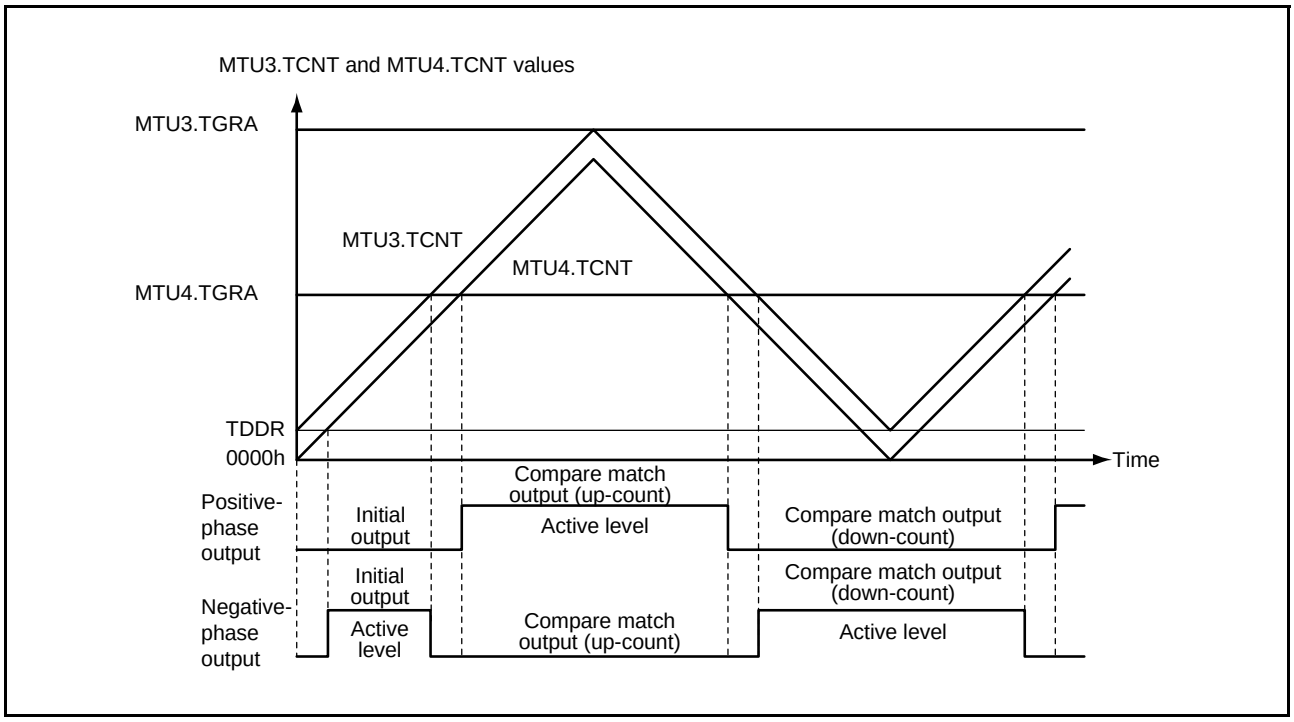


Figure 24.3 Example of Output in Complementary PWM Mode

24.2.23 Timer Output Control Register 2m (TOCR2m) (m = A, B)

Address(es): MTU.TOCR2A 000C 120Fh, MTU.TOCR2B 000C 1A0Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OLS1P	Output Level Select 1P*1, *2	This bit selects the output level on MTIOC3B or MTIOC6B in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 24.44.	R/W
b1	OLS1N	Output Level Select 1N*1, *2	This bit selects the output level on MTIOC3D or MTIOC6D in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 24.45.	R/W
b2	OLS2P	Output Level Select 2P*1, *2	This bit selects the output level on MTIOC4A or MTIOC7A in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 24.46.	R/W
b3	OLS2N	Output Level Select 2N*1, *2	This bit selects the output level on MTIOC4C or MTIOC7C in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 24.47.	R/W
b4	OLS3P	Output Level Select 3P*1, *2	This bit selects the output level on MTIOC4B or MTIOC7B in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 24.48.	R/W
b5	OLS3N	Output Level Select 3N*1, *2	This bit selects the output level on MTIOC4D or MTIOC7D in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 24.49.	R/W
b7, b6	BF[1:0]	TOLBR Buffer Transfer Timing Select	These bits select the timing for transferring data from TOLBRm to TOCR2m. Refer to Table 24.50 for details.	R/W

m = A, B

Note 1. Setting the TOCR1m.TOCS bit to 1 makes this bit setting valid.

Note 2. If dead-time is not generated, the negative-phase output is the exact inverse of the positive-phase output. In this case, only the OLSiP bits are valid (i = 1 to 3).

TOCR2A and TOCR2B control inversion of PWM output level in complementary PWM mode and reset-synchronized PWM mode.

The initial output is selected while the counter is stopped.

Table 24.44 MTIOCnB 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

n = 3, 6

Table 24.45 MTIOCnD 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

n = 3, 6

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 24.46 MTIOCnA 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

n = 4, 7

Table 24.47 MTIOCnC 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

n = 4, 7

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 24.48 MTIOCnB 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

n = 4, 7

Table 24.49 MTIOCnD 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

n = 4, 7

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 24.50 Setting of TOCR2m.BF[1:0] Bits

Bit 7	Bit 6	Description	
BF[1]	**BF[0]**	**Complementary PWM Mode**	**Reset-Synchronized PWM Mode**
0	0	Does not transfer data from the buffer register (TOLBRm) to TOCR2m.	Does not transfer data from the buffer register (TOLBRm) to TOCR2m.
0	1	Transfers data from the buffer register (TOLBRm) to TOCR2m at the crest of the MTUn.TCNT count.	Transfers data from the buffer register (TOLBRm) to TOCR2m when MTUk.TCNT or MTUn.TCNT is cleared.
1	0	Transfers data from the buffer register (TOLBRm) to TOCR2m at the trough of the MTUn.TCNT count.	Setting prohibited
1	1	Transfers data from the buffer register (TOLBRm) to TOCR2m at the crest and trough of the MTUn.TCNT count.	Setting prohibited

n = 4, 7; k = 3, 6; m = A, B

24.2.24 Timer Output Level Buffer Register m (TOLBRm) (m = A, B)

Address(es): MTU.TOLBRA 000C 1236h, MTU.TOLBRB 000C 1A36h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OLS1P	Output Level Select 1P	Specify the buffer value to be transferred to the OLS1P bit in TOCR2m.	R/W
b1	OLS1N	Output Level Select 1N	Specify the buffer value to be transferred to the OLS1N bit in TOCR2m.	R/W
b2	OLS2P	Output Level Select 2P	Specify the buffer value to be transferred to the OLS2P bit in TOCR2m.	R/W
b3	OLS2N	Output Level Select 2N	Specify the buffer value to be transferred to the OLS2N bit in TOCR2m.	R/W
b4	OLS3P	Output Level Select 3P	Specify the buffer value to be transferred to the OLS3P bit in TOCR2m.	R/W
b5	OLS3N	Output Level Select 3N	Specify the buffer value to be transferred to the OLS3N bit in TOCR2m.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

m = A, B

TOLBRA and TOLBRB are buffer registers for TOCR2A and TOCR2B and specify the PWM output level in complementary PWM mode and reset-synchronized PWM mode.

Figure 24.4 shows an example of the PWM output level setting procedure in buffer operation.

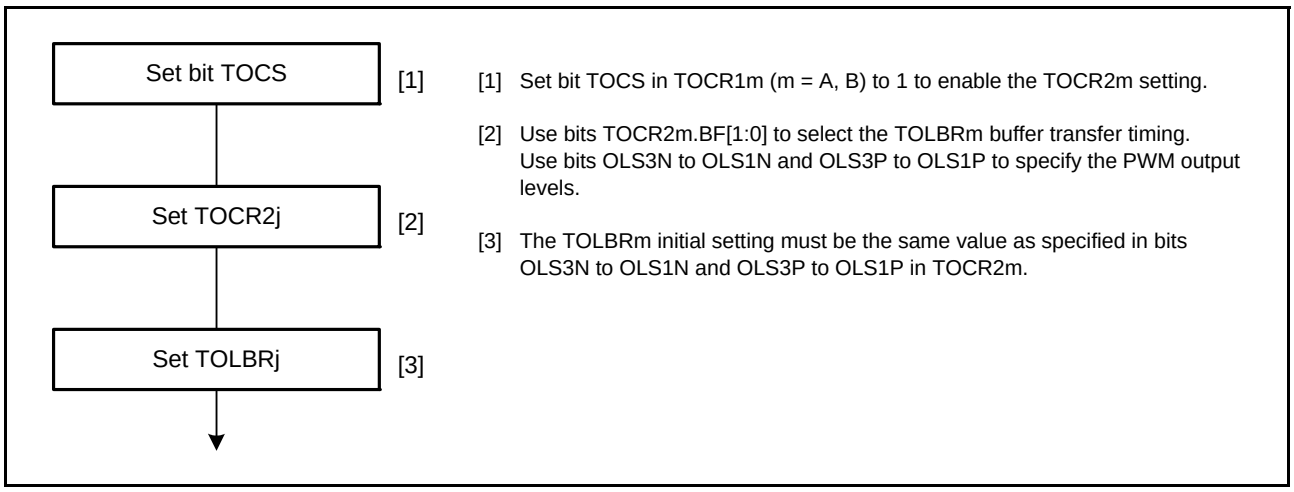


Figure 24.4 Example of PWM Output Level Setting Procedure in Buffer Operation

24.2.25 Timer Gate Control Register A (TGCRA)

Address(es): MTU.TGCRA 000C 120Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	BDC	N	P	FB	WF	VF	UF
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	UF	Output Phase Switch	These bits turn on or off the positive-phase/negative-phase output. The setting of these bits is valid only when the FB bit is set to 1. In this case, the setting of b0 to b2 is used instead of the external input. Refer to Table 24.51.	R/W
b1	VF			R/W
b2	WF			R/W
b3	FB	External Feedback Signal Enable	0: Output is switched by external input (input sources are TGRA, TGRB, and TGRC input capture signals in MTU0) 1: Output is switched by software (TGCRA's UF, VF, and WF settings)	R/W
b4	P	Positive-Phase Output (P) Control	0: Level output 1: Reset-synchronized PWM or complementary PWM output	R/W
b5	N	Negative-Phase Output (N) Control	0: Level output 1: Reset-synchronized PWM or complementary PWM output	R/W
b6	BDC	Brushless DC Motor	0: Ordinary output 1: Functions of this register are made effective	R/W
b7	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

TGCRA controls the output waveform necessary for brushless DC motor control in reset-synchronized PWM mode and complementary PWM mode. TGCRA register settings are ineffective for anything other than complementary PWM mode and reset-synchronized PWM mode.

UF, VF, and WF Bits (Output Phase Switch)

The setting of these bits is valid only when the FB bit is set to 1. In this case, the setting of b0 to b2 is used instead of the external input. Refer to Table 24.51 for details.

FB Bit (External Feedback Signal Enable)

This bit selects whether the positive-/negative-phase output is switched automatically with the TGRA, TGRB, and TGRC input capture signals in MTU0 or by writing 0 or 1 to bits 2 to 0 in TGCRA.
When the TGCRA.FB bit is 0, output of MTU3 and MTU4 can be switched with the TGRA, TGRB, and TGRC input capture signals in MTU0.

P Bit (Positive-Phase Output (P) Control)

This bit selects the level output or the reset-synchronized PWM/complementary PWM output for the positive-phase output pins (MTIOC3B, MTIOC4A, and MTIOC4B pins).

N Bit (Negative-Phase Output (N) Control)

This bit selects the level output or the reset-synchronized PWM/complementary PWM output for the negative-phase output pins (MTIOC3D, MTIOC4C, and MTIOC4D pins).

BDC Bit (Brushless DC Motor)

This bit selects whether to make the functions of TGCRA effective or ineffective.

Table 24.51 Output Level Select Function

Bit 2	Bit 1	Bit 0	Function					
			MTIOC3B	MTIOC4A	MTIOC4B	MTIOC3D	MTIOC4C	MTIOC4D
WF	VF	UF	U Phase	V Phase	W Phase	U Phase	V Phase	W Phase
0	0	0	OFF	OFF	OFF	OFF	OFF	OFF
0	0	1	ON	OFF	OFF	OFF	OFF	ON
0	1	0	OFF	ON	OFF	ON	OFF	OFF
0	1	1	OFF	ON	OFF	OFF	OFF	ON
1	0	0	OFF	OFF	ON	OFF	ON	OFF
1	0	1	ON	OFF	OFF	OFF	ON	OFF
1	1	0	OFF	OFF	ON	ON	OFF	OFF
1	1	1	OFF	OFF	OFF	OFF	OFF	OFF

24.2.26 Timer Subcounter m (TCNTSm) (m = A, B)

Address(es): MTU.TCNTSA 000C 1220h, MTU.TCNTSB 000C 1A20h

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

Note: TCNTSA and TCNTSB must not be accessed in 8 bits; it should be accessed in 16 bits.

TCNTSA and TCNTSB are 16-bit read-only counters used only in complementary PWM mode.
The initial value of TCNTSA and TCNTSB after a reset is 0000h.

24.2.27 Timer Period Data Register m (TCDRm) (m = A, B)

Address(es): MTU.TCDRA 000C 1214h, MTU.TCDRB 000C 1A14h

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

Note: TCDRA and TCDRB must not be accessed in 8 bits; it should be accessed in 16 bits.

TCDRA and TCDRB are 16-bit readable/writable registers used only in complementary PWM mode. Set half the PWM carrier period as the TCDRA and TCDRB values. The TCDRA and TCDRB registers are constantly compared with the TCNTSA and TCNTSB counters in complementary PWM mode, respectively. When a match occurs, the TCNTSA and TCNTSB counters switch the count direction (down-count to up-count). The initial value of TCDRA and TCDRB after a reset is FFFFh.

24.2.28 Timer Period Buffer Register m (TCBRm) (m = A, B)

Address(es): MTU.TCBRA 000C 1222h, MTU.TCBRB 000C 1A22h

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

Note: TCBRA and TCBRB must not be accessed in 8 bits; it should be accessed in 16 bits.

TCBRA and TCBRB are 16-bit readable/writable registers, used only in complementary PWM mode, that function as buffer registers for TCDRA and TCDRB. The TCBRA and TCBRB values are transferred to TCDRA and TCDRB with the transfer timing set in TMDR1. The initial value of TCBRA and TCBRB after a reset is FFFFh.

24.2.29 Timer Dead Time Data Register m (TDDRm) (m = A, B)

Address(es): MTU.TDDRA 000C 1216h, MTU.TDDRB 000C 1A16h

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

Note: TDDRA and TDDRB must not be accessed in 8 bits; it should be accessed in 16 bits.

TDDRA and TDDRB are 16-bit readable/writable registers, used only in complementary PWM mode, that specify the MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT) counter offset value. In complementary PWM mode, when the MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT) counters are cleared and then restarted, the TDDRA (TDDRB) value is loaded into the MTU3.TCNT (MTU6.TCNT) counter and the count operation starts. The initial value of TDDRA and TDDRB after a reset is FFFFh.

24.2.30 Timer Dead Time Enable Register m (TDERm) (m = A, B)

Address(es): MTU.TDERA 000C 1234h, MTU.TDERB 000C 1A34h

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

Bit	Symbol	Bit Name	Description	R/W
b0	TDER	Dead Time Enable	0: No dead time is generated 1: Dead time is generated*1	R/(W)
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. TDDRA and TDDRB must be set to 1 or a larger value.

TDERA and TDERB control dead time generation in complementary PWM mode. The MTU has one TDER each for MTU3 and MTU6. TDERA and TDERB should be modified only while TCNT stops.

TDER Bit (Dead Time Enable)

This bit specifies whether to generate dead time.

[Clearing condition]

- When 0 is written to TDER after reading TDER = 1

24.2.31 Timer Buffer Transfer Set Register m (TBTERm) (m = A, B)

Address(es): MTU.TBTERA 000C 1232h, MTU.TBTERB 000C 1A32h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	BTE[1:0]	Buffer Transfer Disable and Interrupt Skipping Link Setting	These bits enable or disable transfer from the buffer registers*1 used in complementary PWM mode to the temporary registers, and specify whether to link the transfer with interrupt skipping function 1. For details, refer to Table 24.52.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Applicable buffer registers (TBTERA):
MTU3.TGRC, MTU3.TGRD, MTU4.TGRC, MTU4.TGRD, and TCBRA
Applicable buffer registers (TBTERB):
MTU6.TGRC, MTU6.TGRD, MTU7.TGRC, MTU7.TGRD, and TCBRB

TBTERA and TBTERB enable or disable transfer from the buffer registers used in complementary PWM mode to the temporary registers, and specify whether to link the transfer with interrupt skipping 1 operation.

Table 24.52 Setting of TBTERA.BTE[1:0] Bits and TBTERB.BTE[1:0] Bits

Bit 1	Bit 0	
BTE[1]	BTE[0]	Description
0	0	Enables transfer from the buffer registers to the temporary registers*1 and does not link the transfer with interrupt skipping function 1.
0	1	Disables transfer from the buffer registers to the temporary registers.
1	0	Links transfer from the buffer registers to the temporary registers with interrupt skipping function 1.*2
1	1	Setting prohibited

Note 1. Data is transferred according to the MD3 to MD0 bit setting in TMDR1. For details, refer to section 24.3.8, Complementary PWM Mode.

Note 2. When interrupt skipping is disabled the T3AEN and T4VEN (T6AEN and T7VEN) bits are set to 0 in the timer interrupt skipping set register (TITCR1A (TITCR1B)) or the skipping count set bits (T3ACOR and T4VCOR (T6ACOR and T7VCOR)) in TITCR1A (TITCR1B) are set to 0), be sure to disable link of buffer transfer with interrupt skipping (set the BTE1 bit in the timer buffer transfer set register (TBTERA (TBTERB)) to 0). If link with interrupt skipping is enabled while interrupt skipping is disabled, buffer transfer will not be performed.

24.2.32 Timer Waveform Control Register m (TWCRm) (m = A, B)

Address(es): MTU.TWCRA 000C 1260h, MTU.TWCRB 000C 1A60h

	b7	b6	b5	b4	b3	b2	b1	b0
	CCE	—	—	—	—	—	SCC	WRE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WRE	Waveform Retain Enable	0: Initial values specified in TOCR1A and TOCR2A (TOCR1B and TOCR2B) are output 1: Initial output is inhibited	R/(W) *3
b1	SCC	Synchronous Clearing Control *1, *3	(Only valid in TWCRB) 0: Clearing of MTU6.TCNT and MTU7.TCNT in response to synchronous clearing for MTU0, MTU1, MTU2–MTU6, MTU7 is enabled. 1: Clearing of MTU6.TCNT and MTU7.TCNT in response to synchronous clearing for MTU0, MTU1, MTU2–MTU6, MTU7 is disabled.	R/(W)
b6 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	CCE	Compare Match Clear Enable *2	0: Counters are not cleared at MTU3.TGRA (MTU6.TGRA) compare match 1: Counters are cleared at MTU3.TGRA (MTU6.TGRA) compare match	R/(W)

Note 1. This bit is only valid in register TWCRB and is a reserved bit in register TWCRA.
Note 2. Do not set to 1 when complementary PWM mode 1 is not selected.
Note 3. Do not set to 1 when complementary PWM mode is not selected.

TWCRA and TWCRB control the output waveform when synchronous counter clearing occurs in MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) in complementary PWM mode and specifies whether to clear the counters at MTU3.TGRA (MTU6.TGRA) compare match.
The CCE bit and WRE bit in TWCRA and TWCRB should be modified only while TCNT stops.

WRE Bit (Waveform Retain Enable)

This bit selects the waveform output when synchronous counter clearing occurs in complementary PWM mode.
The initial output is inhibited with this function only when synchronous clearing occurs within the Tb interval at the trough in complementary PWM mode. When synchronous clearing occurs outside this interval, the initial values specified in TOCR1A and TOCR2A (TOCR1B and TOCR2B) are output regardless of the WRE bit setting. The initial values specified in TOCR1A and TOCR2A (TOCR1B and TOCR2B) are also output when synchronous clearing occurs in the Tb interval at the trough immediately after MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) start operation.
For the Tb interval at the trough in complementary PWM mode, refer to Figure 24.50.
[Setting condition]

- When 1 is written to the WRE bit after reading WRE = 0

SCC Bit (Synchronous Clearing Control)

The setting of this bit selects whether MTU6.TCNT and MTU7.TCNT are or are not cleared when counter-synchronous clearing is generated for MTU0, MTU1, MTU2–MTU6, MTU7 in complementary PWM mode.
Make the complementary PWM mode settings for MTU6 and MTU7 when this function is in use. When writing a new value to the SCC bit while the counter is operating, do so in such a way that the values of the CCE and WRE bits are not changed.
Synchronous clearing from the MTU module only becomes disabled due to the setting of the SCC bit when synchronous clearing is generated outside the Tb interval in the trough. If synchronous clearing is generated within the Tb interval in the trough including immediately after the value at which MTU6.TCNT and MTU7.TCNT start, MTU6.TCNT and MTU7.TCNT are cleared.
Regarding the Tb interval in the trough in complementary PWM mode, refer to Figure 24.50.
[Setting condition]

- Writing of 1 to the SCC bit after reading it as 0

The corresponding bit in register TWCRA is reserved and is read as 0. When writing to TWCRA, write 0 to this bit.

CCE Bit (Compare Match Clear Enable)

This bit specifies whether to clear counters at MTU3.TGRA (MTU6.TGRA) compare match in complementary PWM mode.
[Setting condition]

- When 1 is written to CCE after reading CCE = 0

24.2.33 Noise Filter Control Register n (NFCRn) (n = 0 to 4, 6, 7, 8, C)

- MTU0.NFCR0, MTU1.NFCR1, MTU2.NFCR2, MTU3.NFCR3, MTU4.NFCR4, MTU6.NFCR6, MTU7.NFCR7, MTU8.NFCR8

Address(es): MTU0.NFCR0 000C 1290h, MTU1.NFCR1 000C 1291h, MTU2.NFCR2 000C 1292h, MTU3.NFCR3 000C 1293h, MTU4.NFCR4 000C 1294h, MTU6.NFCR6 000C 1A93h, MTU7.NFCR7 000C 1A94h, MTU8.NFCR8 000C 1298h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	NFCS[1:0]		NFDEN	NFCEN	NFBEN	NFAEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NFAEN	Noise Filter A Enable	0: The noise filter for the MTIOCnA pin is disabled. 1: The noise filter for the MTIOCnA pin is enabled.	R/W
b1	NFBEN	Noise Filter B Enable	0: The noise filter for the MTIOCnB pin is disabled. 1: The noise filter for the MTIOCnB pin is enabled.	R/W
b2	NFCEN	Noise Filter C Enable*1	0: The noise filter for the MTIOCnC pin is disabled. 1: The noise filter for the MTIOCnC pin is enabled.	R/W
b3	NFDEN	Noise Filter D Enable*1	0: The noise filter for the MTIOCnD pin is disabled. 1: The noise filter for the MTIOCnD pin is enabled.	R/W
b5, b4	NFCS[1:0]	Noise Filter Clock Select	b5 b4 0 0: PCLKA/1 0 1: PCLKA/8 1 0: PCLKA/32 1 1: Clock source for counting	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits are reserved in MTU1 and MTU2. These bits are read as 0 and writing to them has no effect.

The NFCRn register (n = 0 to 4, 6, 7, 8) sets the noise filter function of input capture pins for the corresponding channel.

NFAEN Bit (Noise Filter A Enable)

This bit disables or enables the noise filter for input from the MTIOCnA pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFBEN Bit (Noise Filter B Enable)

This bit disables or enables the noise filter for input from the MTIOCnB pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFCEN Bit (Noise Filter C Enable)

This bit disables or enables the noise filter for input from the MTIOCnC pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFDEN Bit (Noise Filter D Enable)

This bit disables or enables the noise filter for input from the MTIOCnD pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFCS[1:0] Bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. When setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval before setting the input-capture function. When the NFCS[1:0] bits are set to 11b, i.e. selecting the external clock as the source to drive counting, wait for two cycles of the external clock before setting the input capture function.

- MTU0.NFCRC

Address(es): MTU0.NFCRC 000C 1299h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	NFCS[1:0]		NFDEN	NFCEN	NFBEN	NFAEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NFAEN	Noise Filter A Enable	0: The noise filter for the MTCLKA pin is disabled. 1: The noise filter for the MTCLKA pin is enabled.	R/W
b1	NFBEN	Noise Filter B Enable	0: The noise filter for the MTCLKB pin is disabled. 1: The noise filter for the MTCLKB pin is enabled.	R/W
b2	NFCEN	Noise Filter C Enable	0: The noise filter for the MTCLKC pin is disabled. 1: The noise filter for the MTCLKC pin is enabled.	R/W
b3	NFDEN	Noise Filter D Enable	0: The noise filter for the MTCLKD pin is disabled. 1: The noise filter for the MTCLKD pin is enabled.	R/W
b5, b4	NFCS[1:0]	Noise Filter Clock Select	b5 b4 0 0: PCLKA/1 0 1: PCLKA/2 1 0: PCLKA/8 1 1: PCLKA/32	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The NFCRC register sets the noise filter function of external clock pins common to each channel.

NFAEN Bit (Noise Filter A Enable)

This bit disables or enables the noise filter for input from the MTCLKA pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

NFBEN Bit (Noise Filter B Enable)

This bit disables or enables the noise filter for input from the MTCLKB pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

NFCEN Bit (Noise Filter C Enable)

This bit disables or enables the noise filter for input from the MTCLKC pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

NFDEN Bit (Noise Filter D Enable)

This bit disables or enables the noise filter for input from the MTCLKD pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

NFCS[1:0] Bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. After setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval to set the input capture function.

24.2.34 Noise Filter Control Register 5 (NFCR5)

Address(es): MTU5.NFCR5 000C 1A95h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	NFCS[1:0]		—	NFWEN	NFVEN	NFUEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NFUEN	Noise Filter U Enable	0: The noise filter for the MTIC5U pin is disabled. 1: The noise filter for the MTIC5U pin is enabled.	R/W
b1	NFVEN	Noise Filter V Enable	0: The noise filter for the MTIC5V pin is disabled. 1: The noise filter for the MTIC5V pin is enabled.	R/W
b2	NFWEN	Noise Filter W Enable	0: The noise filter for the MTIC5W pin is disabled. 1: The noise filter for the MTIC5W pin is enabled.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	NFCS[1:0]	Noise Filter Clock Select	b5 b4 0 0: PCLKA/1 0 1: PCLKA/8 1 0: PCLKA/32 1 1: Clock source for counting	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

NFUEN Bit (Noise Filter U Enable)

This bit disables or enables the noise filter for input from the MTIC5U pin. Since unexpected edges may be internally generated when the value of this bit is changed, select the output compare function for the relevant pin in the timer I/O control register.

NFVEN Bit (Noise Filter V Enable)

This bit disables or enables the noise filter for input from the MTIC5V pin. Since unexpected edges may be internally generated when the value of this bit is changed, select the output compare function for the relevant pin in the timer I/O control register.

NFWEN Bit (Noise Filter W Enable)

This bit disables or enables the noise filter for input from the MTIC5W pin. Since unexpected edges may be internally generated when the value of this bit is changed, select the output compare function for the relevant pin in the timer I/O control register.

NFCS[1:0] Bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. When setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval before setting the input-capture function.

24.2.35 Timer A/D Conversion Start Request Control Register (TADCR)

- MTU4.TADCR

Address(es): MTU4.TADCR 000C 1240h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BF[1:0]		—	—	—	—	—	—	UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ITB4VE	TCIV4 Interrupt Skipping Link Enable*1, *2, *3	0: A/D conversion start request TRG4BN and TCIV4 interrupt skipping 1 are not linked 1: A/D conversion start request TRG4BN and TCIV4 interrupt skipping 1 are linked	R/W
b1	ITB3AE	TGIA3 Interrupt Skipping Link Enable*1, *2, *3	0: A/D conversion start request TRG4BN and TGIA3 interrupt skipping 1 are not linked 1: A/D conversion start request TRG4BN and TGIA3 interrupt skipping 1 are linked	R/W
b2	ITA4VE	TCIV4 Interrupt Skipping Link Enable*1, *2, *3	0: A/D conversion start request TRG4AN and TCIV4 interrupt skipping 1 are not linked 1: A/D conversion start request TRG4AN and TCIV4 interrupt skipping 1 are linked	R/W
b3	ITA3AE	TGIA3 Interrupt Skipping Link Enable*1, *2, *3	0: A/D conversion start request TRG4AN and TGIA3 interrupt skipping 1 are not linked 1: A/D conversion start request TRG4AN and TGIA3 interrupt skipping 1 are linked	R/W
b4	DT4BE	Down-Count TRG4BN Enable*3	0: A/D conversion start requests (TRG4BN) disabled during MTU4.TCNT down-count operation 1: A/D conversion start requests (TRG4BN) enabled during MTU4.TCNT down-count operation	R/W
b5	UT4BE	Up-Count TRG4BN Enable	0: A/D conversion start requests (TRG4BN) disabled during MTU4.TCNT up-count operation 1: A/D conversion start requests (TRG4BN) enabled during MTU4.TCNT up-count operation	R/W
b6	DT4AE	Down-Count TRG4AN Enable*3	0: A/D conversion start requests (TRG4AN) disabled during MTU4.TCNT down-count operation 1: A/D conversion start requests (TRG4AN) enabled during MTU4.TCNT down-count operation	R/W
b7	UT4AE	Up-Count TRG4AN Enable	0: A/D conversion start requests (TRG4AN) disabled during MTU4.TCNT up-count operation 1: A/D conversion 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/TADCOBRB Transfer Timing Select	Refer to Table 24.53 for details. These bits specify the transfer timing from MTU4.TADCOBRA and MTU4.TADCOBRB to MTU4.TADCORA and MTU4.TADCORB.	R/W

Note: MTU4.TADCR must not be accessed in 8 bits; it should be accessed in 16 bits.
Note 1. Set to 0 when interrupt skipping is disabled (the T3AEN and T4VEN bits in TITCR1A are set to 0 or the T3ACOR and T4VCOR bits in TITCR1A are set to 0).
Note 2. If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D conversion start requests will not be issued.
Note 3. Set to 0 when complementary PWM mode is not selected.

TADCR enables or disables A/D conversion start requests and specifies whether to link A/D conversion start requests with interrupt skipping function. The MTU has one TADCR each for MTU4 and MTU7.

Table 24.53 Setting of Transfer Timing by TADCR.BF[1:0] Bits (MTU4)

Bit 15	Bit 14	Description			
BF[1]	BF[0]	In Complementary PWM Mode	In Reset-Synchronized PWM Mode	In PWM Mode 1	In Normal Mode
0	0	Data is not transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).	Data is not transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).	Data is not transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).	Data is not transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).
0	1	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) at the crest of the MTU4.TCNT.	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) when a compare match occurs between MTU3.TCNT and MTU3.TGRA.	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) when a compare match occurs between MTU4.TCNT and MTU4.TGRA.	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) when a compare match occurs between MTU4.TCNT and MTU4.TGRA.
1	0	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) at the trough of the MTU4.TCNT.	Setting prohibited	Setting prohibited	Setting prohibited
1	1	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) at the crest and trough of the MTU4.TCNT.	Setting prohibited	Setting prohibited	Setting prohibited

- MTU7.TADCR

Address(es): MTU7.TADCR 000C 1A40h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BF[1:0]		—	—	—	—	—	—	UT7AE	DT7AE	UT7BE	DT7BE	ITA6AE	ITA7VE	ITB6AE	ITB7VE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ITB7VE	TCIV7 Interrupt Skipping Link Enable*1, *2, *3	0: A/D conversion start request TRG7BN and TCIV7 interrupt skipping 1 are not linked 1: A/D conversion start request TRG7BN and TCIV7 interrupt skipping 1 are linked	R/W
b1	ITB6AE	TGIA6 Interrupt Skipping Link Enable*1, *2, *3	0: A/D conversion start request TRG7BN and TGIA6 interrupt skipping 1 are not linked 1: A/D conversion start request TRG7BN and TGIA6 interrupt skipping 1 are linked	R/W
b2	ITA7VE	TCIV7 Interrupt Skipping Link Enable*1, *2, *3	0: A/D conversion start request TRG7AN and TCIV7 interrupt skipping 1 are not linked 1: A/D conversion start request TRG7AN and TCIV7 interrupt skipping 1 are linked	R/W
b3	ITA6AE	TGIA6 Interrupt Skipping Link Enable*1, *2, *3	0: A/D conversion start request TRG7AN and TGIA6 interrupt skipping 1 are not linked 1: A/D conversion start request TRG7AN and TGIA6 interrupt skipping 1 are linked	R/W
b4	DT7BE	Down-Count TRG7BN Enable*3	0: A/D conversion start requests (TRG7BN) disabled during MTU7.TCNT down-count operation 1: A/D conversion start requests (TRG7BN) enabled during MTU7.TCNT down-count operation	R/W
b5	UT7BE	Up-Count TRG7BN Enable	0: A/D conversion start requests (TRG7BN) disabled during MTU7.TCNT up-count operation 1: A/D conversion start requests (TRG7BN) enabled during MTU7.TCNT up-count operation	R/W
b6	DT7AE	Down-Count TRG7AN Enable*3	0: A/D conversion start requests (TRG7AN) disabled during MTU7.TCNT down-count operation 1: A/D conversion start requests (TRG7AN) enabled during MTU7.TCNT down-count operation	R/W
b7	UT7AE	Up-Count TRG7AN Enable	0: A/D conversion start requests (TRG7AN) disabled during MTU7.TCNT up-count operation 1: A/D conversion start requests (TRG7AN) enabled during MTU7.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]	MTU7.TADCOBRA/TADCOBRB Transfer Timing Select	Refer to Table 24.54 for details. These bits specify the transfer timing from MTU7.TADCOBRA and MTU7.TADCOBRB to MTU7.TADCORA and MTU7.TADCORB.	R/W

Note: MTU7.TADCR must not be accessed in 8 bits; it should be accessed in 16 bits.
Note 1. Set to 0 when interrupt skipping is disabled (the T6AEN and T7VEN bits in TITCR1B are set to 0 or the T6ACOR and T7VCOR bits in TITCR1B are set to 0).
Note 2. If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D conversion start requests will not be issued.
Note 3. Set to 0 when complementary PWM mode is not selected.

Table 24.54 Setting of Transfer Timing by TADCR.BF[1:0] Bits (MTU7)

Bit 15	Bit 14	Description			
BF[1]	**BF[0]**	**In Complementary PWM Mode**	**In Reset-Synchronized PWM Mode**	**In PWM Mode 1**	**In Normal Mode**
0	0	Data is not transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB).	Data is not transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB).	Data is not transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB).	Data is not transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB).
0	1	Data is transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB) at the crest of the MTU7.TCNT.	Data is transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB) when a compare match occurs between MTU6.TCNT and MTU6.TGRA.	Data is transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB) when a compare match occurs between MTU7.TCNT and MTU7.TGRA.	Data is transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB) when a compare match occurs between MTU7.TCNT and MTU7.TGRA.
1	0	Data is transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB) at the trough of the MTU7.TCNT.	Setting prohibited	Setting prohibited	Setting prohibited
1	1	Data is transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB) at the crest and trough of the MTU7.TCNT.	Setting prohibited	Setting prohibited	Setting prohibited

24.2.36 Timer A/D Conversion Start Request Cycle Set Register m (TADCORm) (m = A, B)

Address(es): MTU4.TADCORA 000C 1244h, MTU4.TADCORB 000C 1246h, MTU7.TADCORA 000C 1A44h, MTU7.TADCORB 000C 1A46h

	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: TADCORA and TADCORB must not be accessed in 8 bits; it should be accessed in 16 bits.

Note 1. When the A/D conversion start request delaying function linked with skipping function 1 (for details, refer to section 24.3.9 (5), A/D Conversion Start Request Delaying Function Linked with Interrupt Skipping Function 1) is used, the value of this register should be 0002h to TCDRA setting – 2 in MTU4 and 0002h to TCDRB setting – 2 in MTU7.

Note 2. When interrupt skipping function 2 is used and the difference between the TADCORA value and the TADCORB value is small, the skipping count may not be counted correctly and the A/D conversion start request may not be generated with the expected timing in some cases. The TADCORA and TADCORB values should satisfy the following conditions.

(1) When skipping function 2 is specified with the skipping count set to 0
- The difference between the TADCORA and TADCORB values should be equal to or greater than 4.
- The TADCORA compare interval should be equal to or greater than 4 PCLKA cycles (the TADCORA update value should be the previous value + 4 or greater, or previous value – 4 or smaller).
- The TADCORB compare interval should be equal to or greater than 4 PCLKA cycles (the TADCORB update value should be the previous value + 4 or greater, or previous value – 4 or smaller).

(2) When skipping function 2 is specified with the skipping count set to 1 or greater
- The difference between the TADCORA and TADCORB values should be equal to or greater than 2.
- The TADCORB compare interval should be equal to or greater than 2 PCLKA cycles (the TADCORB update value should be the previous value + 2 or greater, or previous value – 2 or smaller)

TADCORA and TADCORB are 16-bit readable/writable registers that issue a corresponding A/D conversion start request when the MTUn.TCNT (n = 4, 7) count reaches the value in TADCORA or TADCORB.
MTUn.TADCORA and TADCORB are initialized to FFFFh by a reset.

24.2.37 Timer A/D Conversion Start Request Cycle Set Buffer Register m (TADCOBRm) (m = A, B)

Address(es): MTU4.TADCOBRA 000C 1248h, MTU4.TADCOBRB 000C 124Ah, MTU7.TADCOBRA 000C 1A48h, MTU7.TADCOBRB 000C 1A4Ah

	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: TADCOBRA and TADCOBRB must not be accessed in 8 bits; it should be accessed in 16 bits.

TADCOBRA and TADCOBRB are 16-bit readable/writable registers whose values are transferred to TADCORA and TADCORB, respectively, when the crest or trough of the MTUn.TCNT count is reached.
TADCOBRA and TADCOBRB are initialized to FFFFh by a reset.

24.2.38 Timer Interrupt Skipping Mode Register m (TITMRm) (m = A, B)

Address(es): MTU.TITMRA 000C 123Ah, MTU.TITMRB 000C 1A3Ah

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

Bit	Symbol	Bit Name	Description	R/W
b0	TITM	Interrupt Skipping Function Select	Selects one of the two types of interrupt skipping functions. 0: Selects interrupt skipping function 1*1 1: Selects interrupt skipping function 2*2	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Setting the TITCR1A or TITCR1B register enables interrupt skipping function 1.
Note 2. Setting the TITCR2A or TITCR2B register enables interrupt skipping function 2.

TITMRA and TITMRB are used to select either of two skipping functions for the TITMRA and TITMRB registers.

24.2.39 Timer Interrupt Skipping Set Register 1m (TITCR1m) (m = A, B)

- MTU.TITCR1A

Address(es): MTU.TITCR1A 000C 1230h

	b7	b6	b5	b4	b3	b2	b1	b0
	T3AEN	T3ACOR[2:0]			T4VEN	T4VCOR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	T4VCOR[2:0]	TCIV4 Interrupt Skipping Count Setting	These bits specify the TCIV4 interrupt skipping count within the range from 0 to 7.*1 For details, refer to Table 24.55.	R/W
b3	T4VEN	T4VEN	0: TCIV4 interrupt skipping disabled 1: TCIV4 interrupt skipping enabled	R/W
b6 to b4	T3ACOR[2:0]	TGIA3 Interrupt Skipping Count Setting	These bits specify the TGIA3 interrupt skipping count within the range from 0 to 7.*1 For details, refer to Table 24.56.	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 set the TITCR1A.T3AEN and TITCR1A.T4VEN bits to 0 to clear the skipping counter (TITCNT1A).

- MTU.TITCR1B

Address(es): MTU.TITCR1B 000C 1A30h

	b7	b6	b5	b4	b3	b2	b1	b0
	T6AEN	T6ACOR[2:0]			T7VEN	T7VCOR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	T7VCOR[2:0]	TCIV7 Interrupt Skipping Count Setting	These bits specify the TCIV7 interrupt skipping count within the range from 0 to 7.*1 For details, refer to Table 24.57.	R/W
b3	T7VEN	T7VEN	0: TCIV7 interrupt skipping disabled 1: TCIV7 interrupt skipping enabled	R/W
b6 to b4	T6ACOR[2:0]	TGIA6 Interrupt Skipping Count Setting	These bits specify the TGIA6 interrupt skipping count within the range from 0 to 7.*1 For details, refer to Table 24.58.	R/W
b7	T6AEN	T6AEN	0: TGIA6 interrupt skipping disabled 1: TGIA6 interrupt skipping enabled	R/W

Note 1. When 0 is specified for the interrupt skipping count, no interrupt skipping will be performed.
Before changing the interrupt skipping count, be sure to set the TITCR1B.T6AEN and TITCR1B.T7VEN bits to 0 to clear the skipping counter (TITCNT1B).

Registers TITCR1A and TITCR1B enable or disable interrupt skipping and specify the interrupt skipping count. This setting is valid only while the TITMRA.TITM or TITMRB.TITM bit is set to 0; when the TITMRA.TITM (TITMRB.TITM) bit is set to 1, the setting in the TITCR1A (TITCR1B) register is cleared.

Table 24.55 Setting of Interrupt Skipping Count by T4VCOR[2:0] Bits

Bit 2	Bit 1	Bit 0	
T4VCOR[2]	T4VCOR[1]	T4VCOR[0]	Description
0	0	0	Does not skip TCIV4 interrupts.
0	0	1	Sets the TCIV4 interrupt skipping count to 1.
0	1	0	Sets the TCIV4 interrupt skipping count to 2.
0	1	1	Sets the TCIV4 interrupt skipping count to 3.
1	0	0	Sets the TCIV4 interrupt skipping count to 4.
1	0	1	Sets the TCIV4 interrupt skipping count to 5.
1	1	0	Sets the TCIV4 interrupt skipping count to 6.
1	1	1	Sets the TCIV4 interrupt skipping count to 7.

Table 24.56 Setting of Interrupt Skipping Count by T3ACOR[2:0] Bits

Bit 6	Bit 5	Bit 4	
T3ACOR[2]	T3ACOR[1]	T3ACOR[0]	Description
0	0	0	Does not skip TGIA3 interrupts.
0	0	1	Sets the TGIA3 interrupt skipping count to 1.
0	1	0	Sets the TGIA3 interrupt skipping count to 2.
0	1	1	Sets the TGIA3 interrupt skipping count to 3.
1	0	0	Sets the TGIA3 interrupt skipping count to 4.
1	0	1	Sets the TGIA3 interrupt skipping count to 5.
1	1	0	Sets the TGIA3 interrupt skipping count to 6.
1	1	1	Sets the TGIA3 interrupt skipping count to 7.

Table 24.57 Setting of Interrupt Skipping Count by T7VCOR[2:0] Bits

Bit 2	Bit 1	Bit 0	
T7VCOR[2]	T7VCOR[1]	T7VCOR[0]	Description
0	0	0	Does not skip TCIV7 interrupts.
0	0	1	Sets the TCIV7 interrupt skipping count to 1.
0	1	0	Sets the TCIV7 interrupt skipping count to 2.
0	1	1	Sets the TCIV7 interrupt skipping count to 3.
1	0	0	Sets the TCIV7 interrupt skipping count to 4.
1	0	1	Sets the TCIV7 interrupt skipping count to 5.
1	1	0	Sets the TCIV7 interrupt skipping count to 6.
1	1	1	Sets the TCIV7 interrupt skipping count to 7.

Table 24.58 Setting of Interrupt Skipping Count by T6ACOR[2:0] Bits

Bit 6	Bit 5	Bit 4	
T6ACOR[2]	T6ACOR[1]	T6ACOR[0]	Description
0	0	0	Does not skip TGIA6 interrupts.
0	0	1	Sets the TGIA6 interrupt skipping count to 1.
0	1	0	Sets the TGIA6 interrupt skipping count to 2.
0	1	1	Sets the TGIA6 interrupt skipping count to 3.
1	0	0	Sets the TGIA6 interrupt skipping count to 4.
1	0	1	Sets the TGIA6 interrupt skipping count to 5.
1	1	0	Sets the TGIA6 interrupt skipping count to 6.
1	1	1	Sets the TGIA6 interrupt skipping count to 7.

24.2.40 Timer Interrupt Skipping Counter 1m (TITCNT1m) (m = A, B)

- MTU.TITCNT1A

Address(es): MTU.TITCNT1A 000C 1231h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	T3ACNT[2:0]			—	T4VCNT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	T4VCNT[2:0]	TCIV4 Interrupt Counter	While the T4VEN bit in TITCR1A is set to 1, the count in these bits is incremented every time a TCIV4 interrupt occurs.	R
b3	—	Reserved	This bit is read as 0.	R
b6 to b4	T3ACNT[2:0]	TGIA3 Interrupt Counter	While the T3AEN bit in TITCR1A is set to 1, the count in these bits is incremented every time a TGIA3 interrupt occurs.	R
b7	—	Reserved	This bit is read as 0.	R

Note: To clear the TITCNT1A, set the TITCR1A.T3AEN and TITCR1A.T4VEN bits to 0.

TITCNT1A and TITCNT1B are 8-bit readable/writable counters. TITCNT1A and TITCNT1B retain their values even after stopping the count operation of MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT).

T4VCNT[2:0] Bits (TCIV4 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRA is 1
- When the T4VEN bit in TITCR1A is set to 0
- When the T4VCOR[2:0] bits in TITCR1A are set to 000b
- When the T4VCNT[2:0] bits in TITCNT1A match the T4VCOR[2:0] bits in TITCR1A

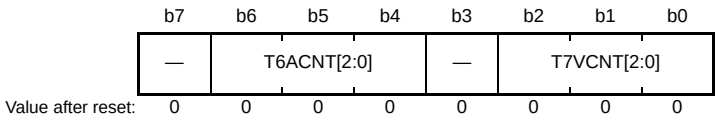
T3ACNT[2:0] Bits (TGIA3 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRA is 1
- When the T3AEN bit in TITCR1A is set to 0
- When the T3ACOR[2:0] bits in TITCR1A are set to 000b
- When the T3ACNT[2:0] bits in TITCNT1A match the T3ACOR[2:0] bits in TITCR1A

- MTU.TITCNT1B

Address(es): MTU.TITCNT1B 000C 1A31h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	T7VCNT[2:0]	TCIV7 Interrupt Counter	While the T7VEN bit in TITCR1B is set to 1, the count in these bits is incremented every time a TCIV7 interrupt occurs.	R
b3	—	Reserved	This bit is read as 0.	R
b6 to b4	T6ACNT[2:0]	TGIA6 Interrupt Counter	While the T6AEN bit in TITCR1B is set to 1, the count in these bits is incremented every time a TGIA6 interrupt occurs.	R
b7	—	Reserved	This bit is read as 0.	R

Note: To clear the TITCNT1B, set the TITCR1B.T6AEN and TITCR1B.T7VEN bits to 0.

T7VCNT[2:0] Bits (TCIV7 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRB is 1
- When the T7VEN bit in TITCR1B is set to 0
- When the T7VCOR[2:0] bits in TITCR1B are set to 000b
- When the T7VCNT[2:0] bits in TITCNT1B match the T7VCOR[2:0] bits in TITCR1B

T6ACNT[2:0] Bits (TGIA6 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRB is 1
- When the T6AEN bit in TITCR1B is set to 0
- When the T6ACOR[2:0] bits in TITCR1B are set to 000b
- When the T6ACNT[2:0] bits in TITCNT1B match the T6ACOR[2:0] bits in TITCR1B

24.2.41 Timer Interrupt Skipping Set Register 2m (TITCR2m) (m = A, B)

- MTU.TITCR2A

Address(es): MTU.TITCR2A 000C 123Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TRG4COR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TRG4COR[2:0]	TRG4AN/TRG4BN Interrupt Skipping Count Setting	These bits specify the TRG4AN/TRG4BN interrupt skipping count within the range from 0 to 7. For details, refer to Table 24.59.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TITCR2A and TITCR2B specify the interrupt skipping count for TRG4AN and TRG4BN (TRG7AN and TRG7BN). This setting is valid only while TITMRA or TITMRB is set to 1.

Table 24.59 Setting of Interrupt Skipping Count by TRG4COR[2:0] Bits

Bit 2	Bit 1	Bit 0	
TRG4COR[2]	TRG4COR[1]	TRG4COR[0]	Description
0	0	0	Does not skip TRG4AN and TRG4BN interrupts.
0	0	1	Sets the TRG4AN and TRG4BN interrupt skipping count to 1.
0	1	0	Sets the TRG4AN and TRG4BN interrupt skipping count to 2.
0	1	1	Sets the TRG4AN and TRG4BN interrupt skipping count to 3.
1	0	0	Sets the TRG4AN and TRG4BN interrupt skipping count to 4.
1	0	1	Sets the TRG4AN and TRG4BN interrupt skipping count to 5.
1	1	0	Sets the TRG4AN and TRG4BN interrupt skipping count to 6.
1	1	1	Sets the TRG4AN and TRG4BN interrupt skipping count to 7.

- MTU.TITCR2B

Address(es): MTU.TITCR2B 000C 1A3Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TRG7COR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TRG7COR[2:0]	TRG7AN/TRG7BN Interrupt Skipping Count Setting	These bits specify the TRG7AN/TRG7BN interrupt skipping count within the range from 0 to 7. For details, refer to Table 24.60.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

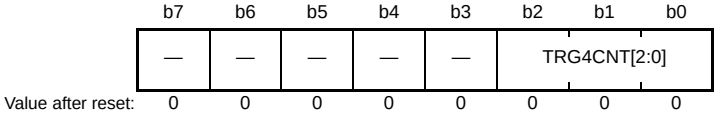
Table 24.60 Setting of Interrupt Skipping Count by TRG7COR[2:0] Bits

Bit 2	Bit 1	Bit 0	
TRG7COR[2]	**TRG7COR[1]**	**TRG7COR[0]**	**Description**
0	0	0	Does not skip TRG7AN and TRG7BN interrupts.
0	0	1	Sets the TRG7AN and TRG7BN interrupt skipping count to 1.
0	1	0	Sets the TRG7AN and TRG7BN interrupt skipping count to 2.
0	1	1	Sets the TRG7AN and TRG7BN interrupt skipping count to 3.
1	0	0	Sets the TRG7AN and TRG7BN interrupt skipping count to 4.
1	0	1	Sets the TRG7AN and TRG7BN interrupt skipping count to 5.
1	1	0	Sets the TRG7AN and TRG7BN interrupt skipping count to 6.
1	1	1	Sets the TRG7AN and TRG7BN interrupt skipping count to 7.

24.2.42 Timer Interrupt Skipping Counter 2m (TITCNT2m) (m = A, B)

- MTU.TITCNT2A

Address(es): MTU.TITCNT2A 000C 123Ch



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TRG4CNT[2:0]	TRG4AN/TRG4BN Interrupt Counter	These bits start counting from the value set in TRG4COR[2:0] and the count decrements every time TRG4AN or TRG4BN is generated. When the count reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts become valid.	R
b7 to b3	—	Reserved	These bits are read as 0.	R

TITCNT2A and TITCNT2B start counting from the values set in the TRG4COR[2:0] and TRG7COR[2:0] bits and the count decrements every time TRG4AN or TRG4BN (TITCNT2A) is generated or TRG7AN or TRG7BN (TITCNT2B) is generated. When the count reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts or the TRG7AN and TRG7BN interrupts become valid.

TRG4CNT[2:0] Bits (TRG4AN/TRG4BN Interrupt Counter)

These bits start counting from the value set in the TRG4COR[2:0] bits and the count decrements every time a TRG4AN or TRG4BN interrupt is generated. When the count reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts become valid.

[Clearing conditions]

- When the TITM bit in TITMRA is 0
- When the TRG4COR[2:0] bits in TITCR2A are set to 000b
- When the count of TRG4AN and TRG4BN occurrence matches the TRG4COR[2:0] value in TITCR2A

- MTU.TITCNT2B

Address(es): MTU.TITCNT2B 000C 1A3Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TRG7CNT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TRG7CNT[2:0]	TRG7AN/TRG7BN Interrupt Counter	These bits start counting from the value set in TRG7COR[2:0] and the count decrements every time TRG7AN or TRG7BN is generated. When the count reaches 0 and is reloaded, the TRG7AN and TRG7BN interrupts become valid.	R
b7 to b3	—	Reserved	These bits are read as 0.	R

TRG7CNT[2:0] Bits (TRG7AN/TRG7BN Interrupt Counter)

These bits start counting from the value set in the TRG7COR[2:0] bits and the count decrements every time a TRG7AN or TRG7BN interrupt is generated. When the count reaches 0 and is reloaded, the TRG7AN and TRG7BN interrupts become valid.

[Clearing conditions]

- When the TITM bit in TITMRB is 0
- When the TRG7COR[2:0] bits in TITCR2B are set to 000b
- When the count of TRG7AN and TRG7BN occurrence matches the TRG7COR[2:0] value in TITCR2B

24.3 Operation

24.3.1 Basic Functions

Each channel has TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, periodic counting, and external event counting.
Each TGR register can be used as an input capture register or an output compare register.

(1) Counter Operation

When one of bits CST0 to CST4 and CST8 in the TSTRA register, bits CST6 and CST7 in the TSTRB register, and bits CSTU5, CSTV5, and CSTW5 in the MTU5.TSTR register is set to 1, TCNT for the corresponding channel begins counting. TCNT can operate as a free-running counter, periodic counter, for example.

(a) Example of Count Operation Setting Procedure

Figure 24.5 shows an example of the count operation setting procedure.

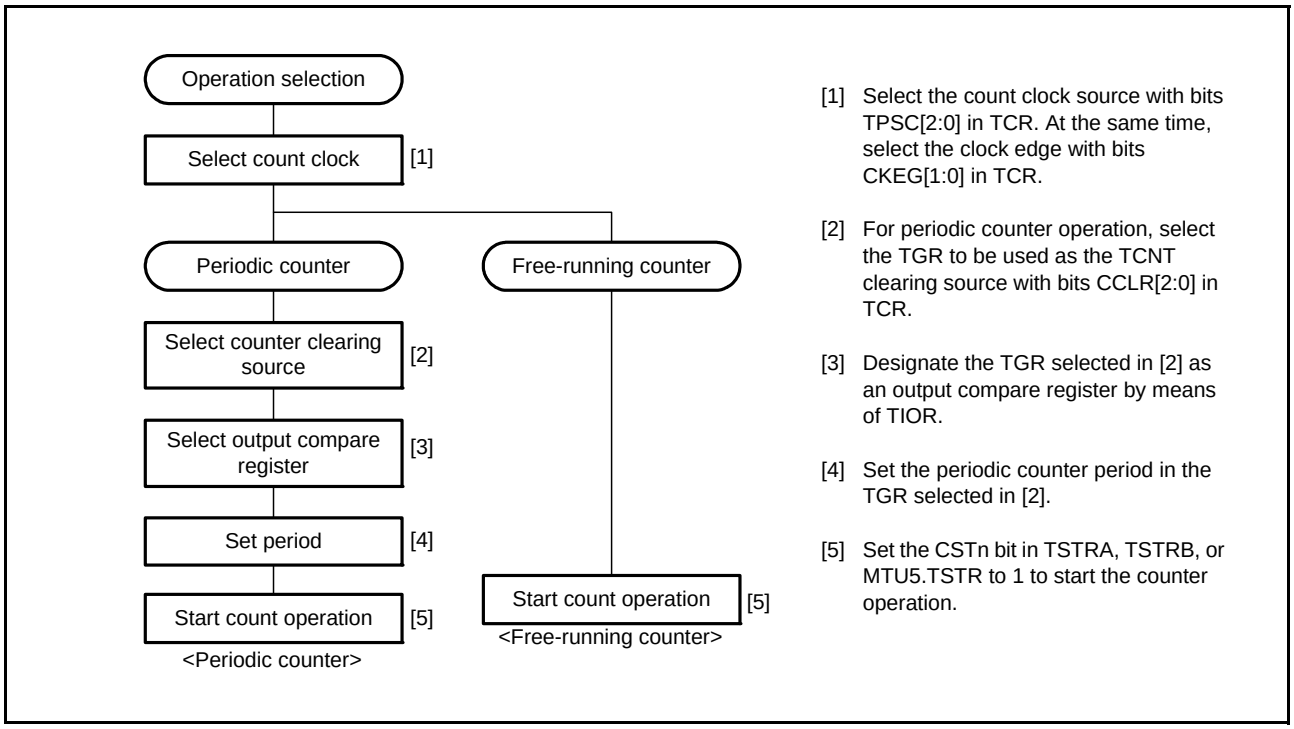


Figure 24.5 Example of Count Operation Setting Procedure

(b) Free-Running Count Operation and Periodic Count Operation

Immediately after a reset, the TCNT counters are all designated as free-running counters. When the CSTn bit in TSTRA, TSTRB, or MTU5.TSTR is set to 1, the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (from FFFFh to 0000h), an interrupt request is issued to the CPU if the corresponding TIER.TCIEV bit is 1. After an overflow, TCNT starts counting up again from 0000h.
Figure 24.6 illustrates free-running counter operation.

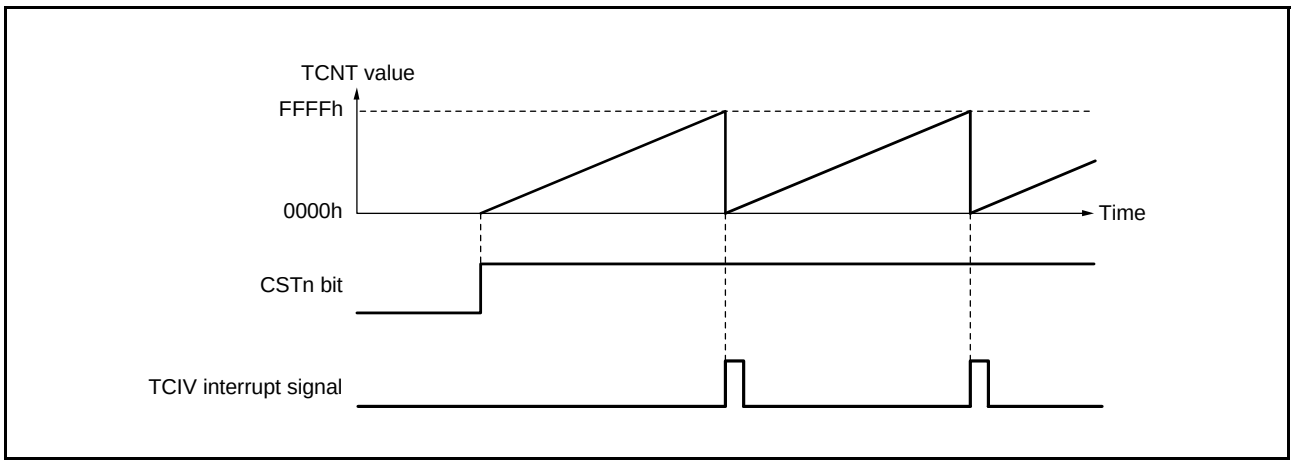


Figure 24.6 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, TCNT for the relevant channel performs periodic count operation. TGR for setting the period 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 CSTn bit in TSTRA, TSTRB or MTU5.TSTR is set to 1. When the count matches the value in TGR, TCNT becomes 0000h.
If the value of the corresponding TIER.TGIE bit is 1 at this point, an interrupt request is issued to the CPU. After a compare match, TCNT starts counting up again from 0000h.
Figure 24.7 illustrates periodic counter operation.

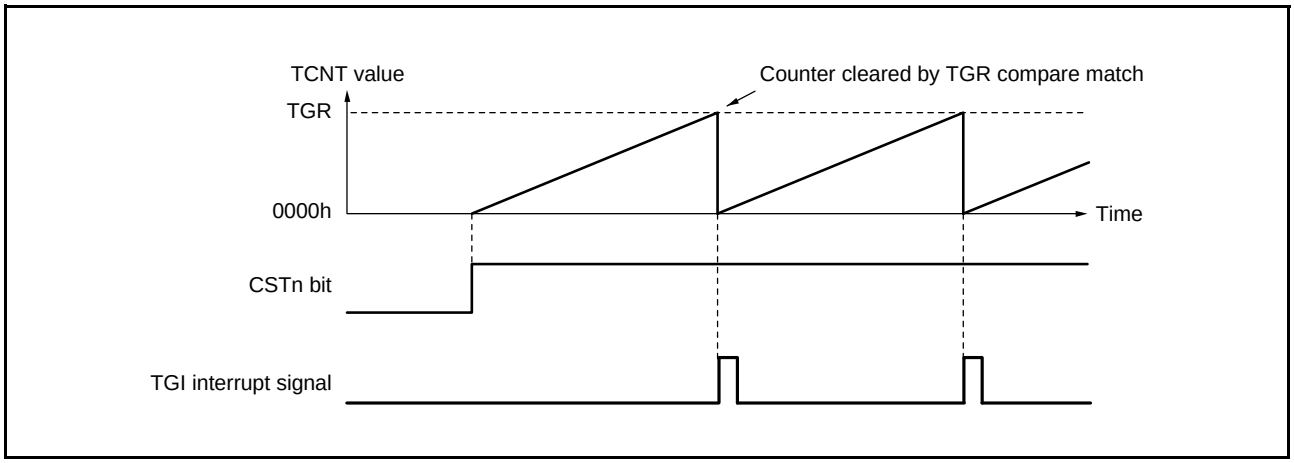


Figure 24.7 Periodic Counter Operation

(2) Waveform Output by Compare Match

Upon compare match, low, high, or toggle output from the corresponding pin can be performed.

(a) Example of Procedure for Setting Waveform Output by Compare Match

Figure 24.8 shows an example of the procedure for setting waveform output by compare match

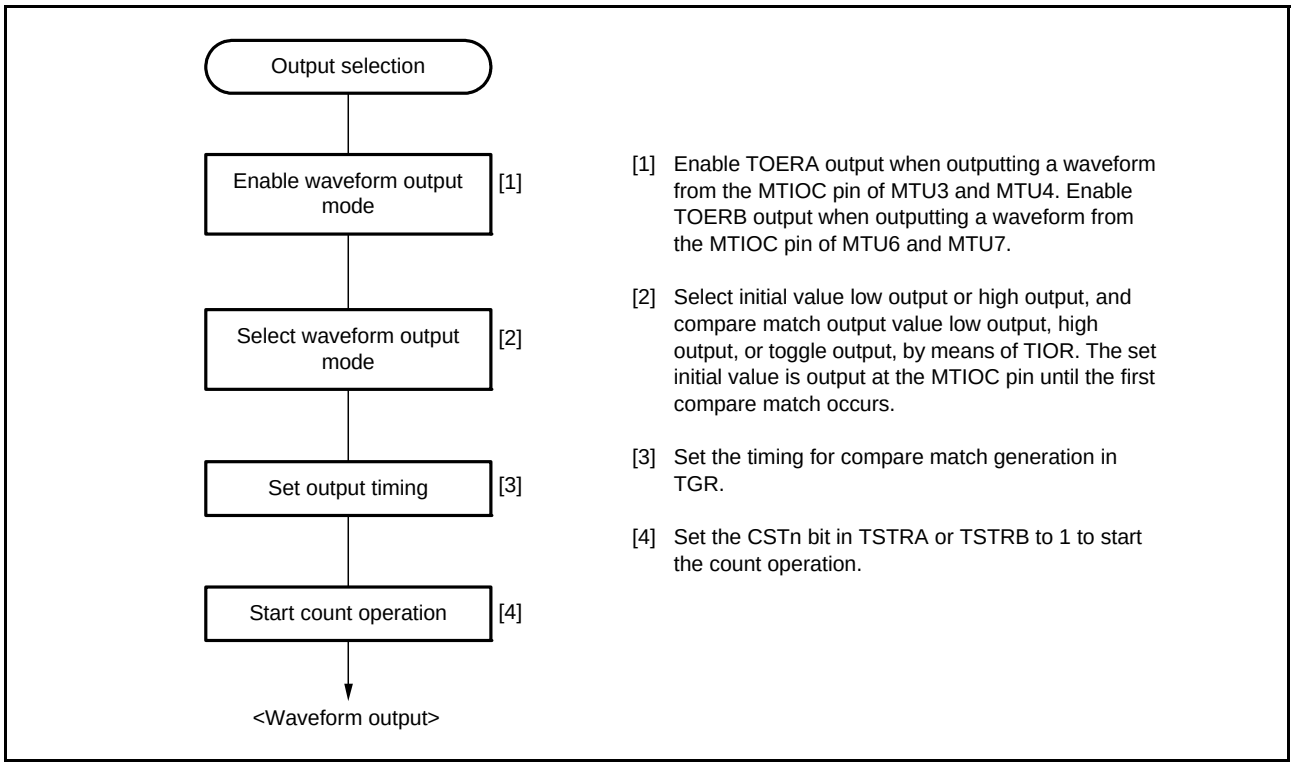


Figure 24.8 Example of Procedure for Setting Waveform Output by Compare Match

(b) Examples of Waveform Output Operation

Figure 24.9 shows an example of low output and high output.
In this example, TCNT has been designated as a free-running counter, and settings have been made so that high is output by compare match A and low is output by compare match B. When the pin level is the same as the specified level, the pin level does not change.

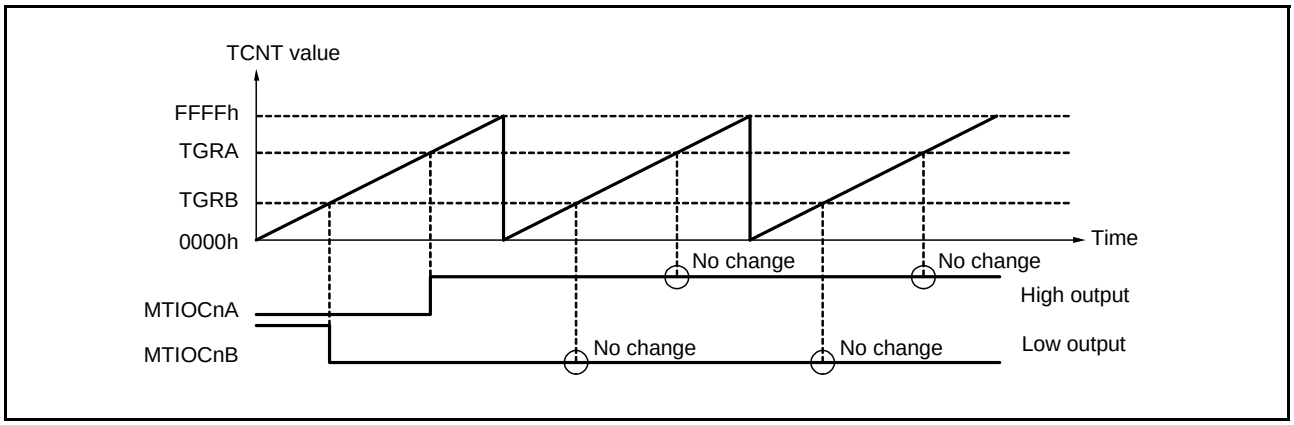


Figure 24.9 Example of low output and high output Operation (n = 0 to 4, 6, 7, 8)

Figure 24.10 shows an example of toggle output.
In this example, TCNT has been designated as a periodic counter (with counter clearing on compare match B), and settings have been made so that the output is toggled by both compare match A and compare match B.

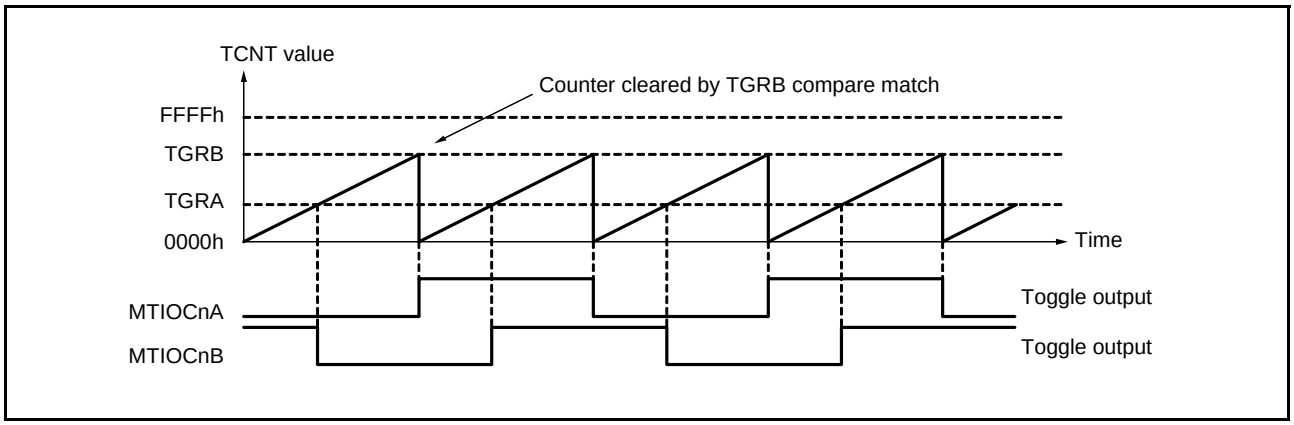


Figure 24.10 Example of Toggle Output Operation (n = 0 to 4, 6, 7, 8)

(3) Input Capture Function

The TCNT value can be transferred to TGR on detection of the MTIOCnm pin (n = 0 to 4, 6, 7, 8; m = A to D) or MTIC5m pin (m = U, V, W) input edge.
The rising edge, falling edge, or both edges can be selected as the detection edge. For MTU0 and MTU1, another channel's count clock or compare match signal can also be specified as the input capture source.

Note: When another channel's count clock is used as the input capture input for MTU0 and MTU1, PCLKA/1 should not be selected as the count clock used for input capture input. Input capture will not be generated if PCLKA/1 is selected.

(a) Example of Input Capture Operation Setting Procedure

Figure 24.11 shows an example of the input capture operation setting procedure.

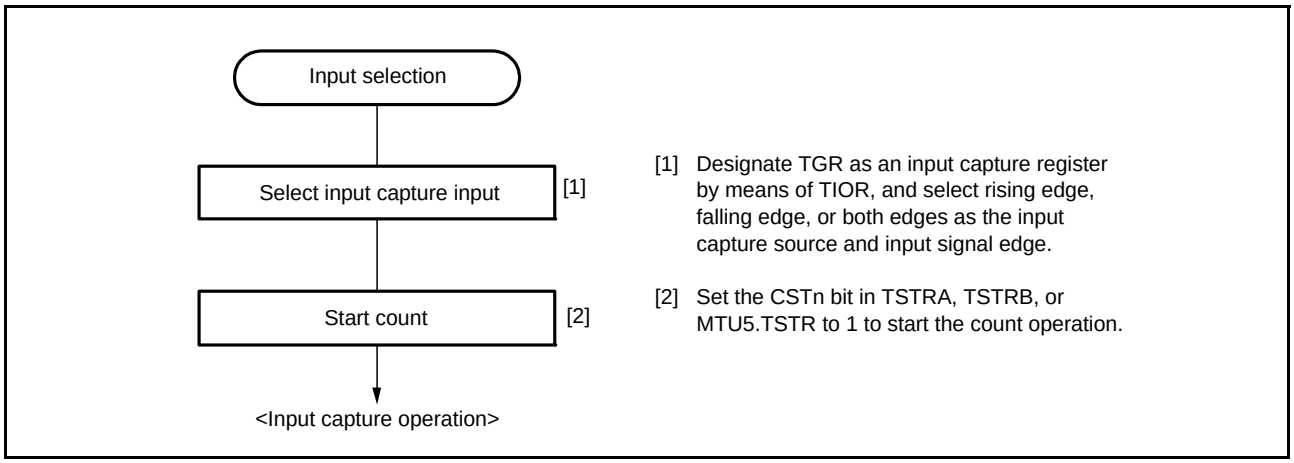


Figure 24.11 Example of Input Capture Operation Setting Procedure

(b) Example of Input Capture Operation

Figure 24.12 shows an example of input capture operation.
In this example, both rising and falling edges have been selected as the MTIOCnA pin input capture input edge, the falling edge has been selected as the MTIOCnB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT. (n = 0 to 4, 6, 7, 8)

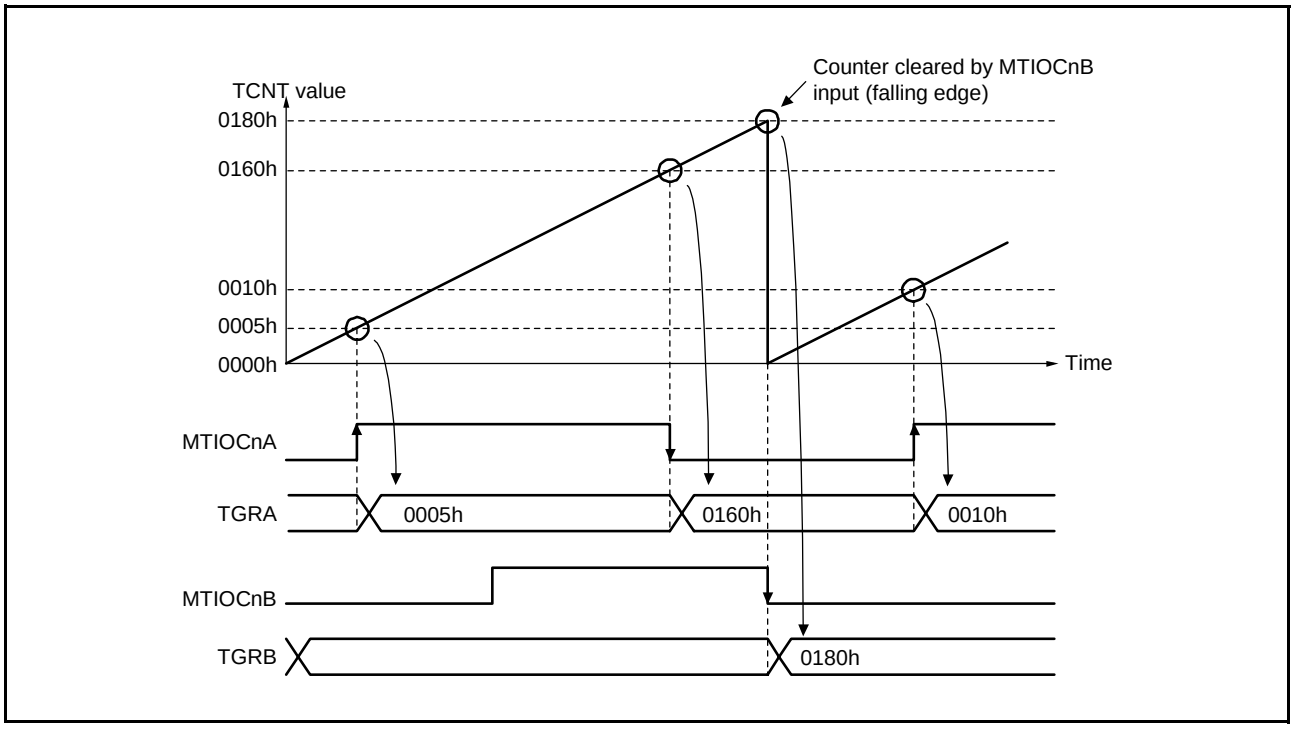


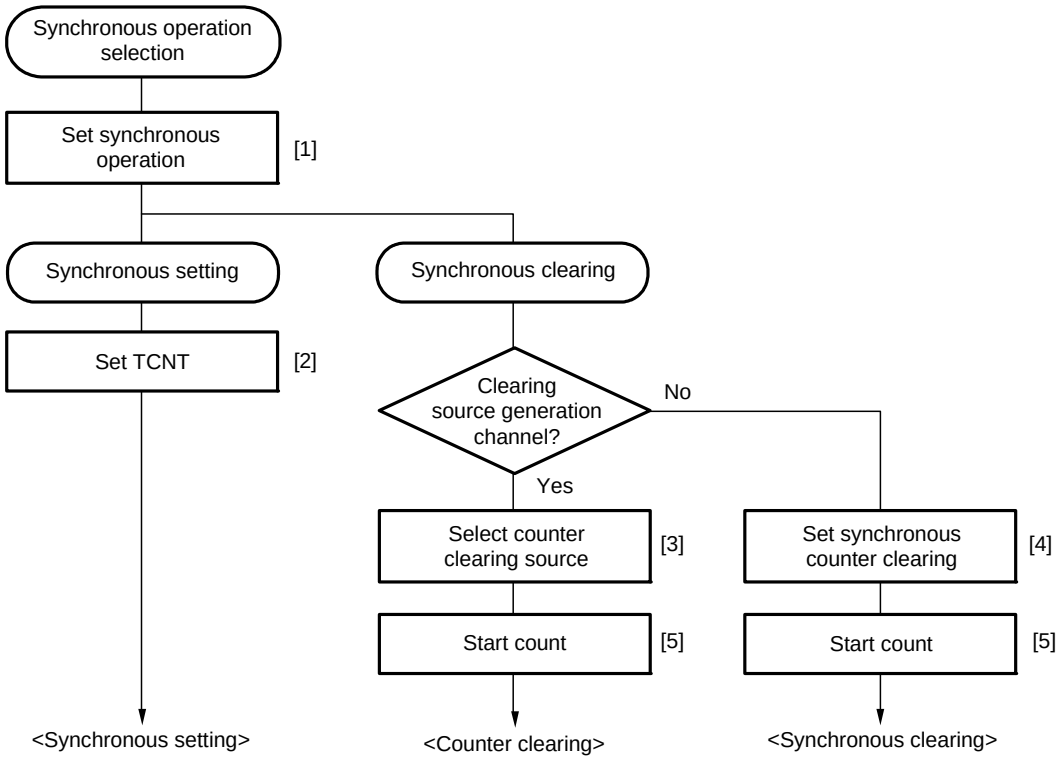
Figure 24.12 Example of Input Capture Operation (n = 0 to 4, 6, 7, 8)

24.3.2 Synchronous Operation

In synchronous operation, the values in multiple TCNT counters can be modified simultaneously (synchronous setting). In addition, multiple TCNT counters can be cleared simultaneously (synchronous clearing) by making the appropriate setting in TCR.
Synchronous operation can increase the number of TGR registers assigned to the same time base.
MTU0 to MTU4, MTU6, and MTU7 can all be designated for synchronous operation. MTU5 and MTU8 cannot be used for synchronous operation.

(1) Example of Synchronous Operation Setting Procedure

Figure 24.13 shows an example of the synchronous operation setting procedure.



[1] Set to 1 in the SYNC bits in TSYRA or TSYRB corresponding to the channels to be designated for synchronous operation.
[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.
[3] Use the CCLR[2:0] bits in TCR to specify TCNT clearing by input capture/output compare, etc.
[4] Use the CCLR[2:0] bits in TCR to designate synchronous clearing for the counter clearing source.
[5] Set to 1 in the CSTn bits in TSTRA or TSTRB for the relevant channels, to start the count operation.

Figure 24.13 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 24.14 shows an example of synchronous operation.
In this example, synchronous operation and PWM mode 1 have been designated for MTU0 to MTU 2, MTU0.TGRB compare match has been set as the counter clearing source in MTU0, and synchronous clearing has been set for the counter clearing source in MTU1 and MTU2.
Three-phase PWM waveforms are output from pins MTIOC0A, MTIOC1A, and MTIOC2A. At this time, synchronous setting and synchronous clearing by MTU0.TGRB compare match are performed for the TCNT counters in MTU0 to MTU2, and the data set in MTU0.TGRB is used as the PWM period.
For details of PWM modes, refer to section 24.3.5, PWM Modes.

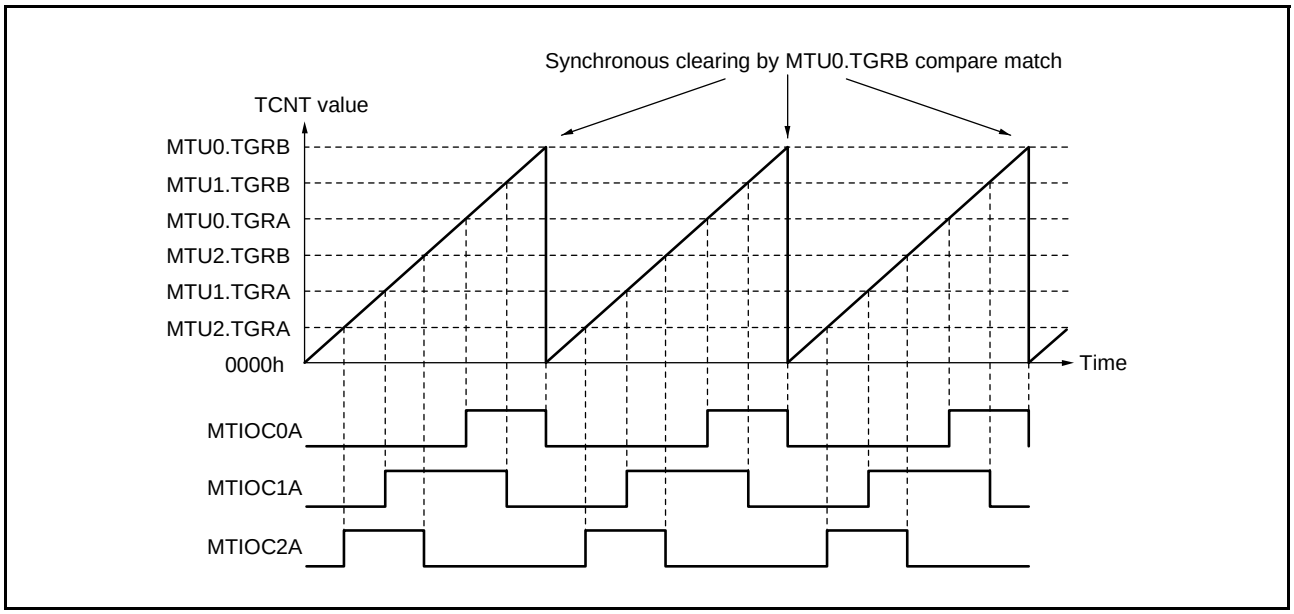


Figure 24.14 Example of Synchronous Operation

24.3.3 Buffer Operation

Buffer operation, provided for MTU0, MTU3, MTU4, MTU6, MTU7, and MTU8, enables TGRC and TGRD to be used as buffer registers. In MTU0, TGRF can also be used as a buffer register.
Buffer operation differs depending on whether TGR has been designated as an input capture register or as a compare match register.

Note: MTU0.TGRE cannot be designated as an input capture register and can only operate as a compare match register.

Table 24.61 shows the register combinations used in buffer operation.

Table 24.61 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
MTU0	TGRA	TGRC
	TGRB	TGRD
	TGRE	TGRF
MTU3	TGRA	TGRC
	TGRB	TGRD
MTU4	TGRA	TGRC
	TGRB	TGRD
MTU6	TGRA	TGRC
	TGRB	TGRD
MTU7	TGRA	TGRC
	TGRB	TGRD
MTU8	TGRA	TGRC
	TGRB	TGRD

- When TGR is an output compare register

When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register.
This operation is illustrated in Figure 24.15.

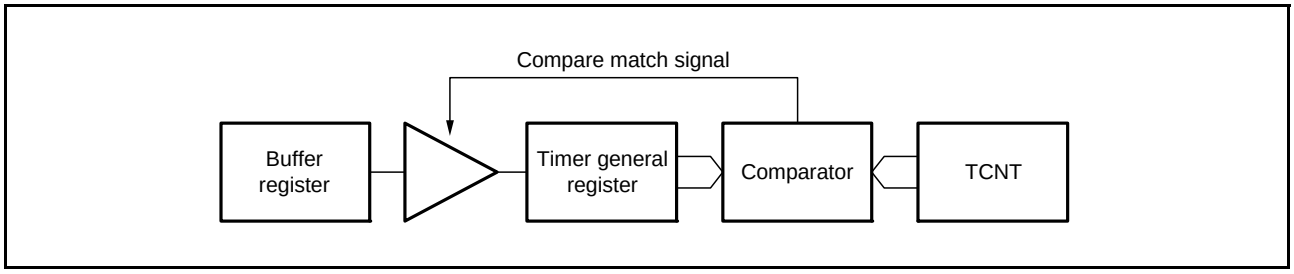


Figure 24.15 Compare Match Buffer Operation

- When TGR is an input capture register

When an input capture occurs, the value in TCNT is transferred to TGR and the value previously held in TGR is transferred to the buffer register.
This operation is illustrated in Figure 24.16.

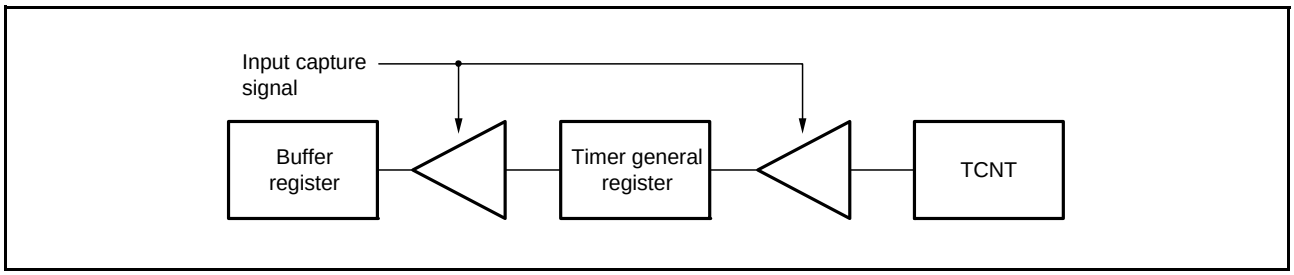


Figure 24.16 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 24.17 shows an example of the buffer operation setting procedure.

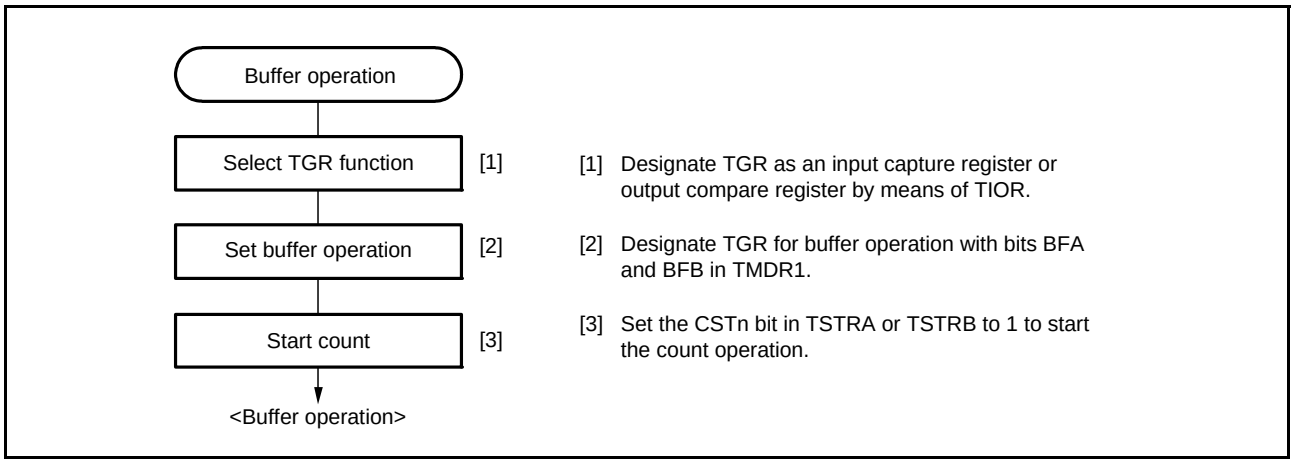


Figure 24.17 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TGR is an Output Compare Register

Figure 24.18 shows an operation example in which PWM mode 1 has been designated for MTU0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, high output at compare match A, and low output at compare match B. In this example, the TTSA bit in TBTM is set to 0. As buffer operation has been set, when compare match A occurs, the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time compare match A occurs.
For details of PWM modes, refer to section 24.3.5, PWM Modes.

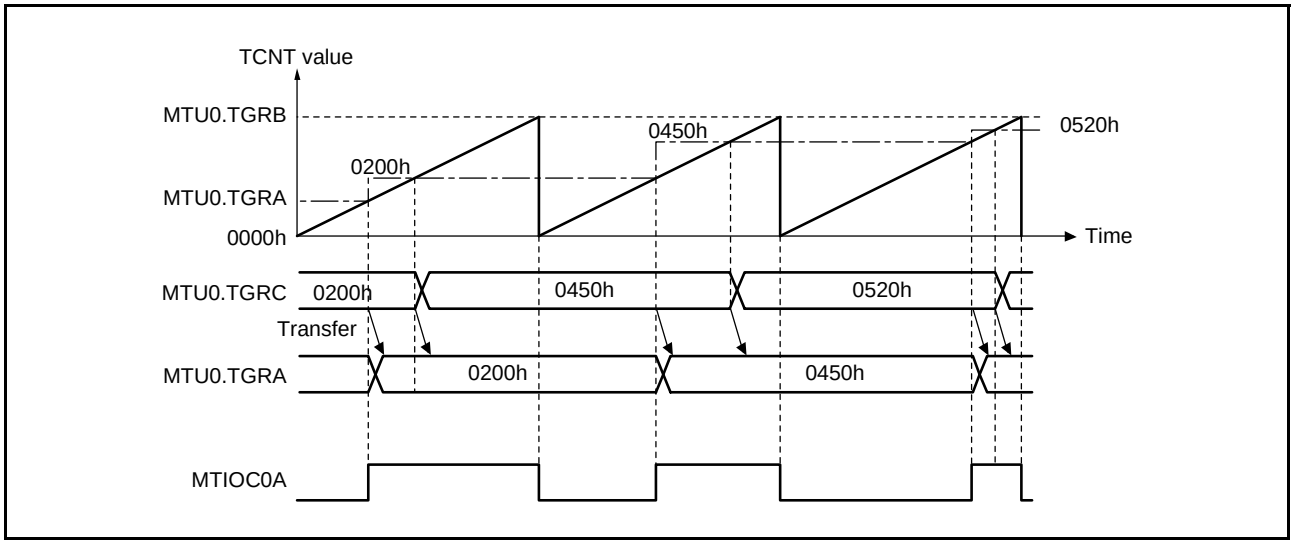


Figure 24.18 Example of Buffer Operation (1)

(b) When TGR is an Input Capture Register

Figure 24.19 shows an operation example in which TGRA has been designated as an input capture register, and buffer operation has been designated for TGRA and TGRC.

Counter clearing by TGRA input capture has been set for TCNT, and both rising and falling edges have been selected as the MTIOCnA pin input capture input edge. (n = 0 to 4, 6, 7, 8)

As buffer operation has been set, when the TCNT value is transferred to TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

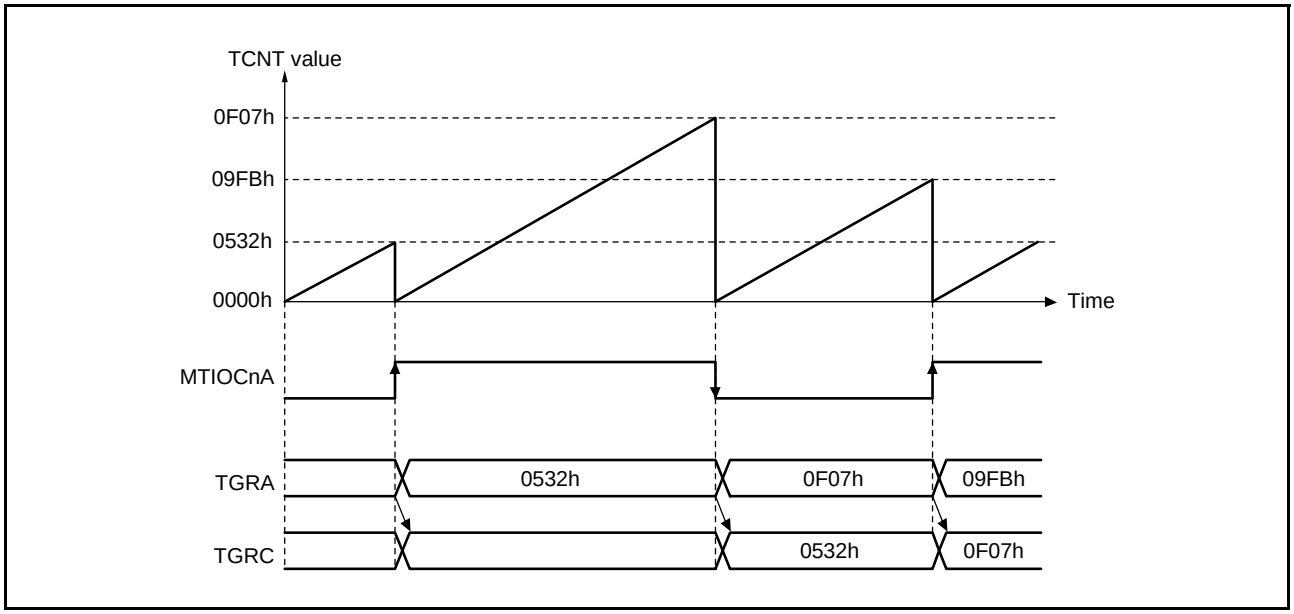


Figure 24.19 Example of Buffer Operation (2) (n = 0 to 4, 6, 7, 8)

(3) Selecting Timing for Transfer from Buffer Registers to Timer General Registers in Buffer Operation

The timing for transfer from buffer registers to timer general registers can be selected in PWM mode 1 or 2 for MTU0 or in PWM mode 1 for MTU3, MTU4, MTU6, and MTU7 by setting the buffer operation transfer mode registers (MTUn.TBTM (n = 0, 3, 4, 6, 7)). Either compare match (value after reset) or TCNT clearing can be selected for the transfer timing. TCNT clearing as transfer timing is one of the following cases.

- When TCNT overflows (FFFFh to 0000h)
- When 0000h is written to TCNT during counting
- When TCNT becomes 0000h under the condition specified in the CCLR[2:0] bits in TCR

Note: TBTM must be modified only while TCNT stops.

Figure 24.20 shows an operation example in which PWM mode 1 is designated for MTU0 and buffer operation is designated for MTU0.TGRA and MTU0.TGRC. The settings used in this example are MTU0.TCNT clearing by compare match B, high output at compare match A, and low output at compare match B. The TTSA bit in MTU0.TBTM is set to 1.

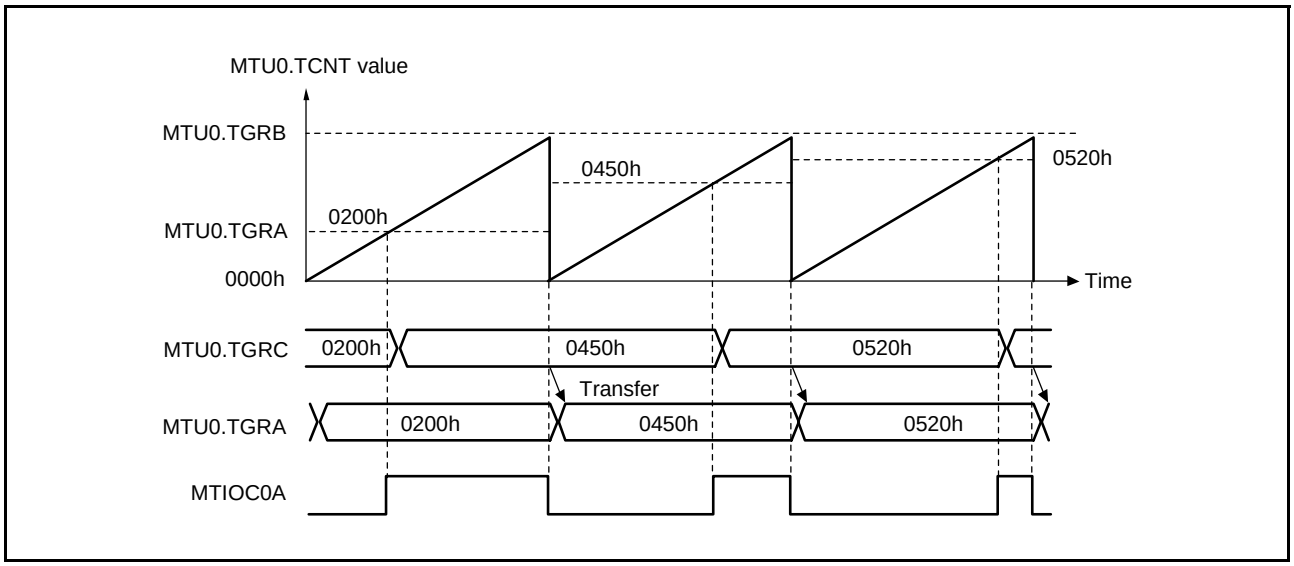


Figure 24.20 Example of Buffer Operation When MTU0.TCNT Clearing is Selected for MTU0.TGRC to MTU0.TGRA Transfer Timing

24.3.4 Cascaded Operation

In cascaded operation, two 16-bit counters in different channels are used together as a 32-bit counter.
There are two functions for connecting MTU1 and MTU2 to use as a 32-bit counter: cascade connection to be set when the MTU1.TMDR3.LWA bit is 0, and cascade connection 32-bit phase counting mode to be set when the MTU1.TMDR3.LWA bit is 1. For details on cascade connection 32-bit phase counting mode, refer to section 24.3.6.2, Cascade Connection 32-Bit Phase Counting Mode. This section describes the cascade connection function to be set when the MTU1.TMDR3.LWA bit is 0.
This function operates when the MTU1.TMDR3.LWA bit is set to 0 and the MTU1.TCR.TPSC[2:0] bits are set so that MTU1.TCNT counts at an overflow/underflow of MTU2.TCNT. Underflow occurs only when the MTU2 to which the lower 16 bits allocated is in phase counting mode.
Table 24.62 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for MTU1, the count clock setting is invalid and the counters operate independently in phase counting mode.

Table 24.62 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
MTU1 and MTU2	MTU1.TCNT	MTU2.TCNT

For simultaneous input capture of MTU1.TCNT and MTU2.TCNT during cascaded operation, additional input capture input pins can be specified by the input capture control register (TICCR). The input-capture condition is of edges in the signal produced by taking the logical OR of the input level on the main input pin and the input level on the added input pin. Accordingly, if either is at the high level, a change in the level of the other will not produce an edge for detection. For details, refer to (4), Cascaded Operation Example (c). For input capture in cascade connection, refer to section 24.6.21, Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection.
Table 24.63 shows the TICCR setting and input capture input pins.

Table 24.63 TICCR Setting and Input Capture Input Pins

Target Input Capture	TICCR Setting	Input Capture Input Pin
Input capture from MTU1.TCNT to MTU1.TGRA	I2AE bit = 0 (Initial value)	MTIOC1A
	I2AE bit = 1	MTIOC1A, MTIOC2A
Input capture from MTU1.TCNT to MTU1.TGRB	I2BE bit = 0 (Initial value)	MTIOC1B
	I2BE bit = 1	MTIOC1B, MTIOC2B
Input capture from MTU2.TCNT to MTU2.TGRA	I1AE bit = 0 (Initial value)	MTIOC2A
	I1AE bit = 1	MTIOC2A, MTIOC1A
Input capture from MTU2.TCNT to MTU2.TGRB	I1BE bit = 0 (Initial value)	MTIOC2B
	I1BE bit = 1	MTIOC2B, MTIOC1B

(1) Example of Cascaded Operation Setting Procedure

Figure 24.21 shows an example of the cascaded operation setting procedure.

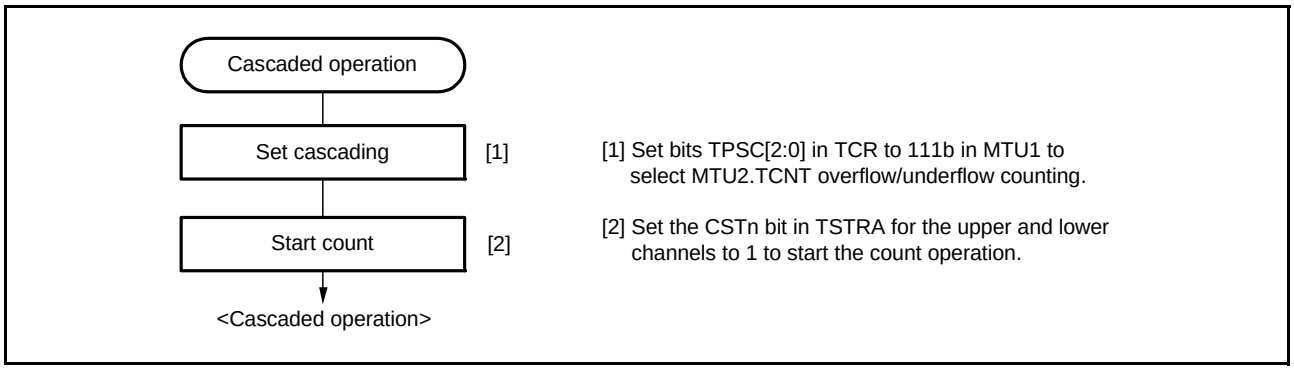


Figure 24.21 Cascaded Operation Setting Procedure

(2) Cascaded Operation Example (a)

Figure 24.22 shows the operation when MTU1.TCNT is set for counting at MTU2.TCNT overflow/underflow and MTU2 is set for phase counting mode 1 while MTU1.TCNT and MTU2.TCNT are cascaded.
MTU1.TCNT is incremented by MTU2.TCNT overflow and decremented by MTU2.TCNT underflow.

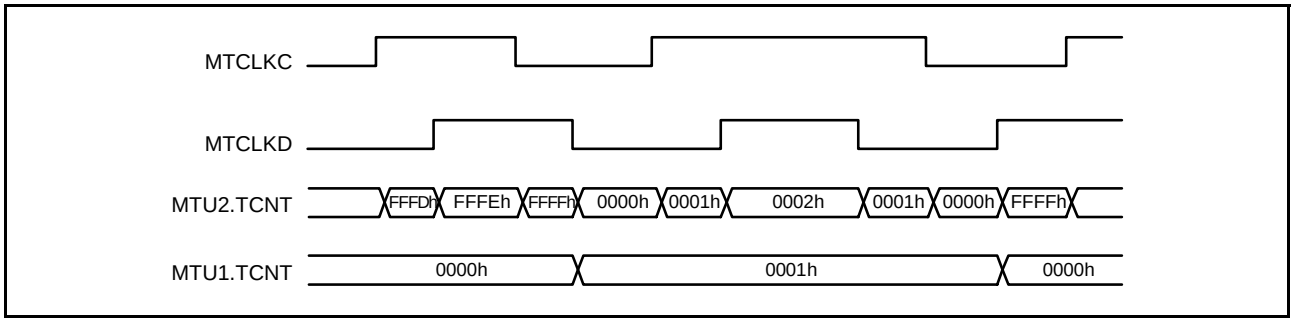


Figure 24.22 Cascaded Operation Example (a)

(3) Cascaded Operation Example (b)

Figure 24.23 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the I2AE bit in TICCR has been set to 1 to include the MTIOC2A pin in the MTU1.TGRA input capture conditions. In this example, the MTU1.TIOR.IOA[3:0] bits have selected the MTIOC1A rising edge for the input capture timing while the MTU2.TIOR.IOA[3:0] bits have selected the MTIOC2A rising edge for the input capture timing.
Under these conditions, the rising edge of both MTIOC1A and MTIOC2A is used for the MTU1.TGRA input capture condition. For the MTU2.TGRA input capture condition, the MTIOC2A rising edge is used.

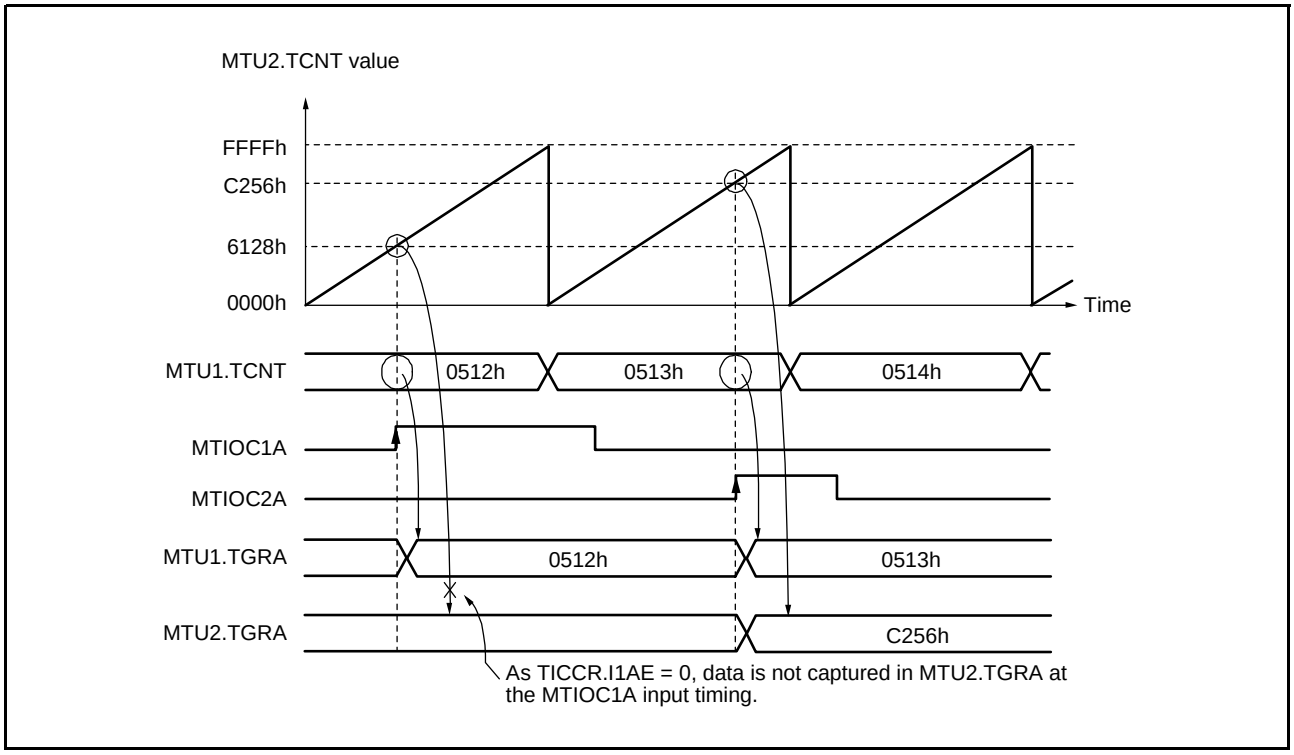
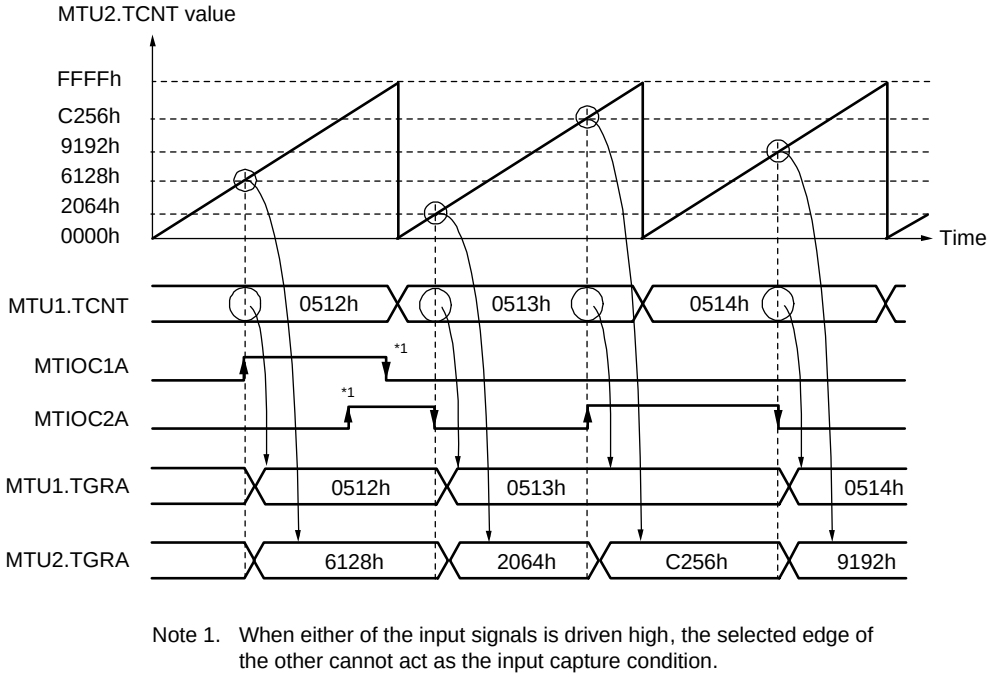


Figure 24.23 Cascaded Operation Example (b)

(4) Cascaded Operation Example (c)

Figure 24.24 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the TICCR.I2AE and I1AE bits have been set to 1 to include the MTIOC2A and MTIOC1A pins in the MTU1.TGRA and MTU2.TGRA input capture conditions, respectively. In this example, the IOA[3:0] bits in both MTU1.TIOR and MTU2.TIOR have selected both the rising and falling edges for the input capture timing. Under these conditions, the ORed result of MTIOC1A and MTIOC2A input is used for the MTU1.TGRA and MTU2.TGRA input capture conditions.



Note 1. When either of the input signals is driven high, the selected edge of the other cannot act as the input capture condition.

Figure 24.24 Cascaded Operation Example (c)

(5) Cascaded Operation Example (d)

Figure 24.25 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the TICCR.I2AE bit has been set to 1 to include the MTIOC2A pin in the MTU1.TGRA input capture conditions. In this example, the IOA[3:0] bits in MTU1.TIOR have selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing while the IOA[3:0] bits in MTU2.TIOR have selected the MTIOC2A rising edge for the input capture timing.

Under these conditions, as MTU1.TIOR has selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing, the MTIOC2A edge is not used for MTU1.TGRA input capture condition although the I2AE bit in TICCR has been set to 1.

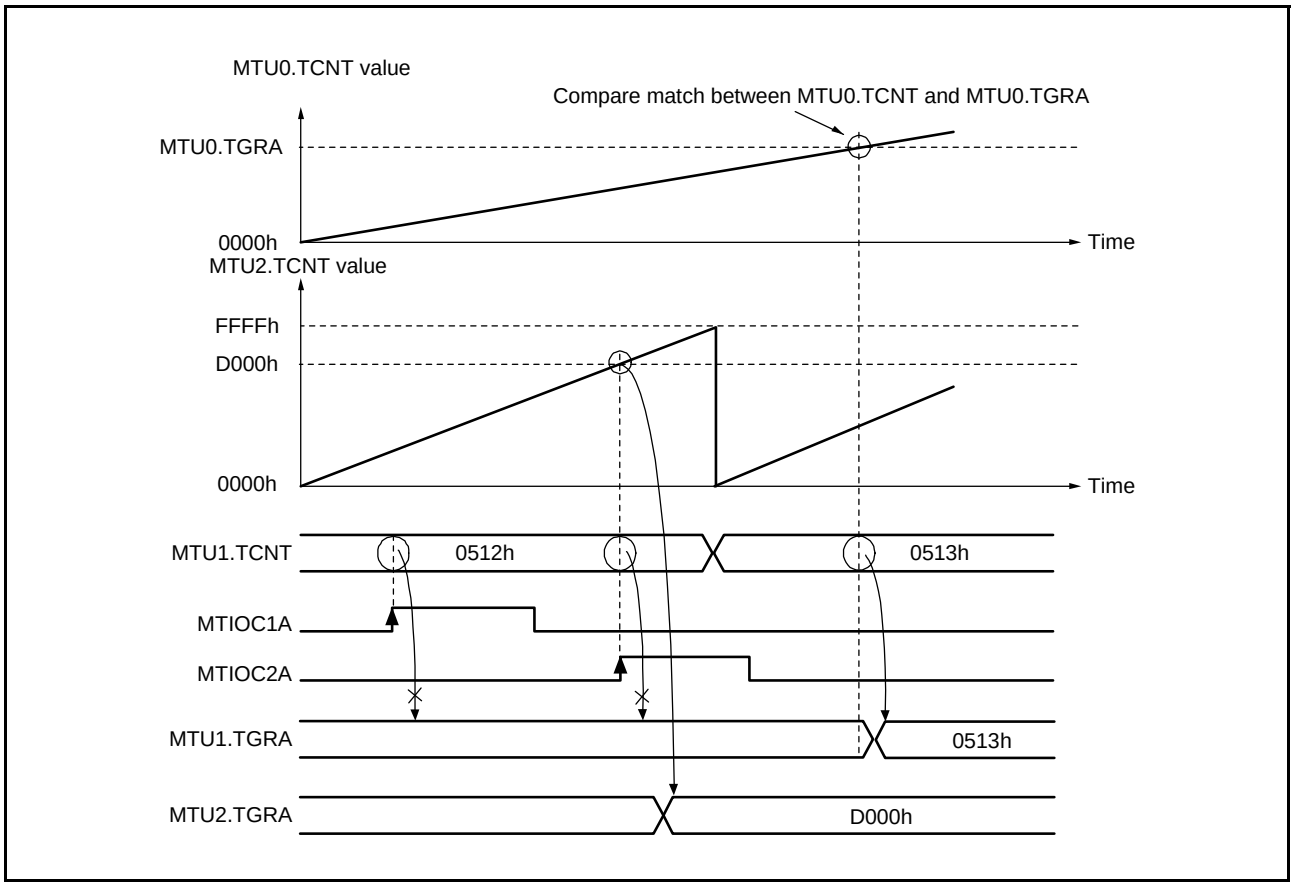


Figure 24.25 Cascaded Operation Example (d)

24.3.5 PWM Modes

PWM modes are provided to output PWM waveforms from the external pins. The output level can be selected as low, high, or toggle output in response to a compare match of each TGR.
PWM waveforms in the range of 0% to 100% duty cycle can be output according to the TGR settings.
By designating TGR compare match as the counter clearing source, the PWM period can be specified in that register.
Every channel except MTU5 and MTU8 can be set to PWM mode independently. Channels set to PWM mode can perform synchronous operation with each other or other channels set to any other mode.
There are two PWM modes as described below.

(a) PWM Mode 1

PWM waveforms are output from the MTIOCnA and MTIOCnC pins by pairing TGRA with TGRB and TGRC with TGRD. The levels specified by the TIOR.IOA[3:0] and IOC[3:0] bits are output from the MTIOCnA and MTIOCnC pins at compare matches A and C, and the level specified by the TIOR.IOB[3:0] and IOD[3:0] bits are output at compare matches B and D (n = 0 to 4, 6, 7). The initial output value is set in TGRA or TGRC. If the values set in paired TGRs are identical, the output value does not change even when a compare match occurs.
In PWM mode 1, PWM waveforms in up to 12 phases can be output.

(b) PWM Mode 2

PWM waveform output is generated using one TGR as the period register and the others as duty registers. The level specified in TIOR is output at compare matches. Upon counter clearing by a period register compare match, the initial value set in TIOR is output from each pin. If the values set in the period and duty registers are identical, the output value does not change even when a compare match occurs.

Up to eight phases of PWM waveforms can be output by combining synchronous clearing of channels that cannot be set to PWM mode 2 as synchronous operation.

The correspondence between PWM output pins and registers is shown in Table 24.64.

Table 24.64 PWM Output Registers and Output Pins

		Output Pins	
Channel	**Register**	**PWM Mode 1**	**PWM Mode 2**
MTU0	TGRA	MTIOC0A	MTIOC0A
	TGRB		MTIOC0B
	TGRC	MTIOC0C	MTIOC0C
	TGRD		MTIOC0D
MTU1	TGRA	MTIOC1A	MTIOC1A
	TGRB		MTIOC1B
MTU2	TGRA	MTIOC2A	MTIOC2A
	TGRB		MTIOC2B
MTU3	TGRA	MTIOC3A	Setting prohibited
	TGRB		
	TGRC	MTIOC3C	
	TGRD		
MTU4	TGRA	MTIOC4A	
	TGRB		
	TGRC	MTIOC4C	
	TGRD		
MTU6	TGRA	MTIOC6A	
	TGRB		
	TGRC	MTIOC6C	
	TGRD		
MTU7	TGRA	MTIOC7A	
	TGRB		
	TGRC	MTIOC7C	
	TGRD		

Note: In PWM mode 2, PWM waveform output is not possible for the TGR register in which the PWM period is set.

(1) Example of PWM Mode Setting Procedure

Figure 24.26 shows an example of the PWM mode setting procedure.

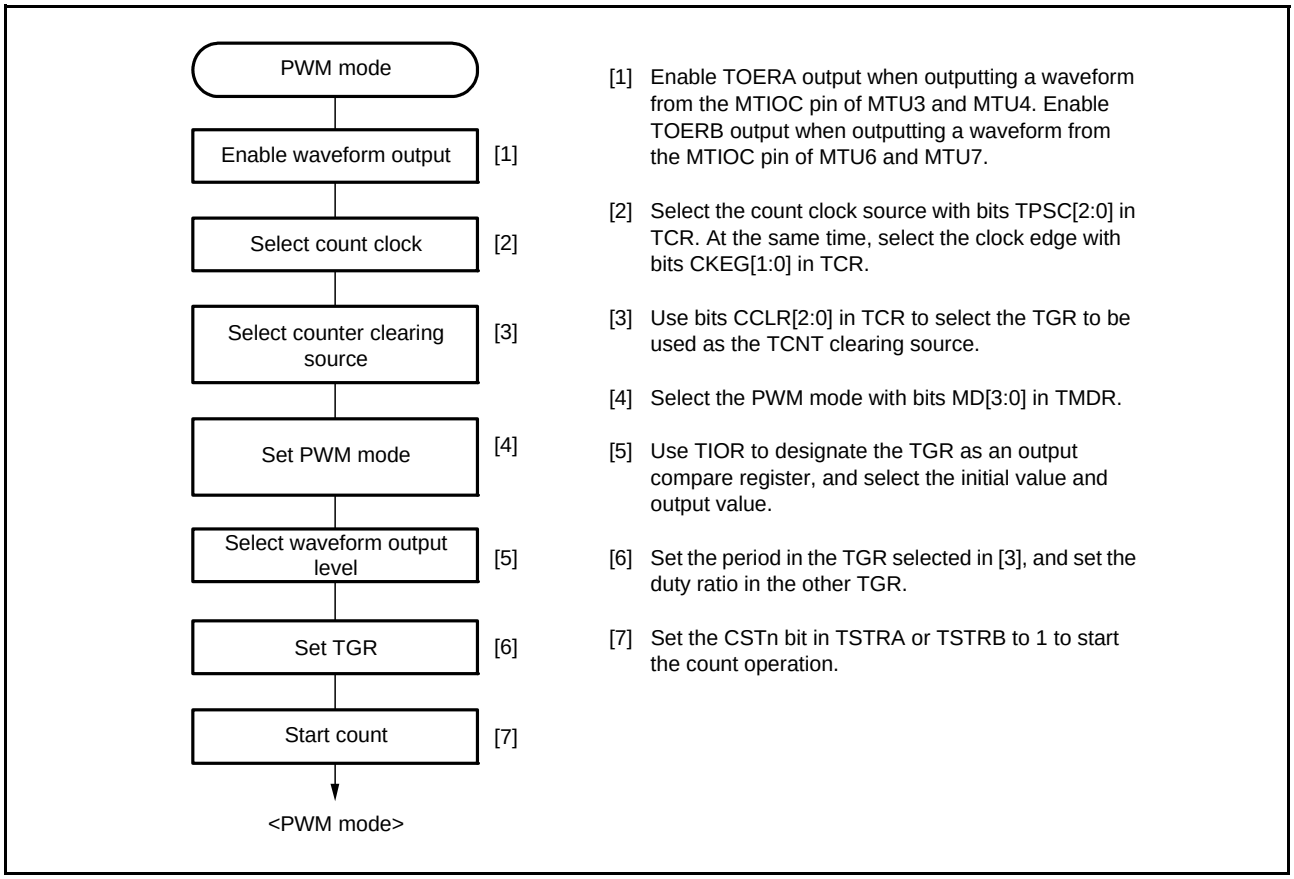


Figure 24.26 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 24.27 shows an example of operation in PWM mode 1.
In this example, TGRA compare match is set as the TCNT clearing source, 0 is set as the initial output value and output value for TGRA, and 1 is set as the output value for TGRB.
In this case, the value set in TGRA is used as the period, and the value set in TGRB is used as the duty ratio.

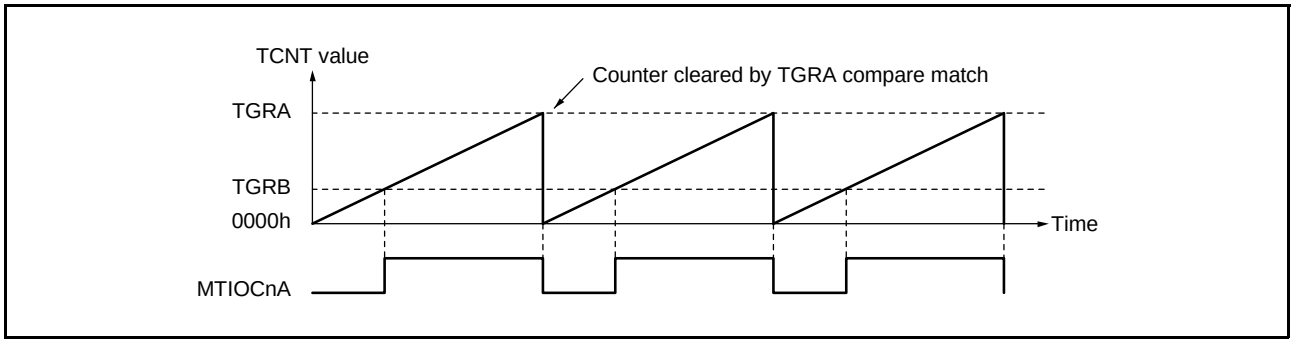


Figure 24.27 Example of PWM Mode 1 Operation (n = 0 to 4, 6, 7)

Figure 24.28 shows an example of operation in PWM mode 2.
In this example, synchronous operation is designated for MTU0 and MTU1, MTU1.TGRB compare match is set as the TCNT clearing source, and low is set as the initial output value and high as the output value for the other TGR registers (MTU0.TGRA to MTU0.TGRD and MTU1.TGRA), outputting 5-phase PWM waveforms.
In this case, the value set in MTU1.TGRB is used as the period, and the values set in the other TGRs are used as the duty ratio.

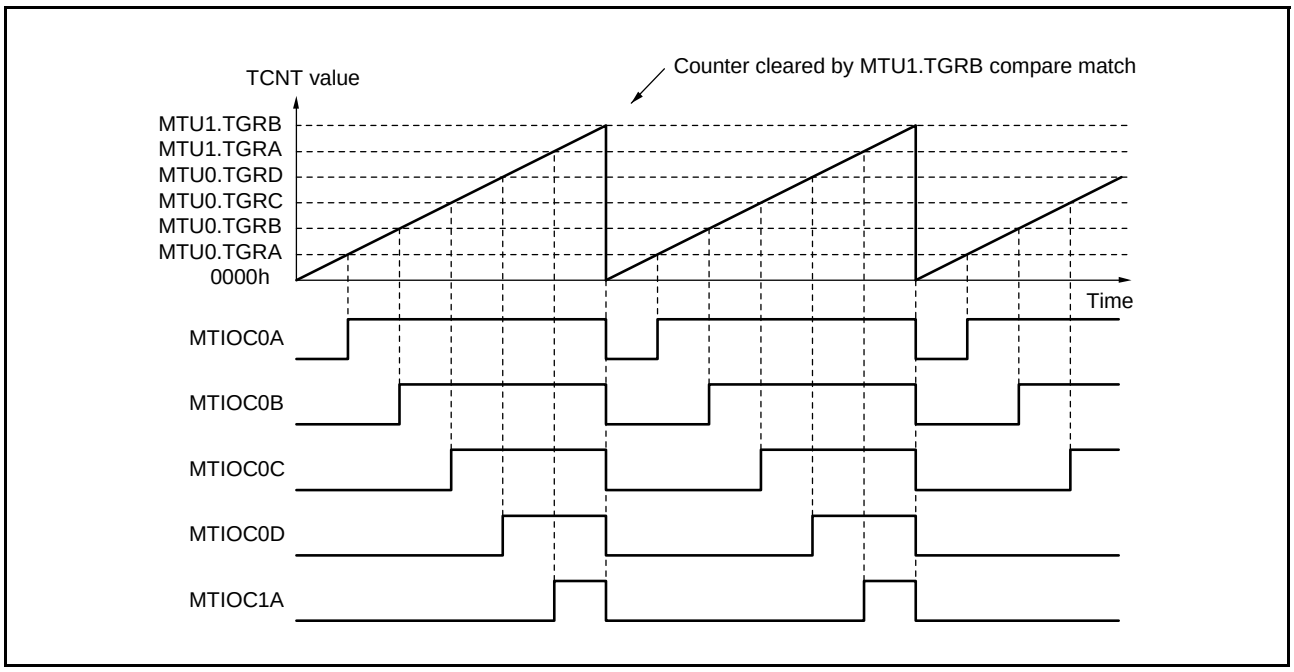


Figure 24.28 Example of PWM Mode 2 Operation

Figure 24.29 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode 1.
In this example, TGRA compare match is set as the TCNT clearing source, a low level is set as the initial output value and output value for TGRA, and a high level is set as the output value for TGRB.

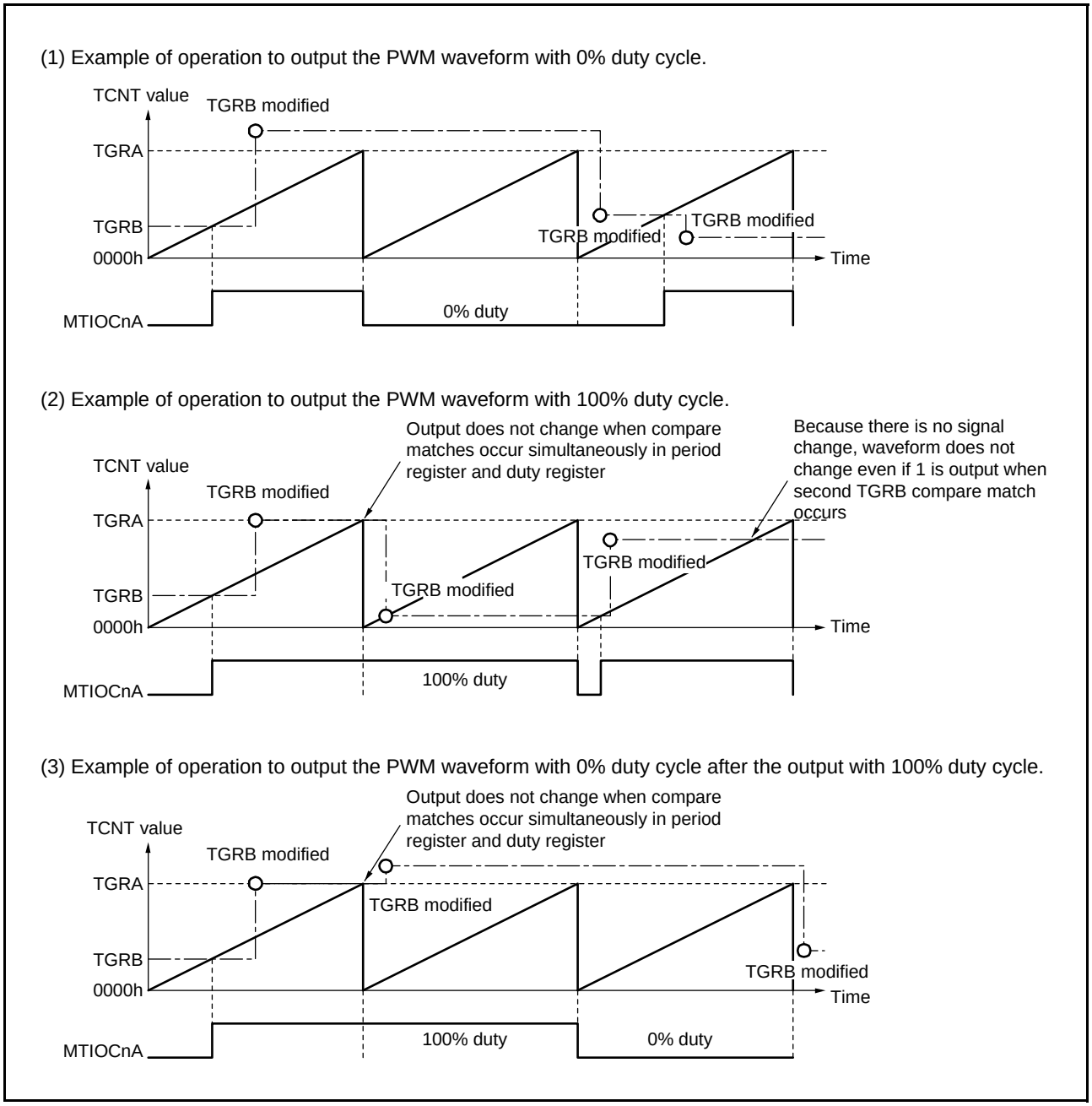


Figure 24.29 Examples of PWM Mode Operation (PWM Waveform Output with 0% Duty and 100% Duty) (n = 0 to 4, 6, 7)

24.3.6 Phase Counting Mode

There are two phase counting modes: 16-bit phase counting mode in which MTU1 and MTU2 operate independently, and cascade connection 32-bit phase counting mode in which MTU1 and MTU2 are cascaded.
In phase counting mode, the phase difference between two external input clocks is detected and the corresponding TCNT is incremented or decremented.
Two external clock input pins for each phase counting mode are not affected by the settings of TCR.TPSC[2:0], TCR2.TPSC2[2:0], and CKEG[1:0]. The two external clock input pins used in 16-bit phase counting mode and cascade connection 32-bit phase counting mode of MTU2 can be selected by MTU1.TMDR3.PHCKSEL. In a phase counting mode other than 16-bit phase counting mode and cascade connection 32-bit phase counting mode of MTU2, MTCLKA and MTCLKB are selected for A-phase and B-phase, respectively. In phase counting mode, the external clock pins MTCLKA, MTCLKB, MTCLKC, and MTCLKD are used for two-phase encoder pulse input.
Table 24.65 lists the external clock input pins to be connected in each phase counting mode.

Table 24.65 Clock Input Pins in Phase Counting Mode

		External Clock Input Pins	
Phase Counting Mode	**TMDR3.PHCKSEL bit**	**A-Phase**	**B-Phase**
MTU1 16-bit phase counting mode	x (Don't care)	MTCLKA	MTCLKB
MTU2 16-bit phase counting mode	0	MTCLKA	MTCLKB
	1 (initial value)	MTCLKC	MTCLKD
Cascade connection 32-bit phase counting mode	0	MTCLKA	MTCLKB
	1 (initial value)	MTCLKC	MTCLKD

24.3.6.1 16-Bit Phase Counting Mode

When the MTU1.TMDR3.LWA is 0, 16-bit phase counting mode can be set individually for MTU1 and MTU2.
In 16-bit phase counting mode, the phase difference between two external input clocks is detected and the 16-bit counter TCNT of the corresponding channel is incremented or decremented.
When 16-bit phase counting mode is specified, an external clock is selected as the count clock and TCNT operates as an up-counter/down-counter regardless of the setting of TCR.TPSC[2:0], TCR2.TPSC2[2:0], and CKEG[1:0]. However, the functions of TCR.CCLR[1:0],TIOR, TIER, and TGR are enabled, and input capture/compare match and interrupt functions can be used.
These external input pins can be used for two-phase encoder pulse input.
When an overflow occurs while TCNT is counting up and the corresponding TIER.TCIEV bit is 1, a TCIV interrupt is generated. When an underflow occurs while TCNT is counting down and the corresponding TIER.TCIEU bit is 1, a TCIU interrupt is generated.
The TSR.TCFD flag is the count direction flag. Read the TCFD flag to check whether TCNT is counting up and down.

(1) Example of 16-Bit Phase Counting Mode Setting Procedure

Figure 24.30 shows an example of the phase counting mode setting procedure.

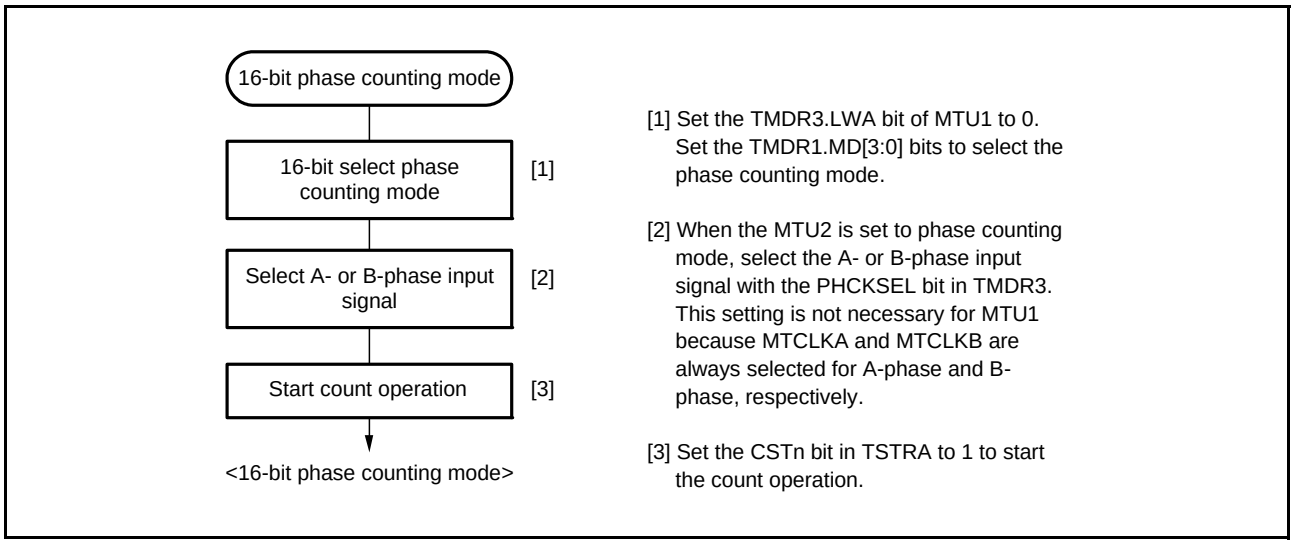


Figure 24.30 Example of 16-Bit Phase Counting Mode Setting Procedure

(2) Examples of 16-Bit Phase Counting Mode Operation

In phase counting mode, TCNT is incremented or decremented according to the phase difference between two external clocks. There are five modes according to the count conditions. Each mode operates under the condition PHCKSEL = 1, which means the phase clock for MTU1 is input from MTCLKA or MTCLKB and that for MTU2 is input from MTCLKC or MTCLKD.

(a) Phase Counting Mode 1

Figure 24.31 shows an example of operation in phase counting mode 1, and Table 24.66 summarizes the TCNT up-counting and down-counting conditions.

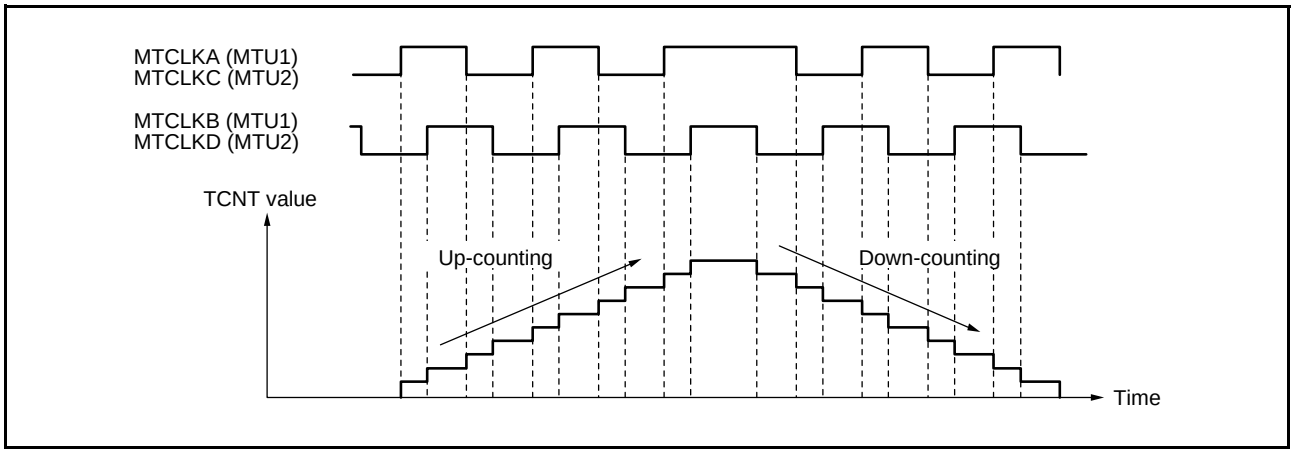


Figure 24.31 Example of Operation in Phase Counting Mode 1

Table 24.66 Up-Counting and Down-Counting Conditions in Phase Counting Mode 1

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High	↑	Up-counting
Low	↓	
↑	Low	
↓	High	
High	↓	Down-counting
Low	↑	
↑	High	
↓	Low	

↑ : Rising edge

↓ : Falling edge

(b) Phase Counting Mode 2

Figure 24.32 to Figure 24.34 show the examples of operation in phase counting mode 2 and Table 24.67 summarizes the TCNT up-counting and down-counting conditions.

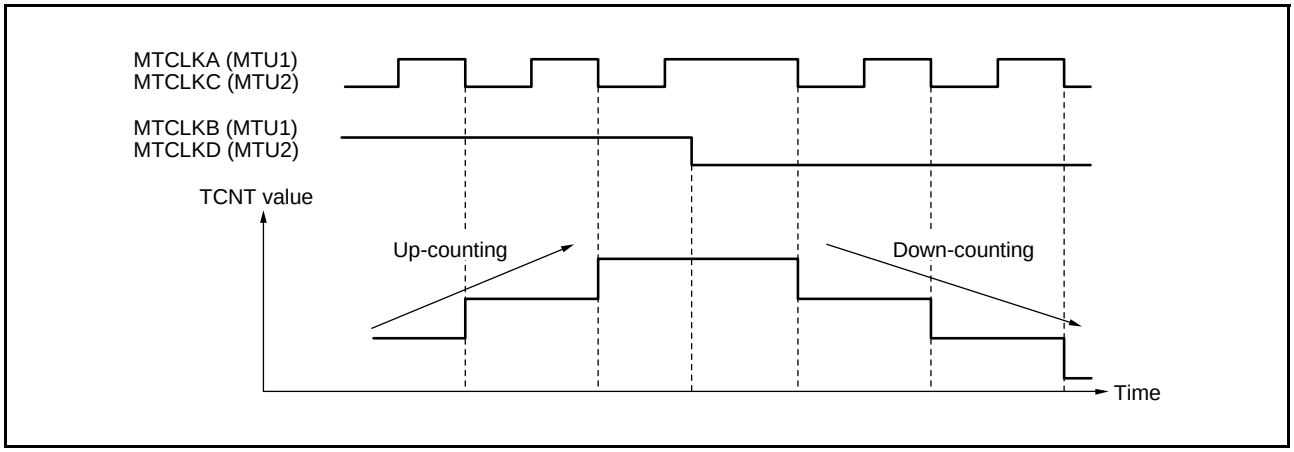


Figure 24.32 Example of Operation in Phase Counting Mode 2 (When MTUn.TCR2.PCB[1:0] = 00b (n = 1, 2))

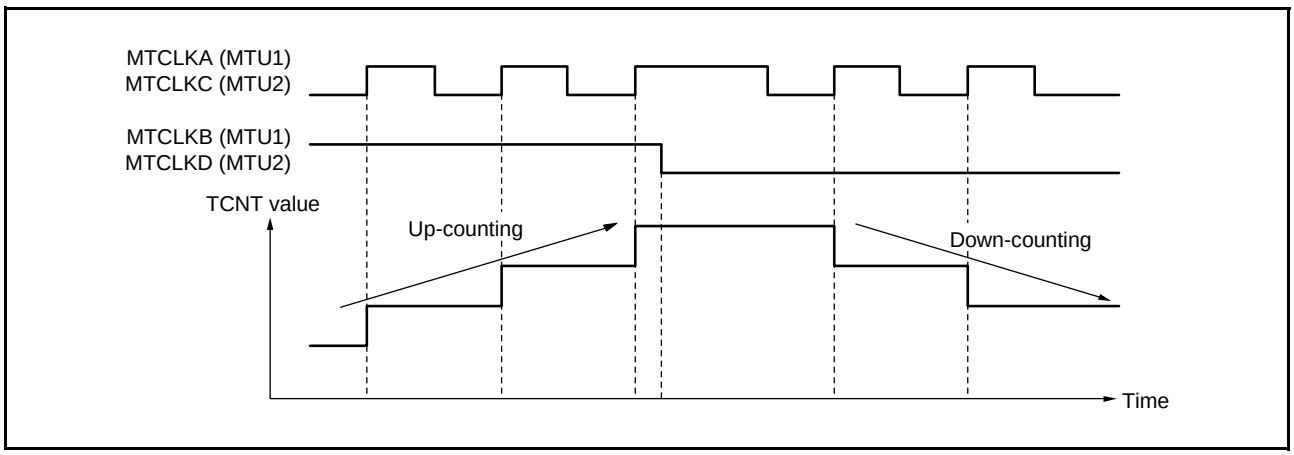


Figure 24.33 Example of Operation in Phase Counting Mode 2 (When MTUn.TCR2.PCB[1:0] = 01b (n = 1, 2))

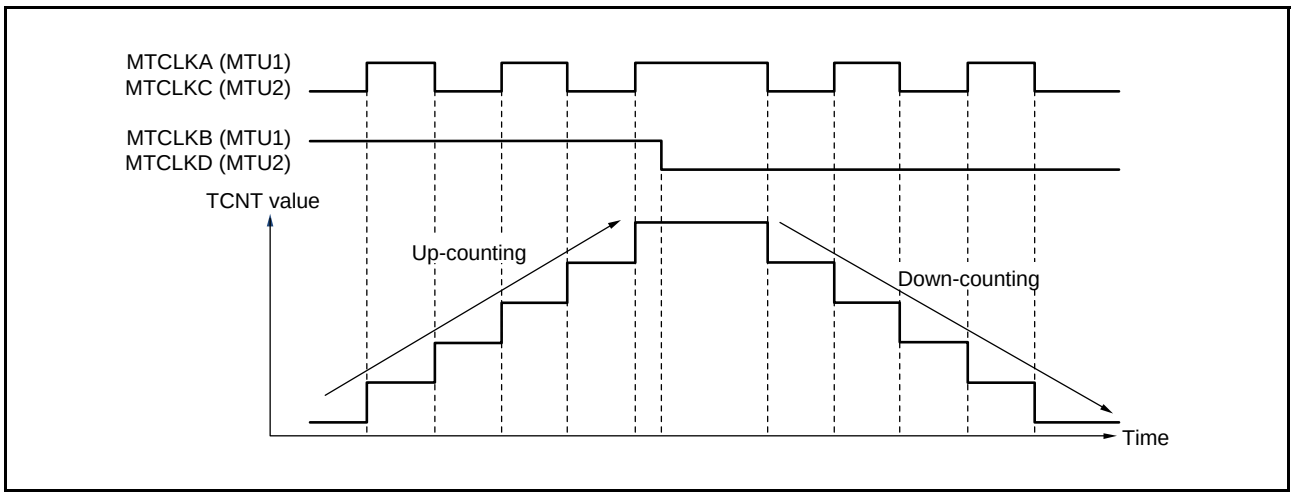


Figure 24.34 Example of Operation in Phase Counting Mode 2 (When MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 24.67 Up-Counting and Down-Counting Conditions in Phase Counting Mode 2

PCB[1:0]	MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
00b	High	↑	Not counted (Don't care)
	Low	↓	
	↑	Low	
	↓	High	Up-counting
	High	↓	Not counted (Don't care)
	Low	↑	
	↑	High	
	↓	Low	Down-counting
01b	High	↑	Not counted (Don't care)
	Low	↓	
	↑	Low	Down-counting
	↓	High	Not counted (Don't care)
	High	↓	
	Low	↑	
	↑	High	Up-counting
	↓	Low	Not counted (Don't care)
1xb	High	↑	Not counted (Don't care)
	Low	↓	
	↑	Low	Down-counting
	↓	High	Up-counting
	High	↓	Not counted (Don't care)
	Low	↑	
	↑	High	Up-counting
	↓	Low	Down-counting

↑ : Rising edge

↓ : Falling edge

(c) Phase Counting Mode 3

Figure 24.35 to Figure 24.37 show the examples of operation in phase counting mode 3 and Table 24.68 summarizes the TCNT up-counting and down-counting conditions.

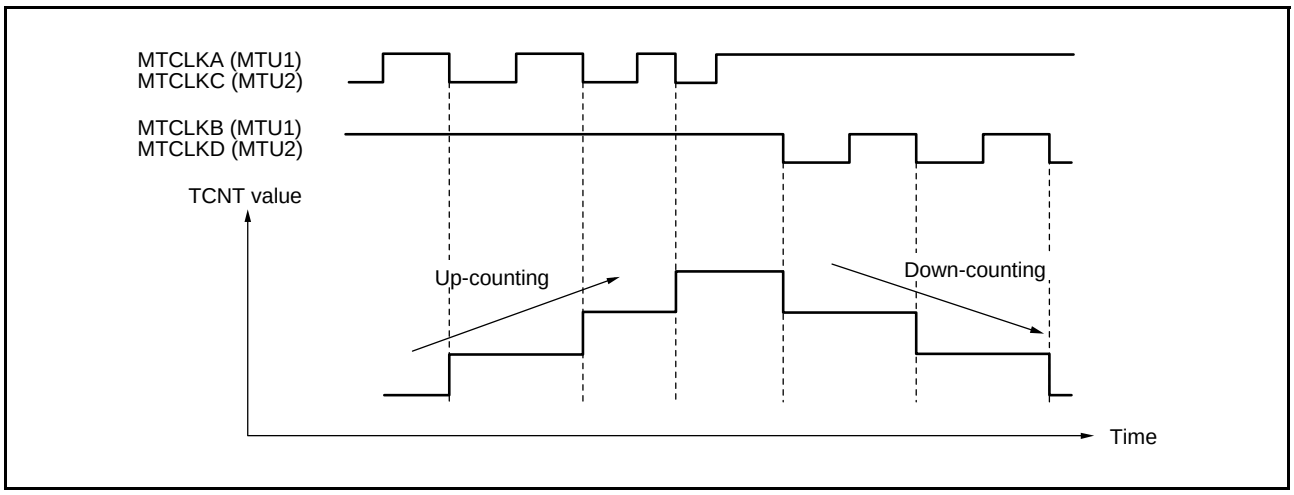


Figure 24.35 Example of Operation in Phase Counting Mode 3 (When MTUn.TCR2.PCB[1:0] = 00b (n = 1, 2))

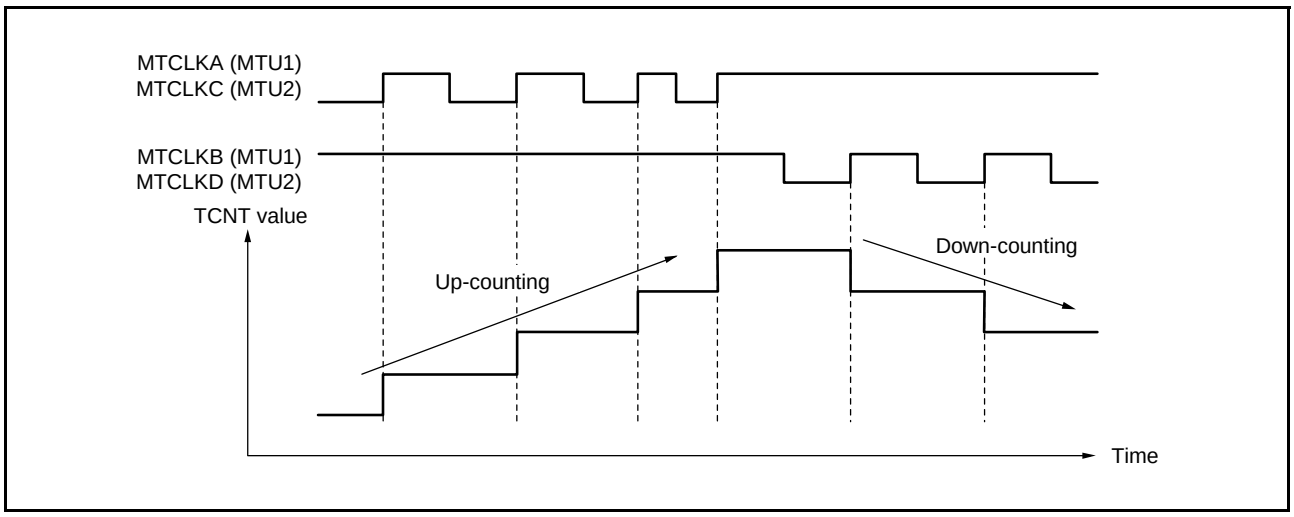


Figure 24.36 Example of Operation in Phase Counting Mode 3 (When MTUn.TCR2.PCB[1:0] = 01b (n = 1, 2))

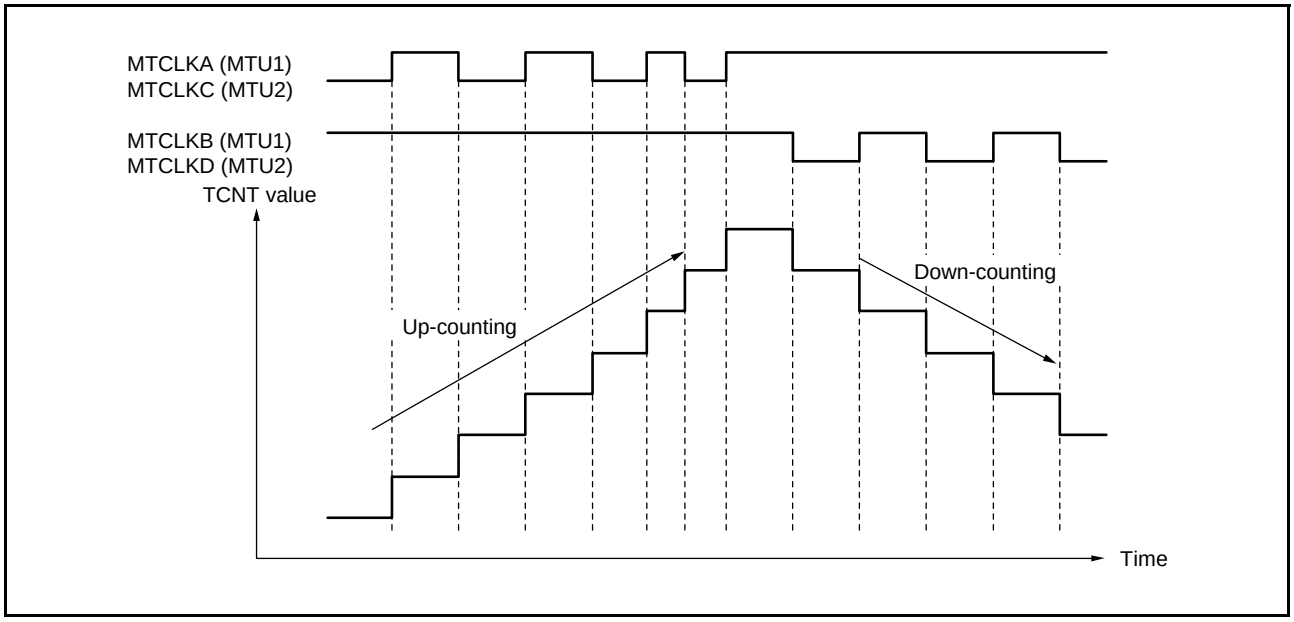


Figure 24.37 Example of Operation in Phase Counting Mode 3 (When MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 24.68 Up-Counting and Down-Counting Conditions in Phase Counting Mode 3

PCB[1:0]	MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
00b	High	↑	Not counted (Don't care)
	Low	↓	
	↑	Low	
	↓	High	Up-counting
	High	↓	Down-counting
	Low	↑	Not counted (Don't care)
	↑	High	
	↓	Low	
01b	High	↑	Down-counting
	Low	↓	Not counted (Don't care)
	↑	Low	
	↓	High	
	High	↓	
	Low	↑	
	↑	High	Up-counting
	↓	Low	Not counted (Don't care)
1xb	High	↑	Down-counting
	Low	↓	Not counted (Don't care)
	↑	Low	
	↓	High	Up-counting
	High	↓	Down-counting
	Low	↑	Not counted (Don't care)
	↑	High	Up-counting
	↓	Low	Not counted (Don't care)

↑ : Rising edge

↓ : Falling edge

(d) Phase Counting Mode 4

Figure 24.38 shows an example of operation in phase counting mode 4, and Table 24.69 summarizes the TCNT up-counting and down-counting conditions.

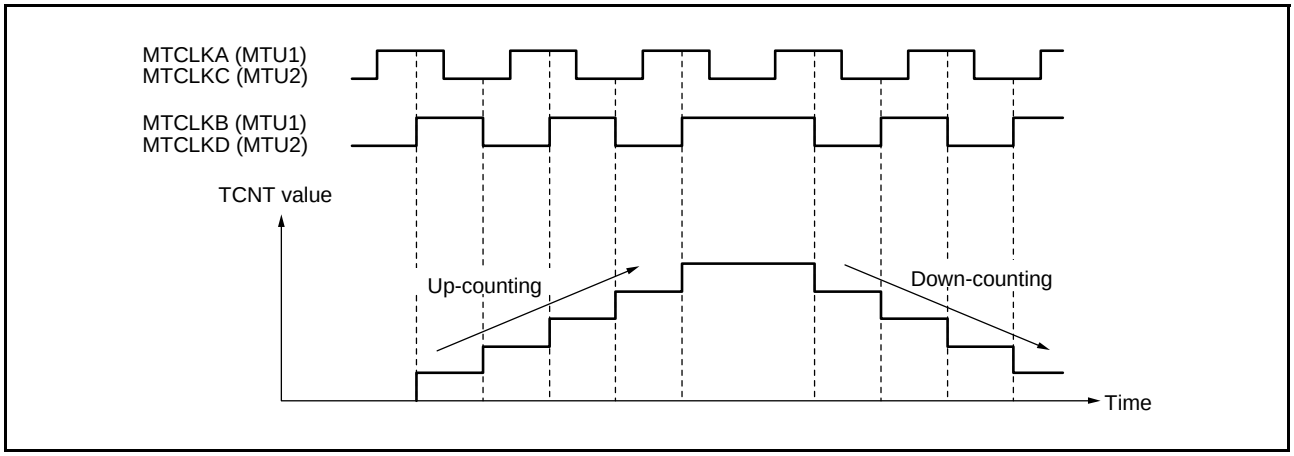


Figure 24.38 Example of Operation in Phase Counting Mode 4

Table 24.69 Up-Counting and Down-Counting Conditions in Phase Counting Mode 4

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High	↑	Up-counting
Low	↓	
↑	Low	Not counted (Don't care)
↓	High	
High	↓	Down-counting
Low	↑	
↑	High	Not counted (Don't care)
↓	Low	

↑ : Rising edge

↓ : Falling edge

(e) Phase Counting Mode 5

Figure 24.39 and Figure 24.40 show the examples of operation in phase counting mode 5 and Table 24.70 summarizes the TCNT up-counting and down-counting conditions.

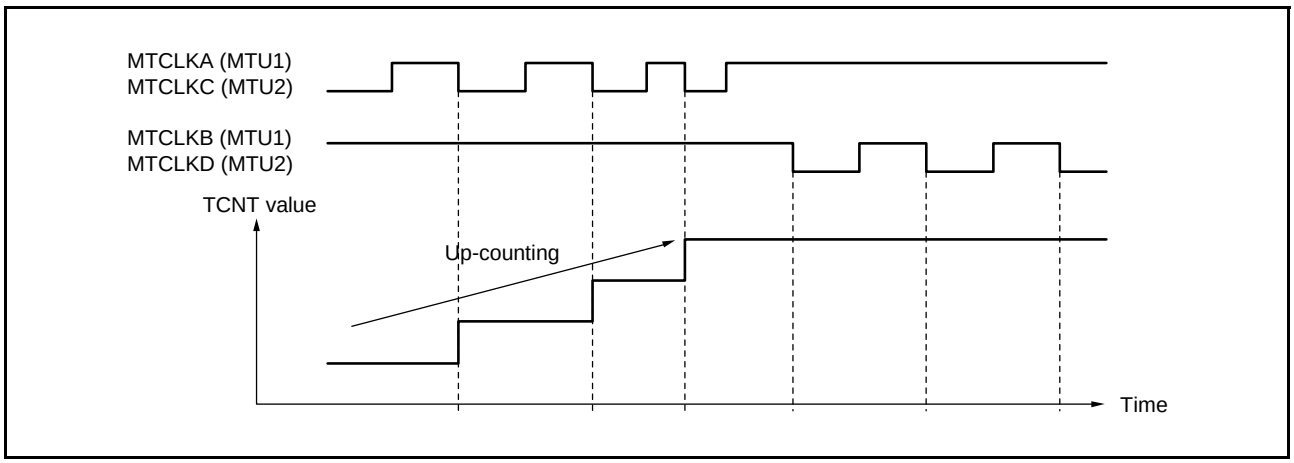


Figure 24.39 Example of Operation in Phase Counting Mode 5 (When MTUn.TCR2.PCB[1:0] = 0xb (n = 1, 2))

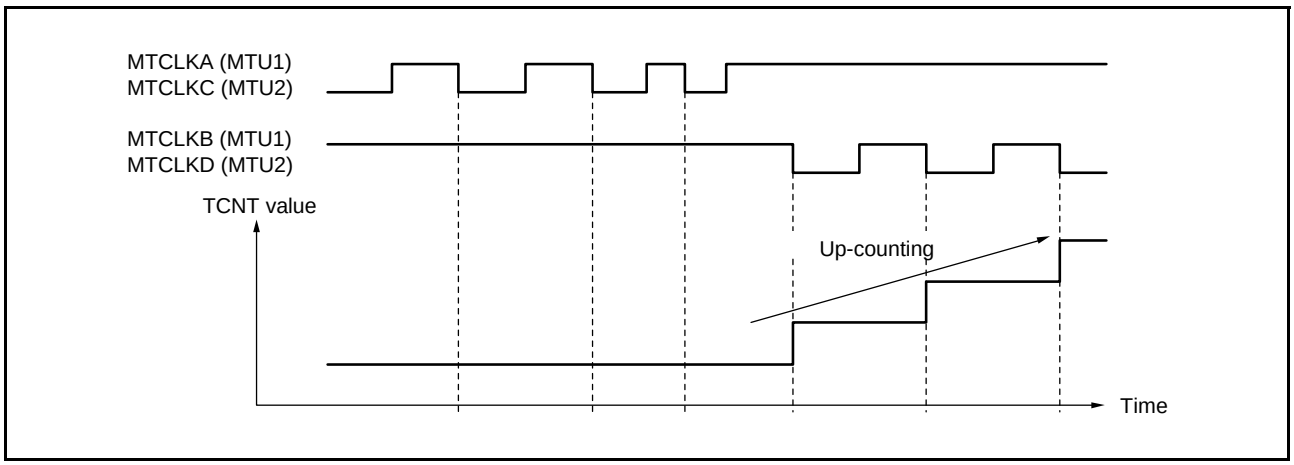


Figure 24.40 Example of Operation in Phase Counting Mode 5 (When MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 24.70 Up-Counting and Down-Counting Conditions in Phase Counting Mode 5

PCB[1:0]	MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
0xb	High	↑	Not counted (Don't care)
	Low	↓	
	↑	Low	
	↓	High	Up-counting
	High	↓	Not counted (Don't care)
	Low	↑	
	↑	High	
	↓	Low	Up-counting
1xb	High	↑	Not counted (Don't care)
	Low	↓	Up-counting
	↑	Low	Not counted (Don't care)
	↓	High	
	High	↓	Up-counting
	Low	↑	Not counted (Don't care)
	↑	High	
	↓	Low	

↑ : Rising edge

↓ : Falling edge

(3) 16-Bit Phase Counting Mode Application Example

Figure 24.41 shows an example in which MTU1 is in phase counting mode, and MTU1 is coupled with MTU0 to input 2-phase encoder pulses of a servo motor in order to detect position or speed.

MTU1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to MTCLKA and MTCLKB.

In MTU0, MTU0.TGRC compare match is specified as the TCNT clearing source and MTU0.TGRA and MTU0.TGRC are used for the compare match function and are set with the speed control period and position control period.

MTU0.TGRB is used for input capture, with MTU0.TGRB and MTU0.TGRD operating in buffer mode. The MTU1 count clock is designated as the MTU0.TGRB input capture source, and the widths of 2-phase encoder 4-multiplication pulses are detected.

MTU1.TGRA and MTU1.TGRB for MTU1 are designated for the input capture function and MTU0.TGRA and MTU0.TGRC compare matches in MTU0 are selected as the input capture sources to store the up/down-counter values for the control periods.

This procedure enables the accurate detection of position and speed.

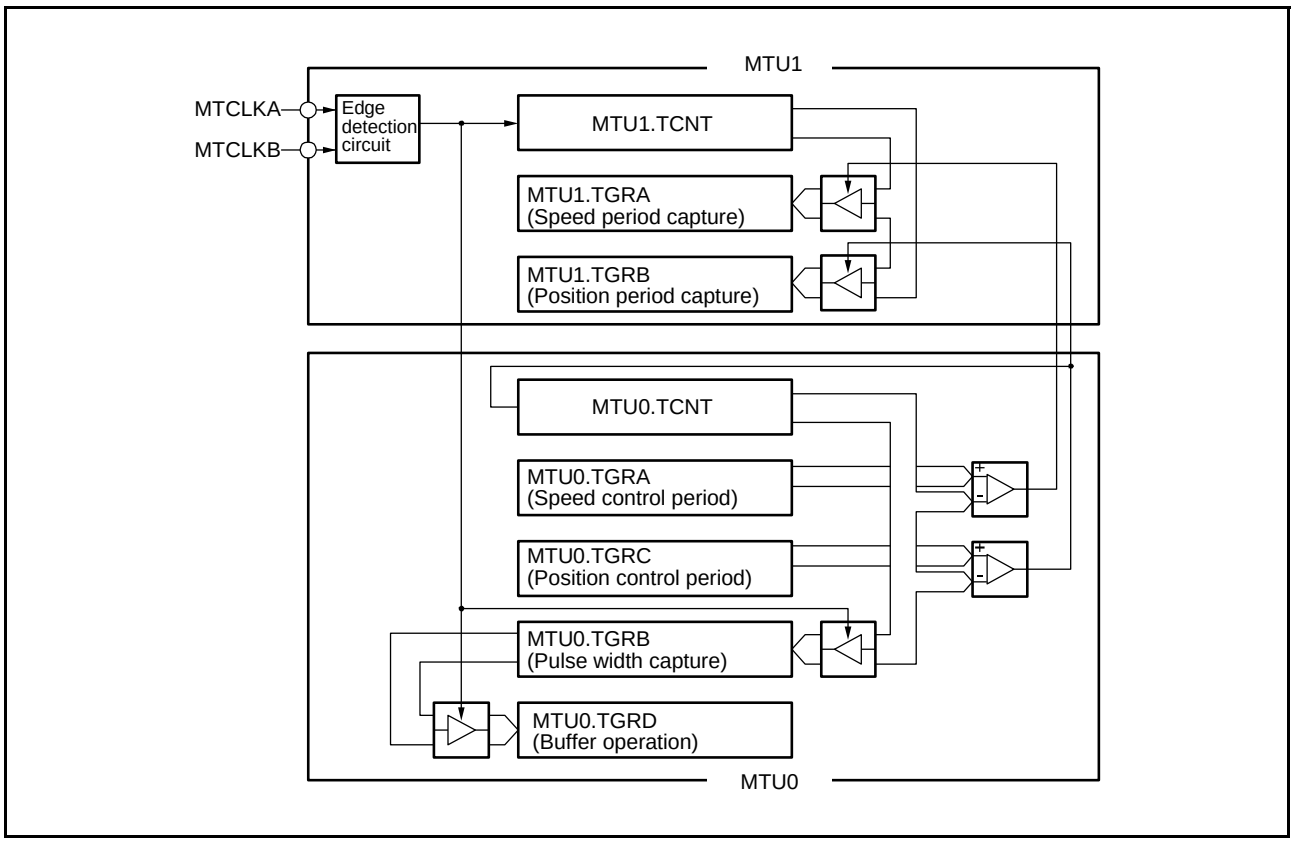


Figure 24.41 16-Bit Phase Counting Mode Application Example

24.3.6.2 Cascade Connection 32-Bit Phase Counting Mode

When MTU1 is set to phase counting mode by setting MTU1.TMDR3.LWA = 1, MTU1 and MTU2 are connected to operate in cascade connection 32-bit phase counting mode as shown in Figure 24.42. When this mode is used, the TCR, TCR2, TIOR, TIER, TGR, and TSR registers are controlled by MTU1 and the settings of MTU2 are disabled. Refer to Figure 24.43 for the procedure for setting cascade connection 32-bit phase counting mode.

In this mode, three-phase (A, B, and Z) signals can be input. As an encoder pulse signal, the external input phase clocks MTCLKA and MTCLKB or MTCLKC and MTCLKD can be selected for A-phase and B-phase, and MTIOC1A can be selected for Z-phase, respectively. Refer to Table 24.69 for selecting external clock input of A-phase and B-phase. A counter event is generated using an A-phase or B-phase pulse and counted by the 32-bit counter MTU1.TCNTLW.

An input capture can also be generated using a Z-phase signal; thus angular velocity can be measured using the captured value in the general register.

Furthermore, MTU8 can be used as a channel for measuring a 1-ms interval, and a compare match signal can be output at a 1-ms interval to the MTU1 and MTU2, which operate in cascade connection 32-bit phase counting mode. That is, a compare match signal of MTU8 is used as a capture signal of MTU1 and MTU2, and the number of A-phase and B-phase pulses for a 1-ms period can be measured.

When MTU0 or MTU5 is specified as the channel for measuring a Z-phase signal pulse, the compare match signal of the MTU8.TGRC register can be output as a capture signal or clear signal to MTU0 or MTU5, thus the Z-phase count at a 1-ms interval can be measured.

In addition, a counter event signal of combined MTU1 and MTU2 can be used as a capture signal of the MTU8.TGRD register, and measurement can be performed including the intervals of A, B, or both phase pulses. In this case, the MTU8.TGRD register should be set to buffer operation.

Refer to section 24.3.4, Cascaded Operation, for details on the cascade connection function for connecting MTU1 and MTU2 in a mode other than cascade connection 32-bit phase counting mode.

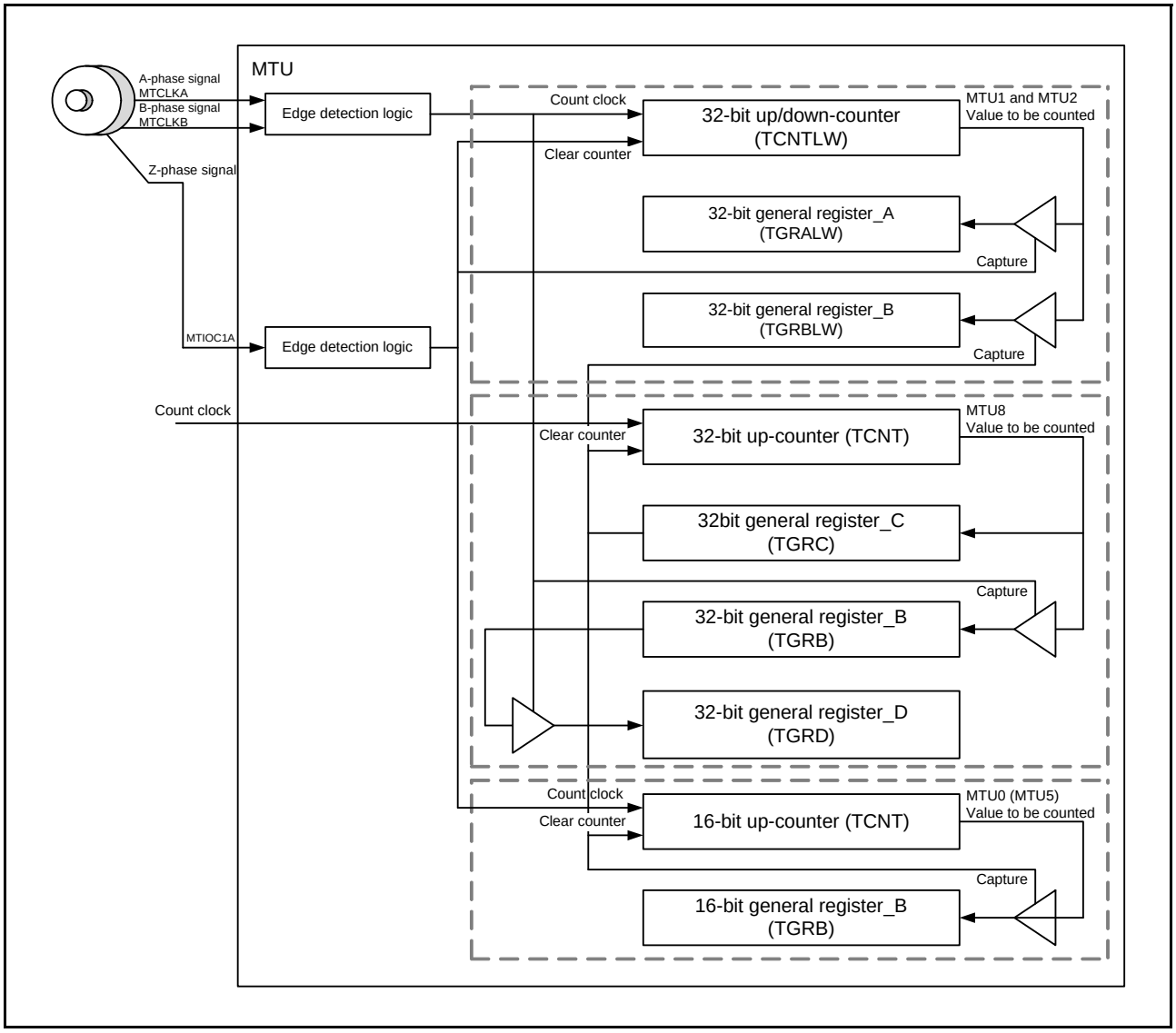


Figure 24.42 Block Diagram for Operation in Cascade Connection 32-Bit Phase Counting Mode

(1) Example of Setting Cascade Connection 32-Bit Phase Counting Mode

Figure 24.43 shows an example of the procedure for setting cascade connection 32-bit phase counting mode.

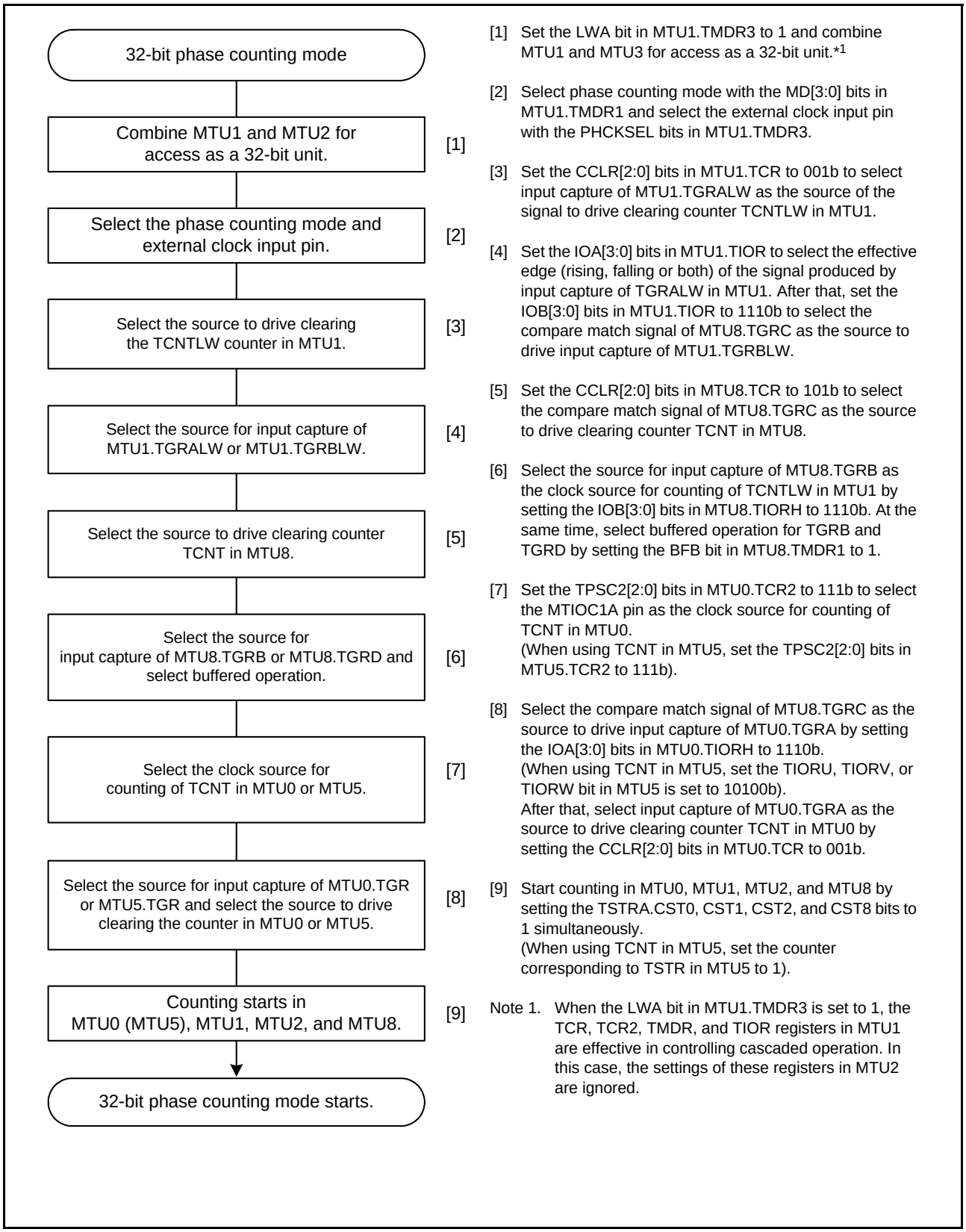


Figure 24.43 Procedure for Setting Cascade Connection 32-Bit Phase Counting Mode

24.3.7 Reset-Synchronized PWM Mode

In the reset-synchronized PWM mode, six phases of positive and negative PWM waveforms (12 phases in total) that share a common wave transition point can be output by combining MTU3 and MTU4 and MTU6 and MTU7.
When set for reset-synchronized PWM mode, the MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D, MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, and MTIOC7D pins function as PWM output pins and timer counters 3 and 6 (MTU3.TCNT and MTU6.TCNT) functions as an up-counter.
Table 24.71 shows the PWM output pins used. Table 24.72 shows the settings of the registers.

Table 24.71 Output Pins for Reset-Synchronized PWM Mode

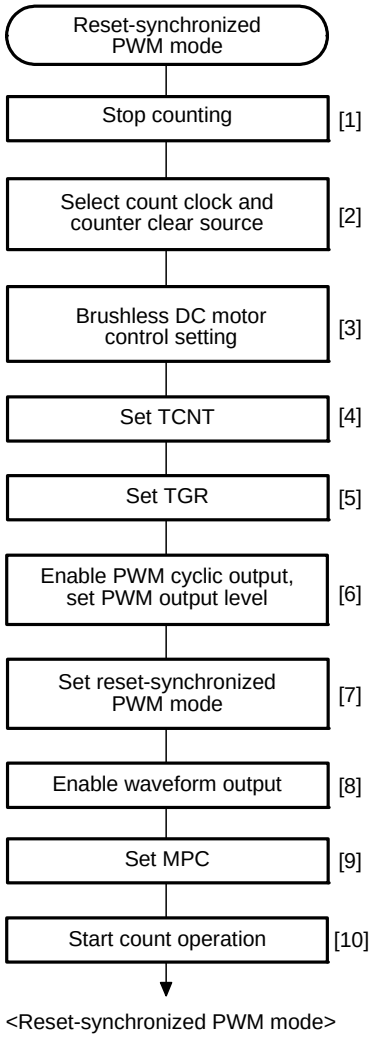
Channel	Output Pin	Description
MTU3	MTIOC3A	Toggle output synchronized with PWM period (or I/O port)
	MTIOC3B	PWM output pin 1
	MTIOC3D	PWM output pin 1' (negative-phase waveform of PWM output 1)
MTU4	MTIOC4A	PWM output pin 2
	MTIOC4C	PWM output pin 2' (negative-phase waveform of PWM output 2)
	MTIOC4B	PWM output pin 3
	MTIOC4D	PWM output pin 3' (negative-phase waveform of PWM output 3)
MTU6	MTIOC6A	Toggle output synchronized with PWM period (or I/O port)
	MTIOC6B	PWM output pin 4
	MTIOC6D	PWM output pin 4' (negative-phase waveform of PWM output 4)
MTU7	MTIOC7A	PWM output pin 5
	MTIOC7C	PWM output pin 5' (negative-phase waveform of PWM output 5)
	MTIOC7B	PWM output pin 6
	MTIOC7D	PWM output pin 6' (negative-phase waveform of PWM output 6)

Table 24.72 Register Settings for Reset-Synchronized PWM Mode

Register	Setting
MTU3.TCNT	Initial setting (0000h)
MTU4.TCNT	Initial setting (0000h)
MTU3.TGRA	Set the count period for MTU3.TCNT
MTU3.TGRB	Set the transition point of the PWM waveform to be output from the MTIOC3B and MTIOC3D pins
MTU4.TGRA	Set the transition point of the PWM waveform to be output from the MTIOC4A and MTIOC4C pins
MTU4.TGRB	Set the transition point of the PWM waveform to be output from the MTIOC4B and MTIOC4D pins
MTU6.TCNT	Initial setting (0000h)
MTU7.TCNT	Initial setting (0000h)
MTU6.TGRA	Set the count period for MTU6.TCNT
MTU6.TGRB	Set the transition point of the PWM waveform to be output from the MTIOC6B and MTIOC6D pins
MTU7.TGRA	Set the transition point of the PWM waveform to be output from the MTIOC7A and MTIOC7C pins
MTU7.TGRB	Set the transition point of the PWM waveform to be output from the MTIOC7B and MTIOC7D pins

(1) Example of Procedure for Setting Reset-Synchronized PWM Mode

Figure 24.44 shows an example of procedure for setting the reset-synchronized PWM mode.



[1] Set the TSTRA.CST3 (TSTRB.CST6) and TSTRA.CST4 (TSTRB.CST7) bits to 0 to stop the TCNT count operation.
Specify the reset-synchronized PWM mode while MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT) are stopped.

[2] Set the TCR.TPSC[2:0] and TCR.CKEG[1:0] bits of MTU3 (MTU6) to select the count clock source and clock edge for MTU3 (MTU6).
Set the TCR.CCLR[2:0] bits to select TGRA compare match as the counter clear source.

[3] To control a brushless DC motor, set the TGCRA.BDC bit and set the feedback signal input source and output chopping or gate signal direct output (only when MTU3 and MTU4 are used).

[4] Set MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT) to 0000h.

[5] MTU3.TGRA (MTU6.TGRA) is the period register. Set the PWM period in MTU3.TGRA (MTU6.TGRA).
Set the PWM output transition timing in MTU3.TGRB (MTU6.TGRB), MTU4.TGRA (MTU7.TGRA), and MTU4.TGRB (MTU7.TGRB).
Note that the setting values should be within the range of compare match with MTU3.TCNT (MTU6.TCNT).

X ≤ MTU3.TGRA (X: Set value)

[6] Set the PSYE bit in the TOCR1A (TOCR1B) register to enable or disable toggle output synchronized with the PWM period, and set the OLSP and OLSN bits to select the PWM output level.
When specifying the PWM output level using the TOCR2A (TOCR2B) and TOLBRA (TOLBRB) in buffer operation, refer to Timer Output Level Buffer Registers (TOLBRA and TOLBRB).

[7] Set the MD[3:0] bits in MTU3.TMDR1 (MTU6.TMDR1) register to 1000b to select the reset-synchronized PWM mode. Specify the buffer operation (BFA = 1) of the MTU3.TGRA (MTU6.TGRA) register and the buffer operation (BFB = 1) of the MTU3.TGRB (MTU6.TGRB) register when necessary. Set the MTU4.TMDR1 (MTU7.TMDR1) register to the initial value (normal operation; BFA = 0 and BFB = 0).

[8] Set the TOERA (TOERB) register to enable or disable output from the PWM output pin. Then, set the MTU3.TIORH (MTU6.TIORH), MTU3.TIORL (MTU6.TIORL), MTU4.TIORH (MTU7.TIORH), and MTU4.TIORL (MTU7.TIORL) registers to 00h.

[9] Set the MPU and the port mode register (PMR) of the I/O ports.

[10] Set the TSTRA.CST3 (TSTRB.CST6) bit to 1 to start count operation.

Note: The output waveform starts to toggle at the point of MTU3.TCNT (MTU6.TCNT) = MTU3.TGRA (MTU6.TGRA) = X by setting X = MTU3.TGRA (MTU6.TGRA), i.e., period = duty ratio.

Figure 24.44 Procedure for Selecting Reset-Synchronized PWM Mode

(2) Example of Reset-Synchronized PWM Mode Operation

Figure 24.45 shows an example of operation in the reset-synchronized PWM mode.
MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) operate as up-counters. The counters are cleared when a compare match occurs between MTU3.TCNT (MTU6.TCNT) and MTU3.TGRA (MTU6.TGRA), and then begin incrementing from 0000h. The output from the PWM pins toggles every time a compare match occurs in MTU3.TGRB (MTU6.TGRB), MTU4.TGRA (MTU7.TGRA), and MTU4.TGRB (MTU7.TGRB) and the counters are cleared.

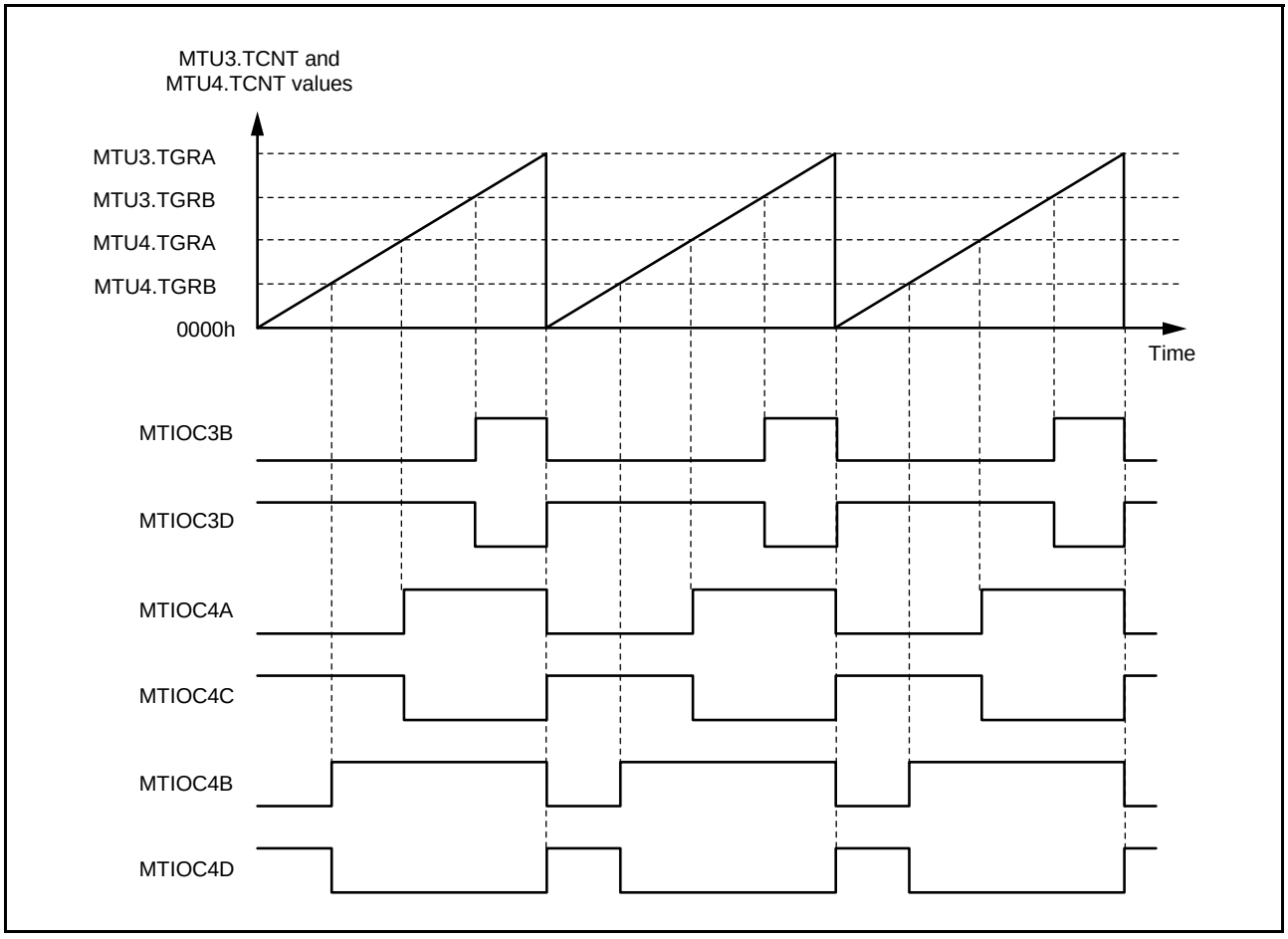


Figure 24.45 Example of Reset-Synchronized PWM Mode Operation (When OLSN = 1 and OLSP = 1 in MTU3.TOCR1 and MTU4.TOCR1)

24.3.8 Complementary PWM Mode

In complementary PWM mode, dead time can be set for PWM waveforms to be output. The dead time is the period during which the upper and lower arm transistors are set to the inactive level in order to prevent short-circuiting of the arms.

Six positive-phase and six negative-phase PWM waveforms (12 phases in total) with dead time can be output by combining MTU3/ MTU4 and MTU6/MTU7. PWM waveforms without dead time can also be output.

In complementary PWM mode, MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D, MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7B, MTIOC7C, and MTIOC7D pins function as PWM output pins, and the MTIOC3A and MTIOC6A pins can be set for toggle output synchronized with the PWM period.

MTU3.TCNT, MTU4.TCNT, MTU6.TCNT, and MTU7.TCNT function as up/down-counters.

Table 24.73 shows the PWM output pins used. Table 24.74 shows the settings of the registers used.

A function to directly cut off the PWM output by using an external signal is supported as a port function.

Table 24.73 Output Pins for Complementary PWM Mode

Channel	Output Pin	Description
MTU3	MTIOC3A	Toggle output synchronized with PWM period (or I/O port)
	MTIOC3B	PWM output pin 1
	MTIOC3C	I/O port*1
	MTIOC3D	PWM output pin 1' (negative-phase waveform output of PWM output 1)
MTU4	MTIOC4A	PWM output pin 2
	MTIOC4C	PWM output pin 2' (negative-phase waveform output of PWM output 1)
	MTIOC4B	PWM output pin 3
	MTIOC4D	PWM output pin 3' (negative-phase waveform output of PWM output 1)
MTU6	MTIOC6A	Toggle output synchronized with PWM period (or I/O port)
	MTIOC6B	PWM output pin 4
	MTIOC6C	I/O port*1
	MTIOC6D	PWM output pin 4' (negative-phase waveform output of PWM output 4)
MTU7	MTIOC7A	PWM output pin 5
	MTIOC7C	PWM output pin 5' (negative-phase waveform output of PWM output 5)
	MTIOC7B	PWM output pin 6
	MTIOC7D	PWM output pin 6' (negative-phase waveform output of PWM output 6)

Note 1. Avoid setting the MTIOC3C and MTIOC6C pins as timer I/O pins in complementary PWM mode.

Table 24.74 Register Settings for Complementary PWM Mode (1/2)

Channel	Counter/ Register	Description	Read/Write from CPU
MTU3	TCNT	Starts up-counting from the value set in the dead time register	Maskable by TRWERA setting*1
	TGRA	Set MTU3.TCNT upper limit value (1/2 carrier period + dead time)	Maskable by TRWERA setting*1
	TGRB	PWM output 1 compare register	Maskable by TRWERA setting*1
	TGRC	MTU3.TGRA buffer register	Readable/writable
	TGRD	PWM output 1/MTU3.TGRB buffer register	Readable/writable
	TGRE	MTU3.TGRB buffer register B (when double buffer function is used)	Readable/writable
MTU4	TCNT	Starts up-counting after being initialized to 0000h	Maskable by TRWERA setting*1
	TGRA	PWM output 2 compare register	Maskable by TRWERA setting*1
	TGRB	PWM output 3 compare register	Maskable by TRWERA setting*1
	TGRC	PWM output 2/MTU4.TGRA buffer register	Readable/writable
	TGRD	PWM output 3/MTU4.TGRB buffer register	Readable/writable
	TGRE	MTU4.TGRA buffer register B (when double buffer function is used)	Readable/writable
	TGRF	MTU4.TGRB buffer register B (when double buffer function is used)	Readable/writable
MTU6	TCNT	Starts up-counting from the value set in the dead time register	Maskable by TRWERB setting*2
	TGRA	Set MTU6.TCNT upper limit value (1/2 carrier period + dead time)	Maskable by TRWERB setting*2
	TGRB	PWM output 4 compare register	Maskable by TRWERB setting*2
	TGRC	MTU6.TGRA buffer register	Readable/writable
	TGRD	PWM output 4/MTU6.TGRB buffer register	Readable/writable
	TGRE	MTU6.TGRB buffer register B (when double buffer function is used)	Readable/writable
MTU7	TCNT	Starts up-counting after being initialized to 0000h	Maskable by TRWERB setting*2
	TGRA	PWM output 5 compare register	Maskable by TRWERB setting*2
	TGRB	PWM output 6 compare register	Maskable by TRWERB setting*2
	TGRC	PWM output 5/MTU7.TGRA buffer register	Readable/writable
	TGRD	PWM output 6/MTU7.TGRB buffer register	Readable/writable
	TGRE	MTU7.TGRA buffer register B (when double buffer function is used)	Readable/writable
	TGRF	MTU7.TGRB buffer register B (when double buffer function is used)	Readable/writable

Note 1. Access can be enabled or disabled according to the setting in TRWERA (timer read/write enable register A).
Note 2. Access can be enabled or disabled according to the setting in TRWERB (timer read/write enable register B).

Table 24.75 Register Settings for Complementary PWM Mode (2/2)

Channel	Counter/ Register	Description	Read/Write from CPU
Timer dead time data register A (TDDRA)		Set MTU4.TCNT and MTU3.TCNT offset value (dead time value)	Maskable by TRWERA setting*1
Timer dead time data register B (TDDRB)		Set MTU7.TCNT and MTU6.TCNT offset value (dead time value)	Maskable by TRWERB setting *2
Timer period data register A (TCDRA)		Set MTU4.TCNT upper limit value (1/2 carrier period)	Maskable by TRWERA setting*1
Timer period data register B (TCDRB)		Set MTU7.TCNT upper limit value (1/2 carrier period)	Maskable by TRWERB setting*2
Timer period buffer register A (TCBRA)		TCDRA buffer register	Readable/writable
Timer period buffer register B (TCBRB)		TCDRB buffer register	Readable/writable
Subcounter A (TCNTSA)		Subcounter A for dead time generation	Read-only
Subcounter B (TCNTSB)		Subcounter B for dead time generation	Read-only
Temporary register 1A (TEMP1A)		PWM output 1/MTU3.TGRB temporary register A	Not readable/writable
Temporary register 1B (TEMP1B)		PWM output 1/MTU3.TGRB temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 2A (TEMP2A)		PWM output 2/MTU4.TGRA temporary register A	Not readable/writable
Temporary register 2B (TEMP2B)		PWM output 2/MTU4.TGRA temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 3A (TEMP3A)		PWM output 3/MTU4.TGRB temporary register A	Not readable/writable
Temporary register 3B (TEMP3B)		PWM output 3/MTU4.TGRB temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 4A (TEMP4A)		PWM output 4/MTU6.TGRB temporary register A	Not readable/writable
Temporary register 4B (TEMP4B)		PWM output 4/MTU6.TGRB temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 5A (TEMP5A)		PWM output 5/MTU7.TGRA temporary register A	Not readable/writable
Temporary register 5B (TEMP5B)		PWM output 5/MTU7.TGRA temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 6A (TEMP6A)		PWM output 6/MTU7.TGRB temporary register A	Not readable/writable
Temporary register 6B (TEMP6B)		PWM output 6/MTU7.TGRB temporary register B (when double buffer function is used)	Not readable/writable

Note 1. Access can be enabled or disabled according to the setting in TRWERA (timer read/write enable register A).
Note 2. Access can be enabled or disabled according to the setting in TRWERB (timer read/write enable register B).

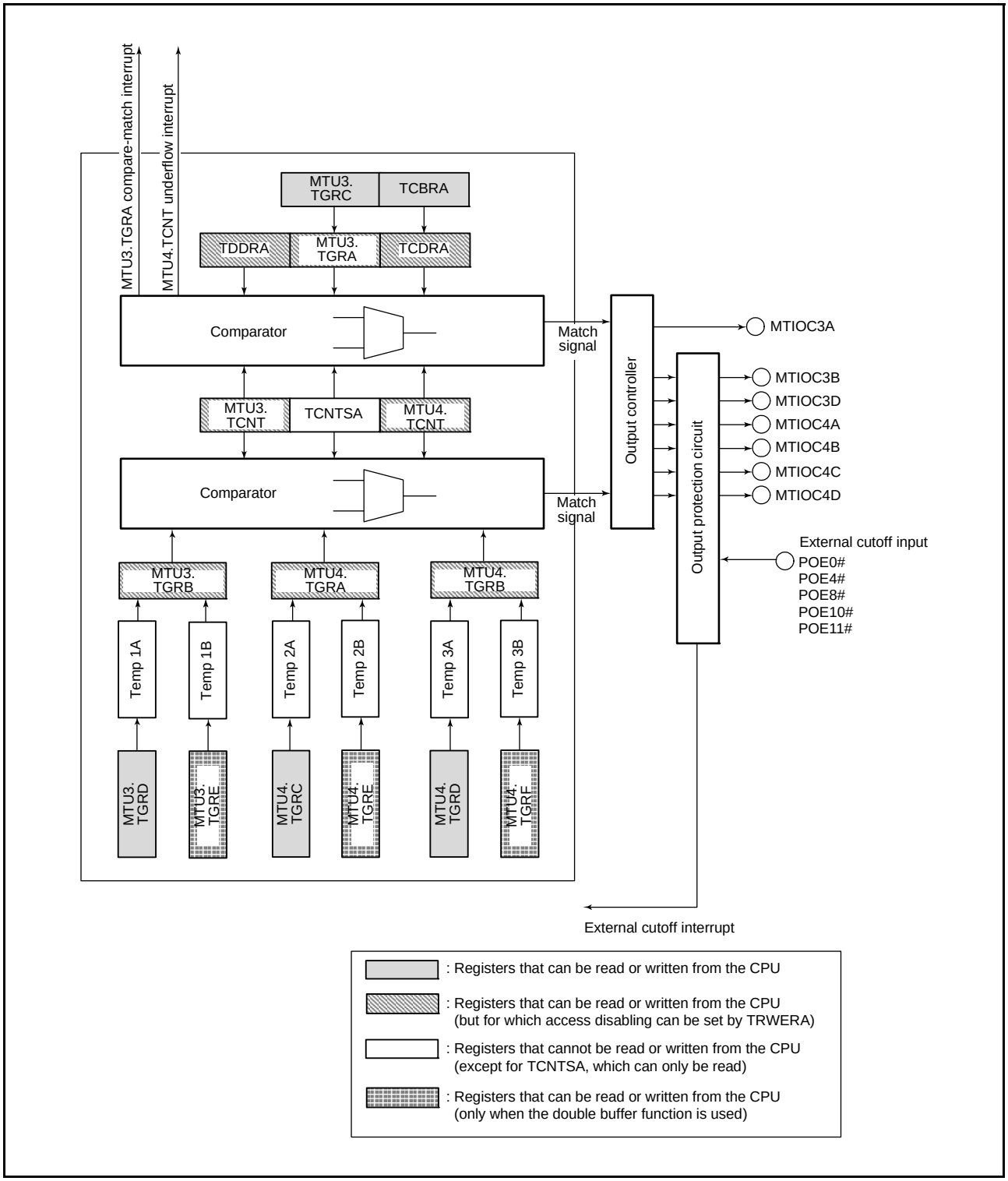


Figure 24.46 Block Diagram of MTU3 and MTU4 in Complementary PWM Mode

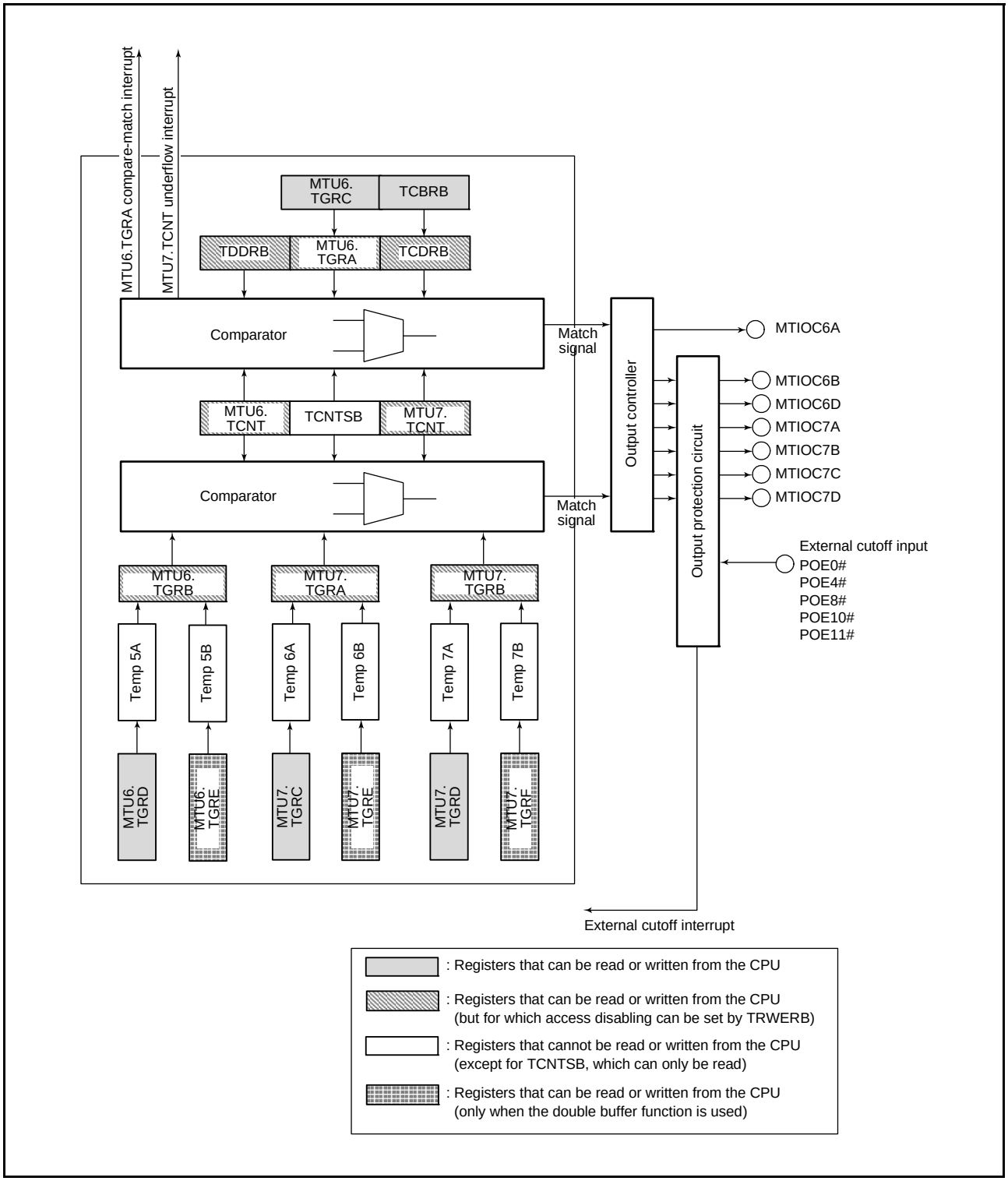
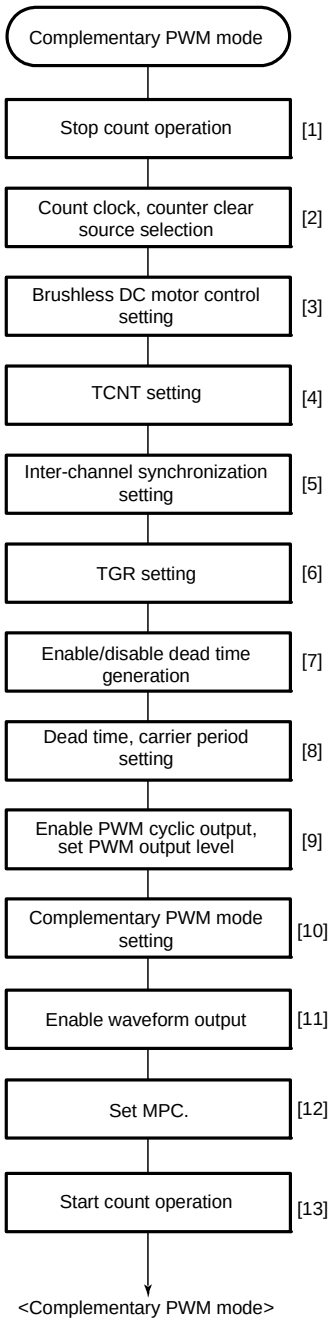


Figure 24.47 Block Diagram of MTU6 and MTU7 in Complementary PWM Mode

(1) Example of Complementary PWM Mode Setting Procedure

Figure 24.48 shows an example of the complementary PWM mode setting procedure.



[1] Set the TSTRA.CST3 (TSTRB.CST6) and TSTRA.CST4 (TSTRB.CST7) bits to 0 to stop the TCNT count operation.
Specify complementary PWM mode while MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT) are stopped.

[2] Set the TCR.TPSC[2:0] and TCR.CKEG[1:0] bits of MTU3 and MTU4 (MTU6 and MTU7) to select the same count clock source and clock edge for MTU3 and MTU4 (MTU6 and MTU7).
Set the TCR.CCLR[2:0] bits only when restarting by synchronous clearing with another channel during complementary PWM operation.

[3] To control a brushless DC motor, set the TGCRA.BDC bit and set the feedback signal input source and output chopping or gate signal direct output (only when MTU3 and MTU4 are used).

[4] Set the dead time in MTU3.TCNT (MTU6.TCNT). Set MTU4.TCNT (MTU7.TCNT) to 0000h.

[5] Make this setting only when restarting by synchronous clearing with another channel during complementary PWM mode operation. In this case, set the TSYRA register (TSYRB) to synchronize the channel generating the synchronous clear with MTU3 and MTU4 (MTU6 and MTU7).

[6] When MTU3 and MTU6 are used, set the PWM output duty ratio in the duty registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB) and buffer registers (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD).
Be sure to set the same value in each pair of the duty registers (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD) and buffer registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB).
Only when the double buffer function is used, set the PWM output duty value – 1 in the buffer registers (MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF).
When MTU6 and MTU7 are used, set the value to the following registers:
Duty registers (MTU6.TGRB, MTU7.TGRA, and MTU7.TGRB)
Buffer registers (MTU6.TGRD, MTU7.TGRC, and MTU7.TGRD)
Buffer registers when the double buffer function is used (MTU6.TGRE, MTU7.TGRE, and MTU7.TGRF)

[7] This setting is necessary only when no dead time should be generated. Make appropriate settings in the TDERA (TDERB) register so that no dead time generated.

[8] Set the dead time in the TDDRA (TDDRB) register, 1/2 the PWM period in the TCDRA (TCDRB) register and TCBRA (TCBRB) register, and 1/2 the PWM period plus the dead time in MTU3.TGRA (MTU6.TGRA) and MTU3.TGRC (MTU6.TGRC).
When no dead time generation is selected, set 1 in the TDDRA (TDDRB) register and 1/2 the PWM period + 1 in MTU3.TGRA (MTU6.TGRA) and MTU3.TGRC (MTU6.TGRC).

[9] Set the PSYE bit in the TOCR1A (TOCR1B) register to enable or disable toggle output synchronized with the PWM period, and set the OLSP and OLSN bits to select the PWM output level.
When specifying the PWM output level using the TOCR2A (TOCR2B) and TOLBRA (TOLBRA) in buffer operation, refer to Timer Output Level Buffer Registers (TOLBRA and TOLBRB).

[10] Select complementary PWM mode in MTU3.TMDR1 (MTU6.TMDR1). Specify the buffer operation (BFA = 1) of MTU3.TGRA (MTU6.TGRA) and the buffer operation (BFB = 1) of MTU3.TGRB (MTU6.TGRB).
Only when the double buffer function is used, set the TMDR2A (TMDR2B) register to enable the double buffer function.

[11] Set the TOERA (TOERB) register to enable or disable output from the PWM output pin. Then, set the MTU3.TIORH (MTU6.TIORH), MTU3.TIORL (MTU6.TIORL), MTU4.TIORH (MTU7.TIORH), and MTU4.TIORL (MTU7.TIORL) to 00h.

[12] Set the MPU and the port mode register (PMR) of the I/O ports.

[13] Set the TSTRA.CST3 (TSTRB.CST6) and TSTRA.CST4 (TSTRB.CST7) bits to 1 simultaneously to start count operation.

Figure 24.48 Example of Complementary PWM Mode Setting Procedure

(2) Outline of Complementary PWM Mode Operation

In complementary PWM mode, six phases (three positive and three negative) PWM waveforms can be output. Figure 24.49 illustrates counter operation in complementary PWM mode (MTU3 and MTU4), and Figure 24.50 shows an example of operation in complementary PWM mode.

(a) Counter Operation

In complementary PWM mode, three counters—MTU3.TCNT, MTU4.TCNT, and TCNTSA (MTU6.TCNT, MTU7.TCNT, and TCNTSB) —in each unit perform up-/down-count operations.
MTU3.TCNT (MTU6.TCNT) is automatically initialized to the value set in TDDRA (TDDRB) when complementary PWM mode is selected and the CST3 bit in TSTRA (TSTRB) is 0. When the CST3 bit is set to 1, MTU3.TCNT (MTU6.TCNT) counts up to the value set in MTU3.TGRA (MTU6.TGRA), then switches to down-counting when it matches MTU3.TGRA (MTU6.TGRA). When the MTU4.TCNT (MTU7.TCNT) value matches 0000h, MTU3.TCNT (MTU6.TCNT) switches to up-counting, and the operation is repeated in this way.
MTU4.TCNT (MTU7.TCNT) should be initialized to 0000h after a reset. When the CST4 bit is set to 1, MTU4.TCNT (MTU7.TCNT) counts up in synchronization with MTU3.TCNT (MTU6.TCNT), and switches to down-counting when MTU3.TCNT (MTU6.TCNT) matches MTU3.TGRA (MTU6.TGRA). On reaching 0000h, MTU4.TCNT (MTU7.TCNT) switches to up-counting, and the operation is repeated in this way. TCNTSA (TCNTSB) is a read-only counter. It does not need to be initialized after a reset.
In counting up by MTU3.TCNT and MTU4.TCNT (or MTU6.TCNT and MTU7.TCNT), MTU3.TCNT (or MTU6.TCNT) starts counting up when it matches TCDRA (or TCDRB) and switches to counting down when it matches MTU3.TGRA (or MTU6.TGRA). Furthermore, when MTU4.TCNT (or MTU7.TCNT) matches TDDRA (or TDDRB), TCNTSA (or TCNTSB) is set to the value in MTU3.TGRA (or MTU6.TGRA) and counting is stopped.
When MTU4.TCNT (MTU7.TCNT) matches TDDRA (TDDRB) during down-counting of MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT), TCNTSA (TCNTSB) starts up-counting, and when MTU4.TCNT (MTU7.TCNT) matches 0000h, the operation switches to down-counting. When MTU3.TCNT (MTU6.TCNT) matches TCDRA (TCDRB), TCNTSA (TCNTSB) becomes 0000h and stops counting.
TCNTSA (TCNTSB) is compared with the compare register and temporary register, in which the PWM duty is specified, only during the count operation.

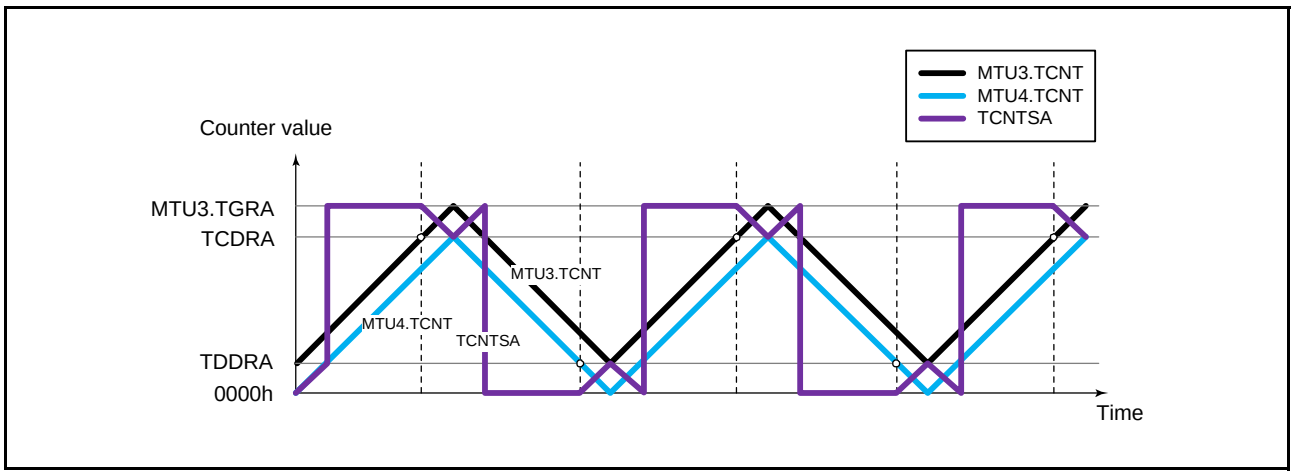


Figure 24.49 Count Operation in Complementary PWM Mode (MTU3 and MTU4)

(b) Register Operation

In complementary PWM mode, nine registers (compare registers, buffer registers, and temporary registers) are used to control the duty ratio for the PWM output. Figure 24.50 shows an example of operation in complementary PWM mode (MTU3 and MTU4).

MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB (MTU6.TGRB, MTU7.TGRA, and MTU7.TGRB) are constantly compared with the counters to generate PWM waveforms. When these registers match the counter, the value set in the OLSN and OLSP bits in the timer output control registers (TOCR1A and TOCR1B) is output from the PWM output pin.

MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD (MTU6.TGRD, MTU7.TGRC, and MTU7.TGRD) are buffer registers for these compare registers.

When the double buffer function is used, MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF (MTU6.TGRE, MTU7.TGRE, and MTU7.TGRF) are also used as buffer registers B. For details of double buffer operation, refer to section 24.3.8 (2) (s), Double Buffer Function in Complementary PWM Mode.

Data in a compare register can be changed by writing new data to the corresponding buffer register. The buffer registers can be read or written at any time.

When modifying data in a buffer register, write to MTU4.TGRD (MTU7.TGRD) last and enable data transfer from the buffer register to a temporary register. At this time, transfer from the TCBRA (TCBRB) register and MTU3.TGRC (MTU6.TGRC) register, which operate as buffer registers for the timer period registers, to temporary registers is also enabled. Data is transferred to all five temporary registers at the same time.

When transfer is enabled in the Ta interval, data written to a buffer register is transferred to the temporary register. The data is not transferred to the temporary register in the Tb1 and Tb2 intervals. Data enabled for transfer in this interval is transferred to the temporary register at the end of this interval.

The value transferred to a temporary register is transferred to the compare register at the end of the Tb1 interval (when matches MTU3.TGRA (MTU6.TGRA) while TCNTSA (TCNTSB) is counting up), or at the end of the Tb2 interval (when matches 0000h while TCNTSA (TCNTSB) is counting down). The timing for transfer from the temporary register to the compare register can be selected with bits MD[3:0] in the timer mode register 1 (TMDR1). Figure 24.50 shows an example in which the trough is selected for the transfer timing.

In the Tb interval in which data is not transferred to the temporary register (Tb1 in Figure 24.50), the temporary register has the same function as the compare register and is compared with the counter. In this interval, therefore, there are two compare match registers for one output phase; the compare register contains the pre-change data and the temporary register contains new data. In this interval, three counters MTU3.TCNT, MTU4.TCNT, and TCNTSA (MTU6.TCNT, MTU7.TCNT, and TCNTSB) and two registers (compare register and temporary register) are compared, and PWM output is controlled accordingly.

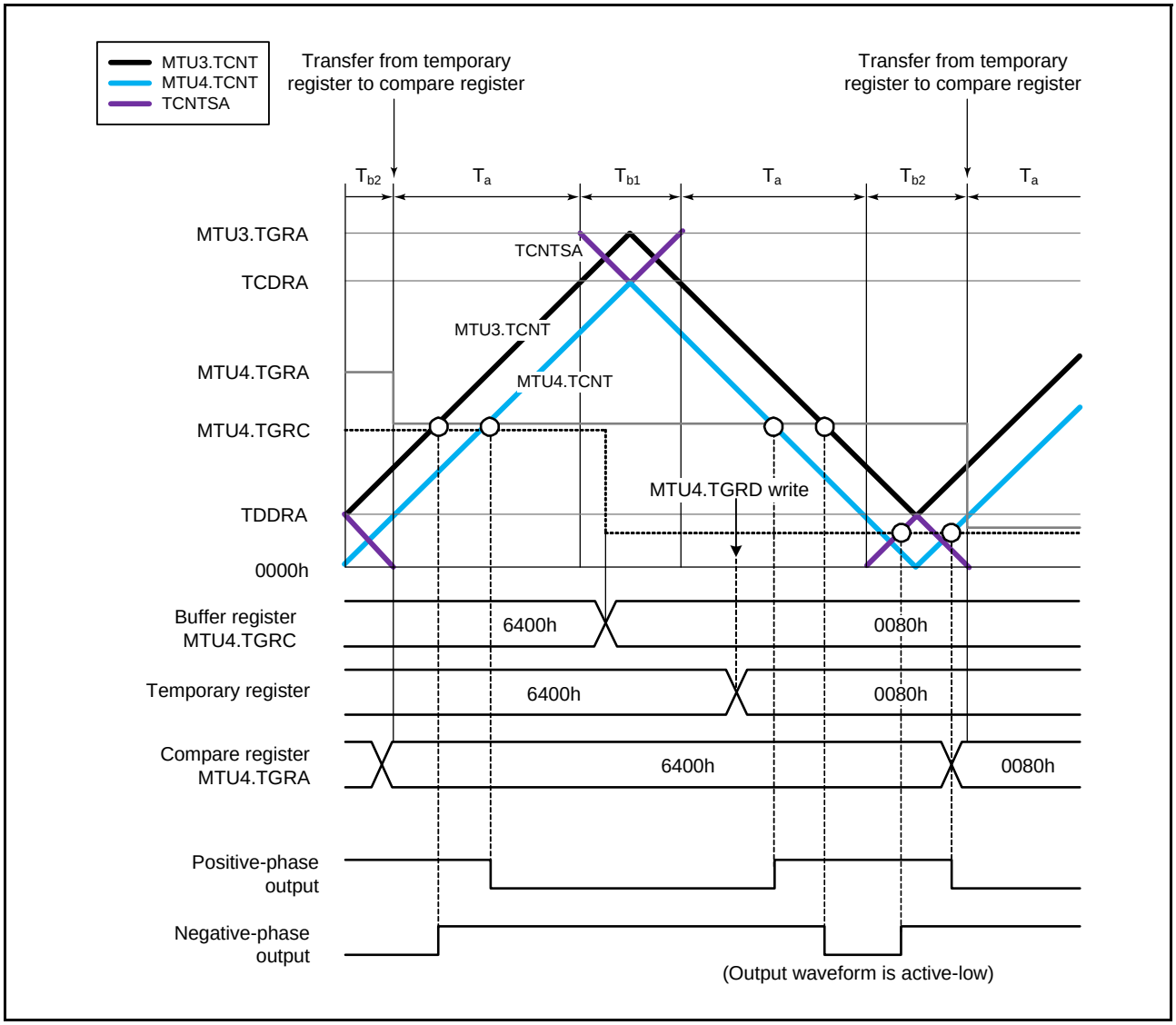


Figure 24.50 Example of Operation in Complementary PWM Mode (MTU3 and MTU4)

(c) Initial Setting

In complementary PWM mode, there are nine registers that require initial setting. In addition, there is a register that specifies whether to generate dead time (it should be used only when dead time generation should be disabled).
Before setting complementary PWM mode with MTU3.TMDR1.MD[3:0] (MTU6.TMDR1.MD[3:0]) bits, initial values should be set in the following registers.
The TOCR1A, TOCR2A, TOCR1B, and TOCR2B registers are used to set the PWM output level. MTU3.TGRC (MTU6.TGRC) operates as the buffer register for MTU3.TGRA (MTU6.TGRA), and should be set with 1/2 the PWM period + dead time Td. The timer period buffer register (TCBRA or TCBRB) operates as the buffer register for the timer period data register (TCDRA or TCDRB), and should be set with 1/2 the PWM period. Set dead time Td in the timer dead time data register (TDDRA or TDDRB).
When dead time is not needed, the TDER bit in the timer dead time enable register (TDERA or TDERB) should be set to 0, MTU3.TGRC and MTU3.TGRA (MTU6.TGRC and MTU6.TGRA) should be set to 1/2 the PWM carrier period + 1, and TDDRA (TDDRB) should be set to 1.
Set the respective initial PWM duty values in three buffer registers A (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD (MTU6.TGRD, MTU7.TGRC, and MTU7.TGRD)).
Set the respective (initial PWM duty – 1) values in three buffer registers B (MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF (MTU6.TGRE, MTU7.TGRE, and MTU7.TGRF)) only when the double buffer function is used.
The values set in the five buffer registers excluding TDDRA (TDDRB) are transferred to the corresponding compare registers as soon as complementary PWM mode is set.
Set MTU4.TCNT (MTU7.TCNT) to 0000h before setting complementary PWM mode.

Table 24.76 Registers and Counters Requiring Initial Setting

Register and Counter	Setting
TOCR1A, TOCR2A, TOCR1B, TOCR2B	PWM output level
MTU3.TGRC MTU6.TGRC	1/2 PWM period + dead time Td (1/2 PWM period + 1 when dead time generation is disabled by TDERA or TDERB)
TDDRA, TDDRB	Dead time Td (1 when dead time generation is disabled by TDERA or TDERB)
TCBRA, TCBRB	1/2 PWM period
MTU3.TGRD, MTU4.TGRC, MTU4.TGRD MTU6.TGRD, MTU7.TGRC, MTU7.TGRD	Initial PWM duty ratio value for each phase
MTU3.TGRE, MTU4.TGRE, MTU4.TGRF MTU6.TGRE, MTU7.TGRE, MTU7.TGRF	Initial PWM duty ratio – 1 value for each phase (only when double buffer function is used)
MTU4.TCNT MTU7.TCNT	0000h

Note: The value set in MTU3.TGRC (MTU6.TGRC) should be the sum of 1/2 the PWM period set in TCBRA (TCBRB) and dead time Td set in TDDRA (TDDRB). When dead time generation is disabled by TDERA (TDERB), TGRC should be set to 1/2 the PWM period + 1.

(d) PWM Output Level Setting

In complementary PWM mode, the PWM output level is set with bits OLSN and OLSP in timer output control register 1 (TOCR1A or TOCR1B) or bits OLS1P to OLS3P and OLS1N to OLS3N in timer output control register 2 (TOCR2A or TOCR2B).
The output level can be set for each of the three positive phases and three negative phases of 6-phase output.
Complementary PWM mode should be cleared before setting or changing output levels.

(e) Dead Time Setting

In complementary PWM mode, dead time can be set for PWM output.
The dead time is set in the timer dead time data register (TDDRA or TDDRB). The value set in TDDRA (TDDRB) is used as the MTU3.TCNT (MTU6.TCNT) counter start value and creates a non-overlapping interval between MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT). Complementary PWM mode should be cleared before changing the contents of TDDRA (TDDRB).

(f) Dead Time Suppressing

Dead time generation is suppressed by setting the TDER bit in the timer dead time enable register (TDERA or TDERB) to 0. TDERA (TDERB) can be set to 0 only when 0 is written to it after reading TDER = 1.
MTU3.TGRA and MTU3.TGRC (MTU6.TGRA and MTU6.TGRC) should be set to 1/2 PWM period + 1 and the timer dead time data register (TDDRA or TDDRB) should be set to 1.
By the above settings, PWM waveforms without dead time can be obtained. Figure 24.51 shows an example of operation without dead time (MTU3 and MTU4).

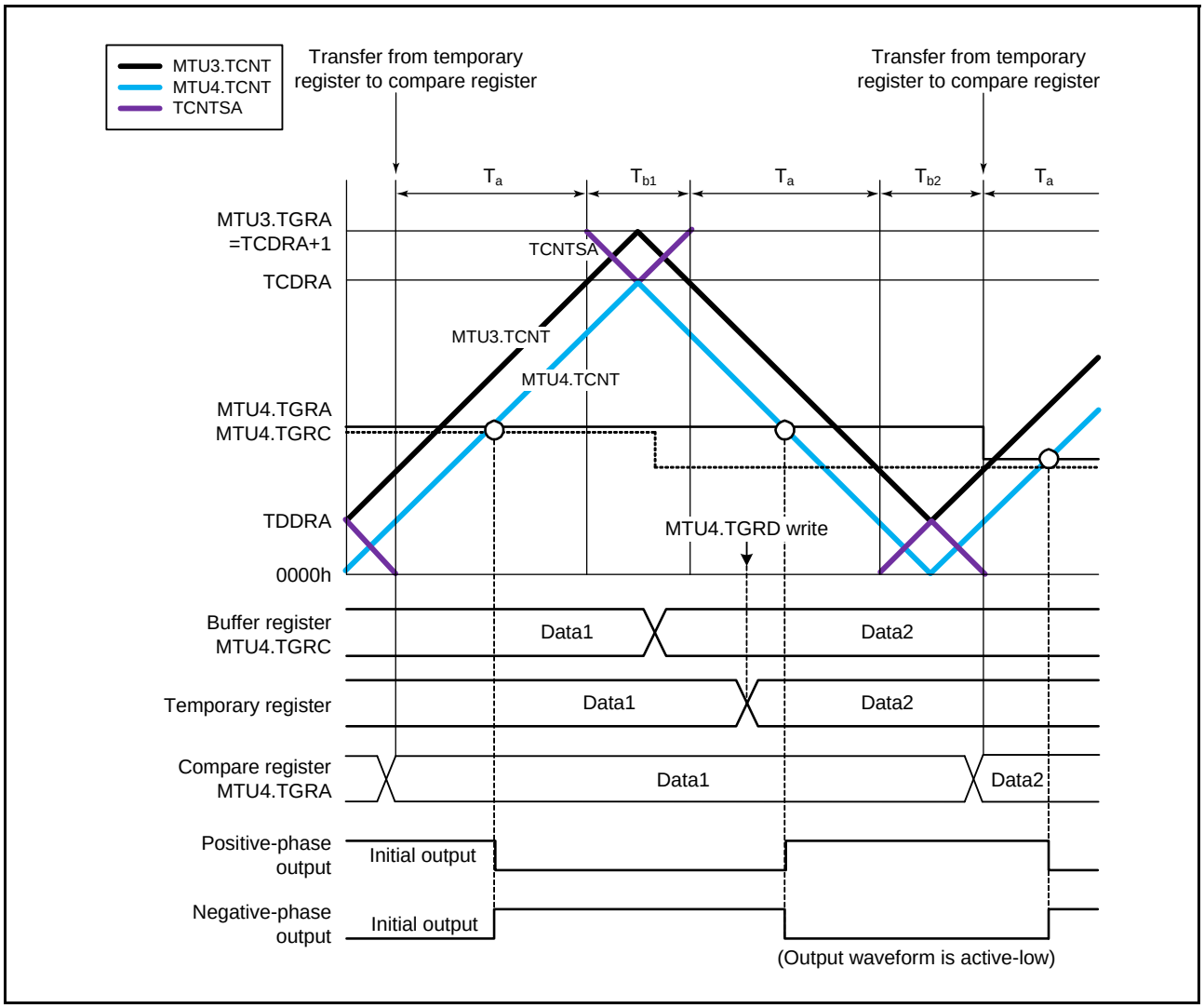


Figure 24.51 Example of Operation without Dead Time (MTU3 and MTU4)

(g) PWM Period Setting

In complementary PWM mode, the PWM period is set in two registers—MTU3.TGRA (MTU6.TGRA), in which the MTU3.TCNT (MTU6.TCNT) upper limit value is set, and TCDRA (TCDRB), in which the MTU4.TCNT (MTU7.TCNT) upper limit value is set. The settings should be made so as to achieve the following relationship between these two registers:

With dead time: MTU3.TGRA (MTU6.TGRA) setting = TCDRA (TCDRB) setting + TDDRA (TDDRB) setting
Without dead time: MTU3.TGRA (MTU6.TGRA) setting = TCDRA (TCDRB) setting + 1

In addition, the settings should be made so as to achieve the following relationship between the TCDRA (TCDRB) register and the TDDRA (TDDRB) register:

TCDRA (TCDRB) setting > TDDRA (TDDRB) setting × 2 + 2

The MTU3.TGRA and TCDRA (MTU6.TGRA and TCDRB) settings are made by setting values in buffer registers MTU3.TGRC and TCBRA (MTU6.TGRC and TCBRB). When data is written to MTU4.TGRD (MTU7.TGRD) to enable transfers, the values set in MTU3.TGRC and TCBRA (MTU6.TGRC and TCBRB) are transferred simultaneously to the MTU3.TGRA and TCDRA (MTU6.TGRA and TCDRB) with the transfer timing selected with the MTU3.TMDR1.MD[3:0] (MTU6.TMDR1.MD[3:0]) bits.
The new PWM period is reflected from the next cycle when data is updated at the crest, or from the current cycle when updated in the trough. Figure 24.52 illustrates the operation when the PWM period is updated at the crest.
Refer to the following section, (h), Register Data Updating, for the method of updating the data in each buffer register.

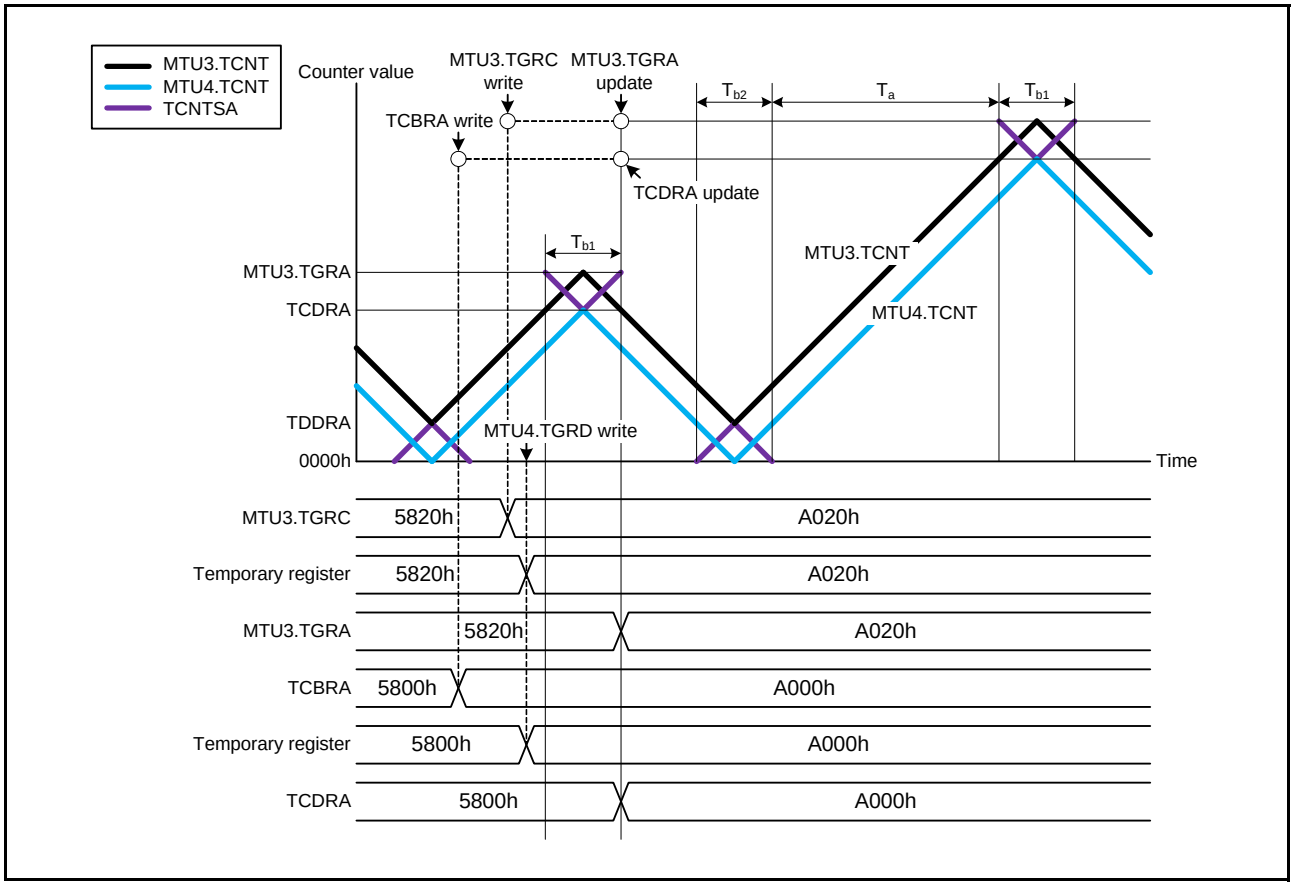


Figure 24.52 Example of PWM Period Updating (MTU3 and MTU4)

(h) Register Data Updating

The buffer registers are used to update the data in five compare registers for the PWM duty and PWM period in complementary PWM mode. The update data can be written to the buffer register at any time.

There is a temporary register between each of these registers and its buffer register. While subcounter TCNTSA (TCNTSB) is not counting, if buffer register data is updated, the temporary register value also changes. Data is not transferred from buffer registers to temporary registers while TCNTSA (TCNTSB) is counting; in this case, the value written to a buffer register is transferred after TCNTSA (TCNTSB) halts.

The temporary register value is transferred to the compare register at the data update timing set with MTU3.TMDR1.MD[3:0] (MTU6.TMDR1.MD[3:0]) bits. Figure 24.53 shows an example of data updating in complementary PWM mode (MTU3 and MTU4). This example shows the mode in which data is updated at both the counter crest and trough.

When updating buffer register data, be sure to write to MTU4.TGRD (MTU7.TGRD) at the end of the update. Data is transferred from buffer registers to the temporary registers simultaneously for all five registers after the write to MTU4.TGRD (MTU7.TGRD).

Even when not updating all five registers or when not updating the MTU4.TGRD (MTU7.TGRD) data, be sure to write to MTU4.TGRD (MTU7.TGRD) after writing data to the registers to be updated. In this case, the data written to MTU4.TGRD (MTU7.TGRD) should be the same as the data prior to the write operation.

Refer to section 24.3.8 (2) (s), Double Buffer Function in Complementary PWM Mode, for data updating when the double buffer function is used.

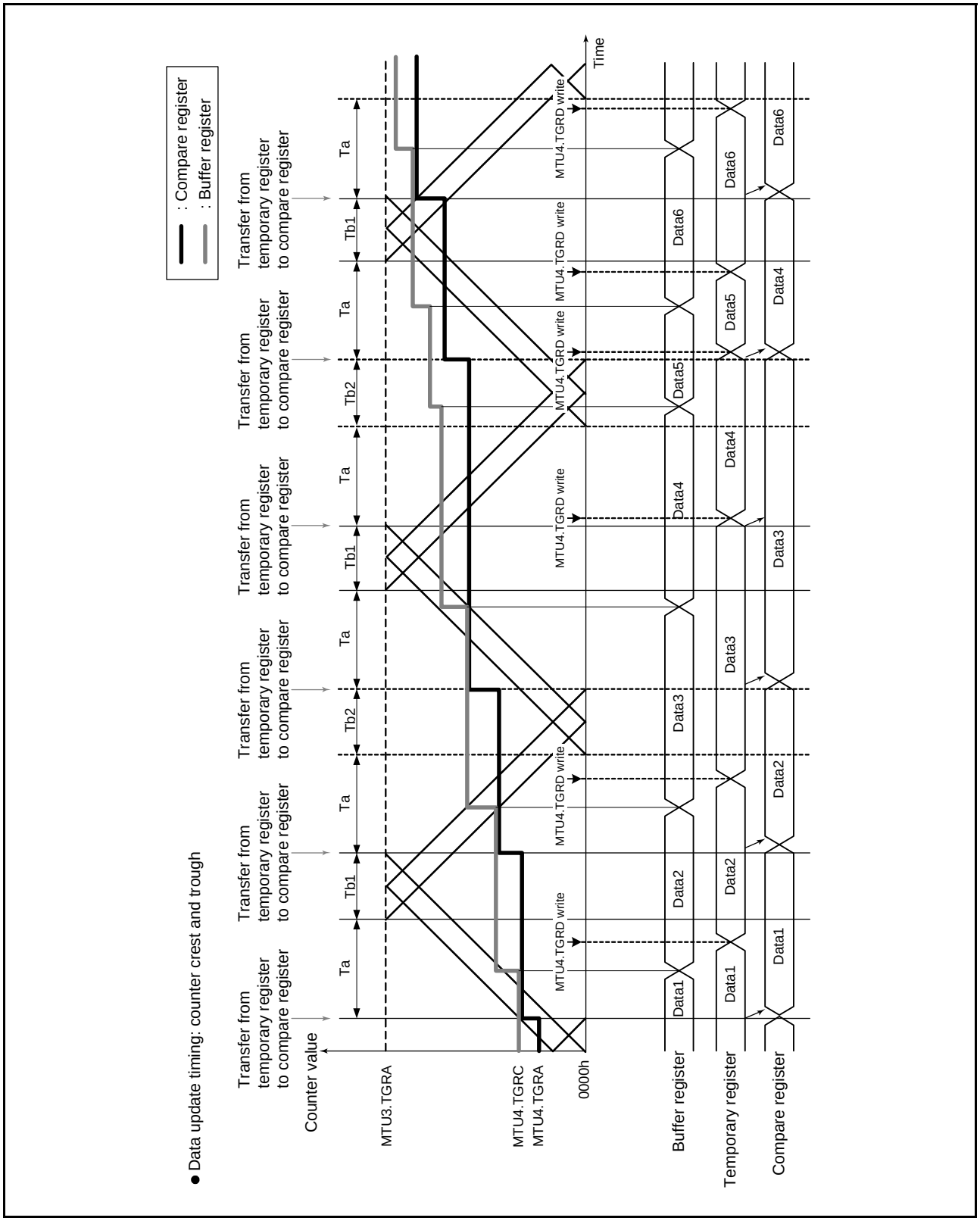


Figure 24.53 Example of Data Updating in Complementary PWM Mode (MTU3 and MTU4)

(i) Initial Output in Complementary PWM Mode

In complementary PWM mode, the initial output is determined by the setting of the OLSN and OLSP bits in the TOCR1A (TOCR1B) register or the OLS1N to OLS3N and OLS1P to OLS3P bits in the TOCR2A register (TOCR2B). This initial output is the non-active level of the PWM output and continues from when complementary PWM mode is set with the MTU3.TMDR1 (MTU6.TMDR1) until MTU4.TCNT (MTU7.TCNT) exceeds the value set in the TDDRA (TDDRB) register. Figure 24.54 shows an example of the initial output in complementary PWM mode.
An example of the waveform when the initial PWM duty ratio value is smaller than the TDDRA (TDDRB) value is shown in Figure 24.55.

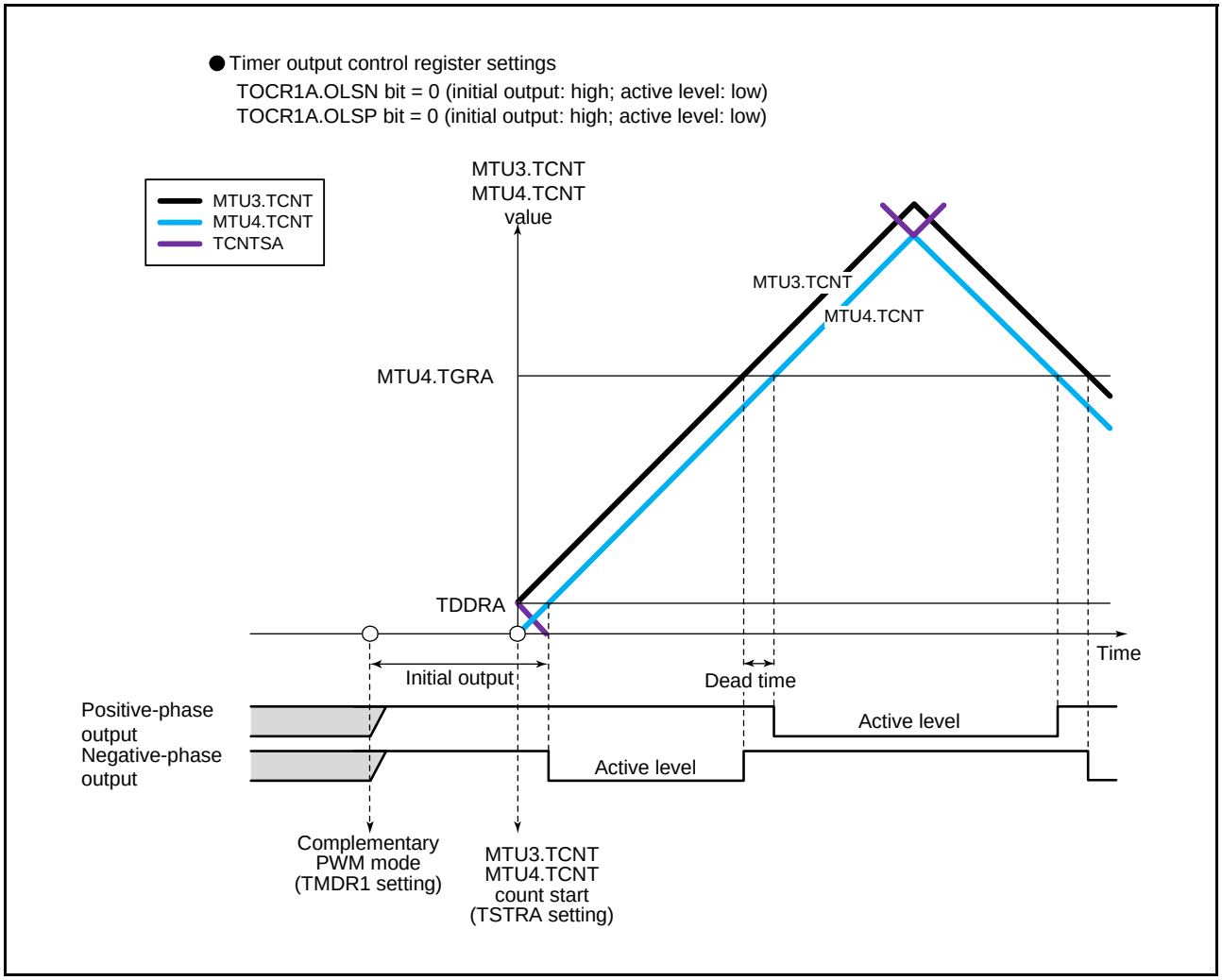


Figure 24.54 Example of Initial Output in Complementary PWM Mode (MTU3 and MTU4) (1))

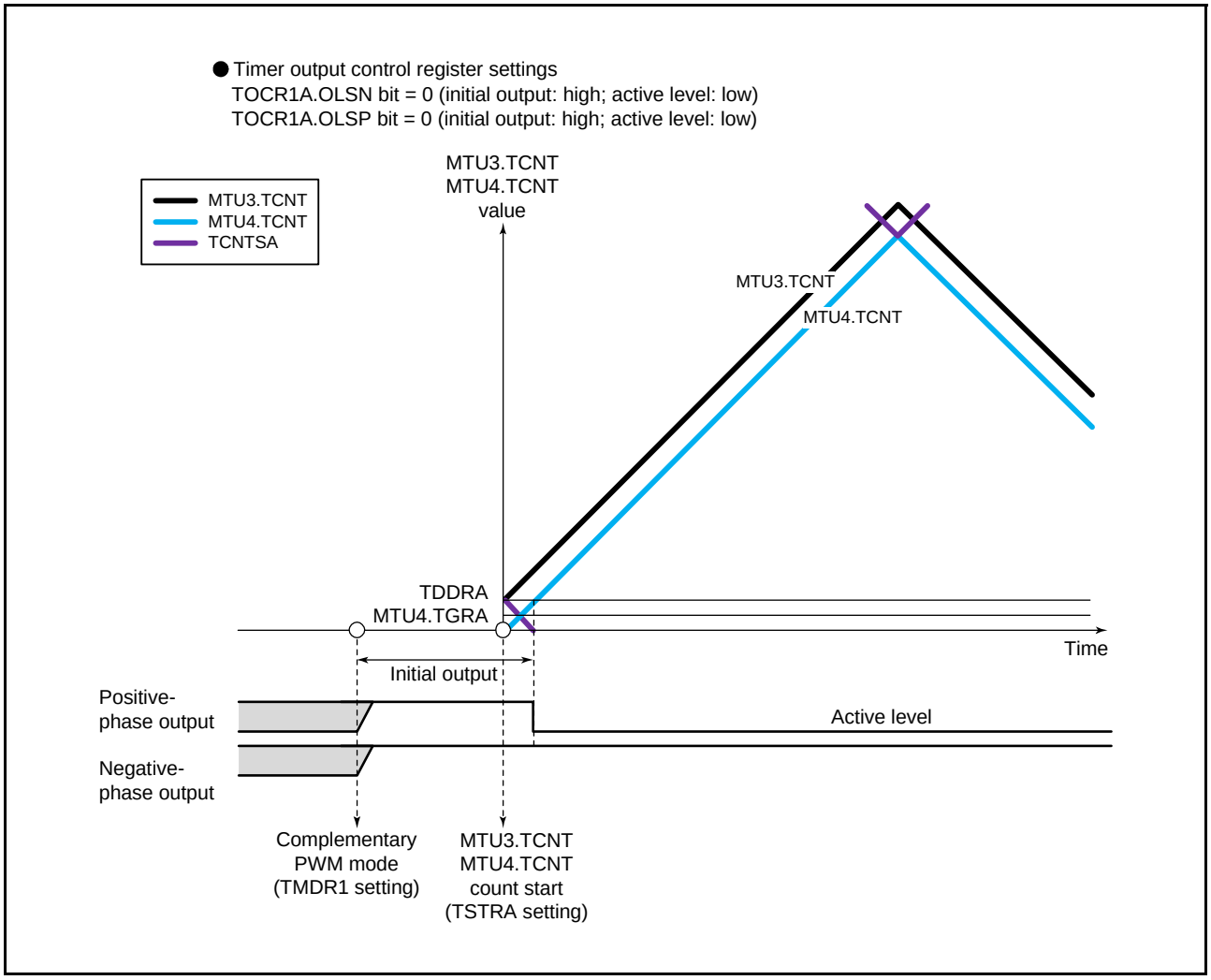


Figure 24.55 Example of Initial Output in Complementary PWM Mode (MTU3 and MTU4) (2)

(j) Method for Generating PWM Output in Complementary PWM Mode

In complementary PWM mode, six phases (three positive and three negative) PWM waveforms can be output. Dead time can be set for PWM waveforms to be output.

A PWM waveform is generated by output of the level selected in the timer output control register in the event of a compare match between a counter and a compare register. While TCNTSA (TCNTSB) is counting, the compare register and temporary register values are simultaneously compared to create consecutive PWM output from 0 to 100% duty ratio. The relative timing of turn-on and turn-off compare match occurrence may vary, but the compare match that turns off each phase takes precedence to secure the dead time and ensure that the positive-phase and negative-phase turn-on times do not overlap. Figure 24.56 to Figure 24.58 show examples of waveform generation in complementary PWM mode.

The positive-phase and negative-phase turn-off timing is generated by a compare match with the counter indicated by a solid line, and the turn-on timing is generated by a compare match with the counter indicated by a dotted line, which operates with a delay equal to the dead time behind the counter indicated by a solid line. In the T1 period, compare match a that turns off the negative phase has the highest priority, and compare matches before a are ignored. In the T2 period, compare match c that turns off the positive phase has the highest priority, and compare matches before c are ignored.

In most cases, compare matches occur in the order a → b → c → d (or c → d → a' → b') as shown in Figure 24.56.

If compare matches deviate from the a → b → c → d order, since the time for which the negative phase is off is shorter than twice the dead time, the positive phase is not turned on. If compare matches deviate from the c → d → a' → b' order, since the time for which the positive phase is off is shorter than twice the dead time, the negative phase is not turned on.

As shown in Figure 24.57, if compare match c follows compare match a before compare match b, compare match b is ignored and the negative phase is turned on by compare match d. This is because turning off the positive phase has priority due to the occurrence of compare match c (positive-phase off timing) before compare match b (positive-phase on timing) (consequently, the waveform does not change because the positive phase goes from off to off).

Similarly, in the example in Figure 24.58, turning off the negative phase has priority due to the occurrence of compare match a' (negative-phase off timing) before compare match d (negative-phase on timing). As a result, the negative phase is not turned on.

Thus, in complementary PWM mode, compare matches at turn-off timings take precedence, and turn-on timing compare matches that occur before a turn-off timing compare match are ignored.

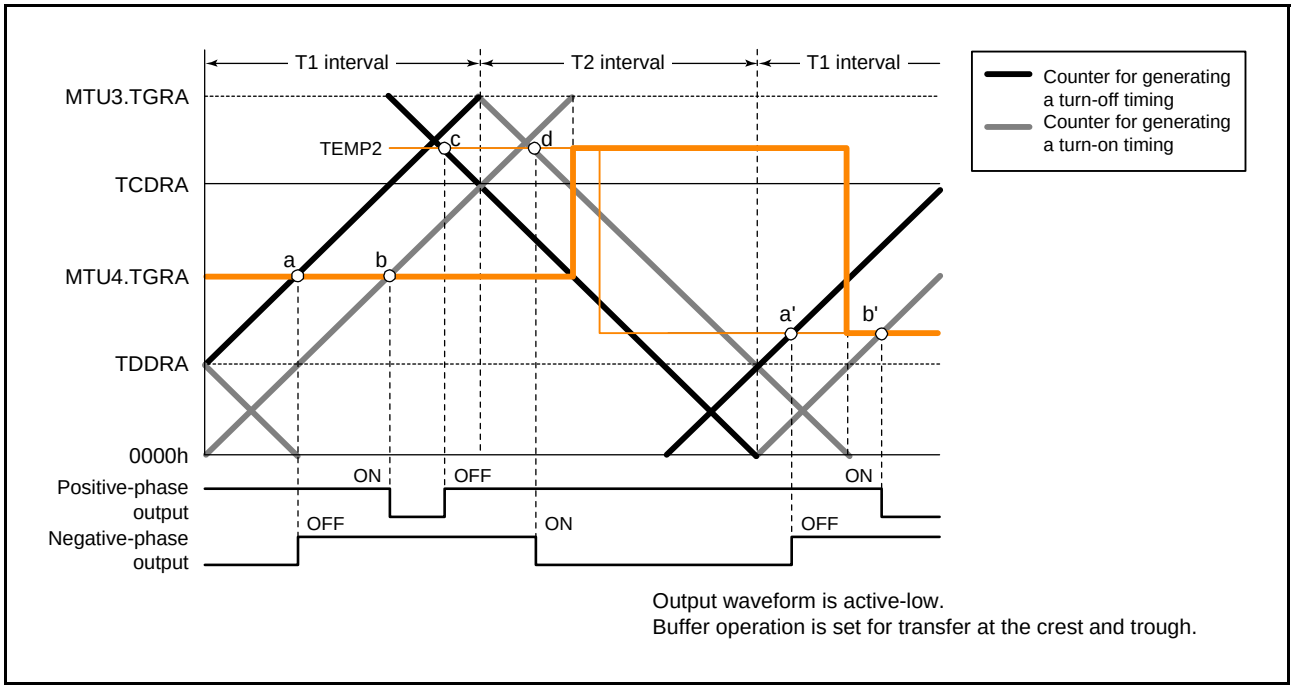


Figure 24.56 Example of Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (1)

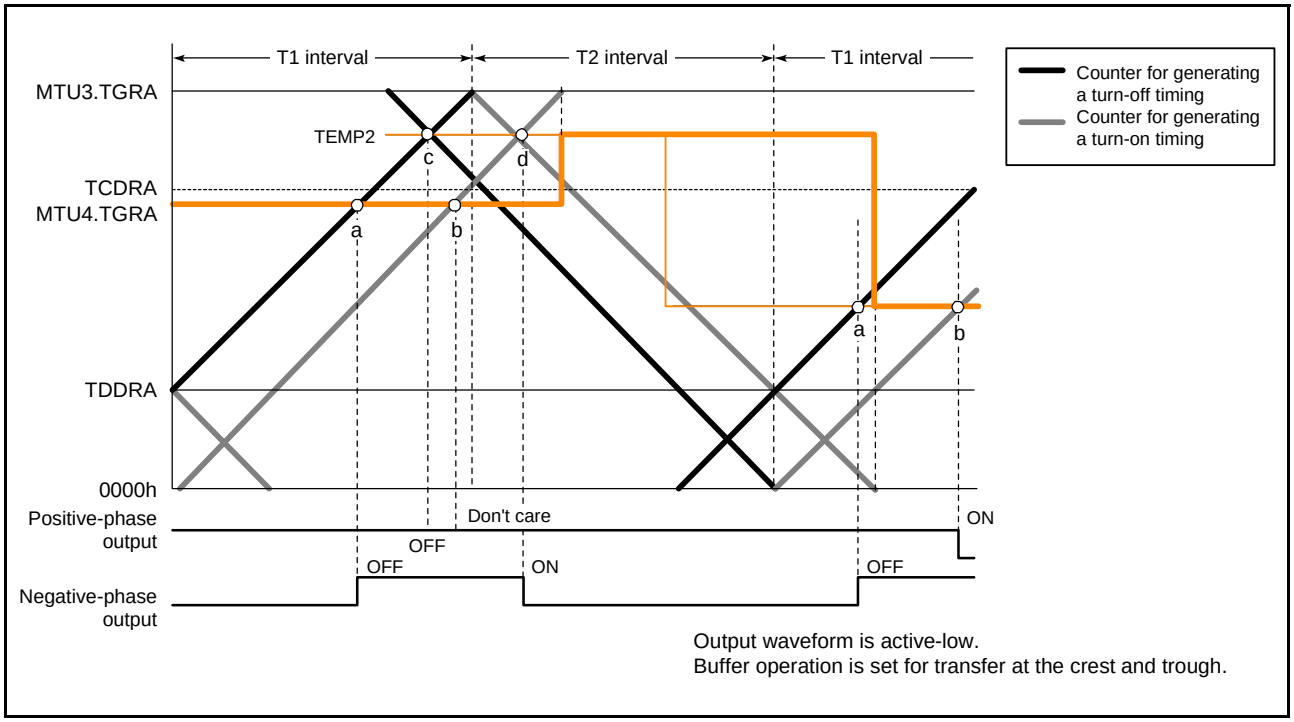


Figure 24.57 Example of Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (2)

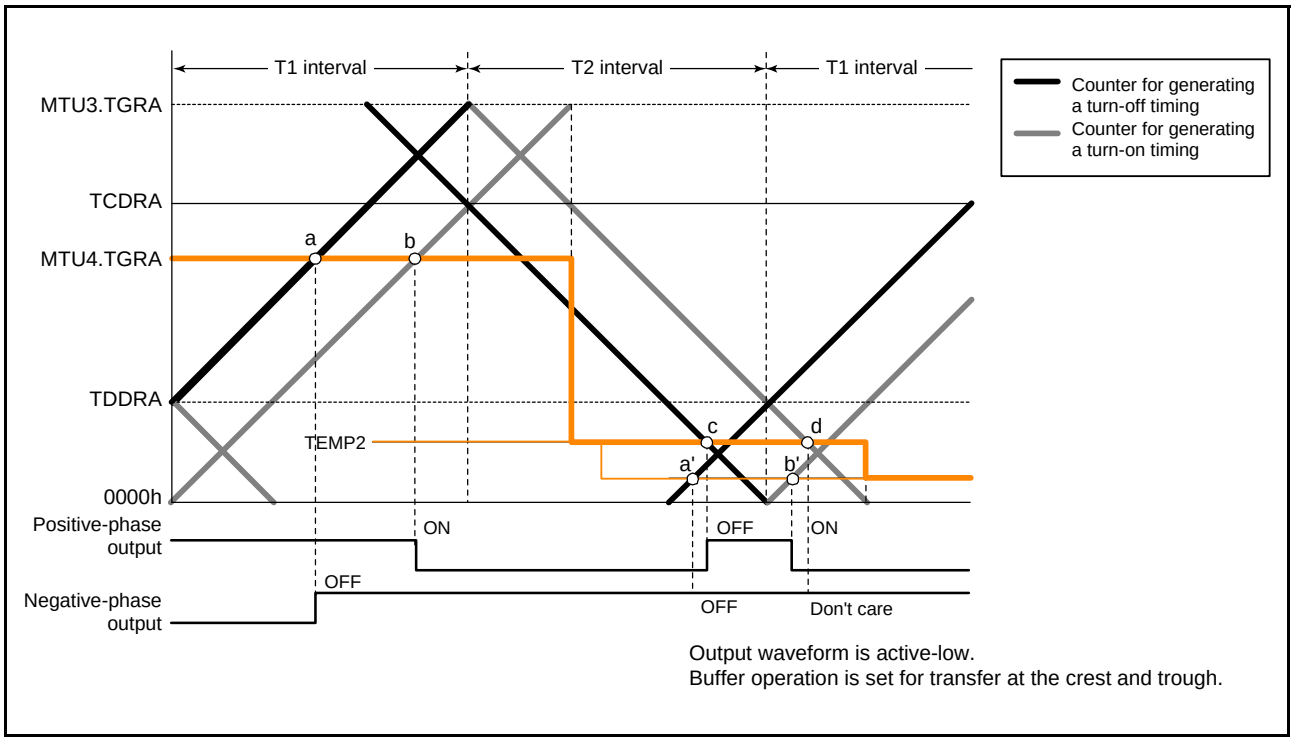


Figure 24.58 Example of Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (3)

(k) 0% and 100% Duty Ratio Output in Complementary PWM Mode

In complementary PWM mode, 0% and 100% duty PWM output can be output as required. Figure 24.59 to Figure 24.63 show output examples.

A 100% duty waveform is output when the compare register value is set to 0000h. The waveform in this case has a positive phase with a 100% on-state. A 0% duty waveform is output when the compare register value is set to the same value as MTU3.TGRA (MTU6.TGRA). The waveform in this case has a positive phase with a 100% off-state.

Turn-on and turn-off compare matches occur simultaneously, but if a turn-on compare match and turn-off compare match for the same phase occur simultaneously, both compare matches are ignored and the waveform does not change.

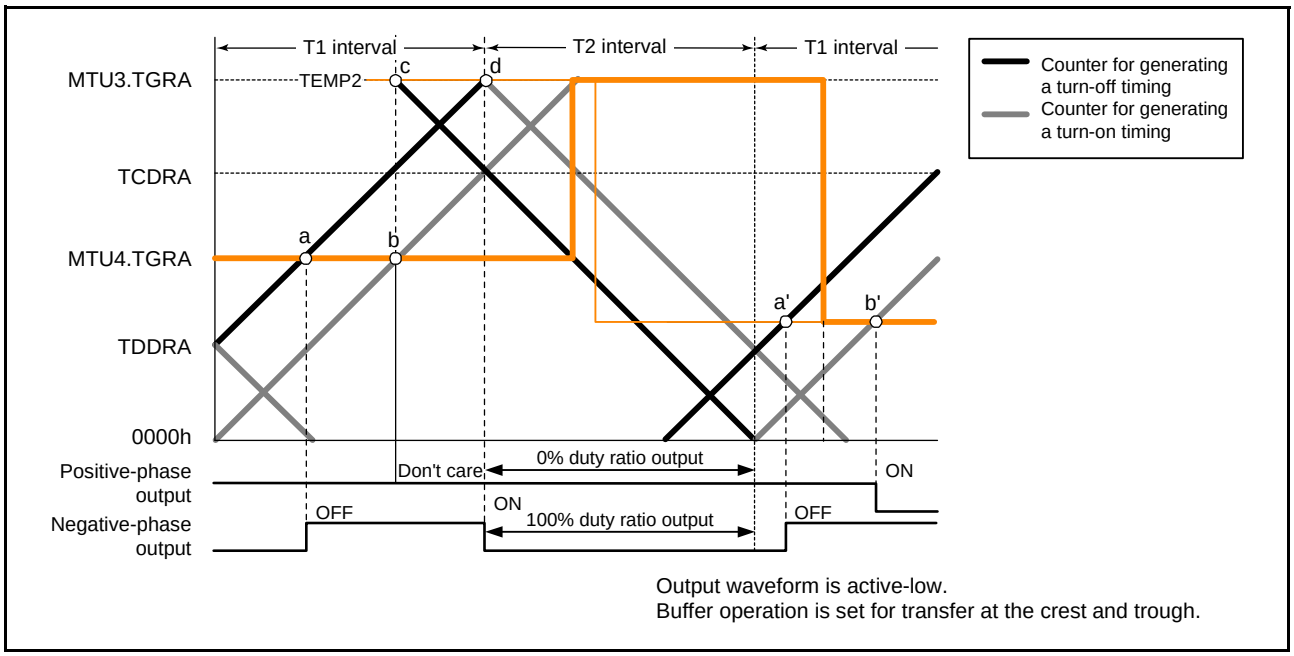


Figure 24.59 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (1)

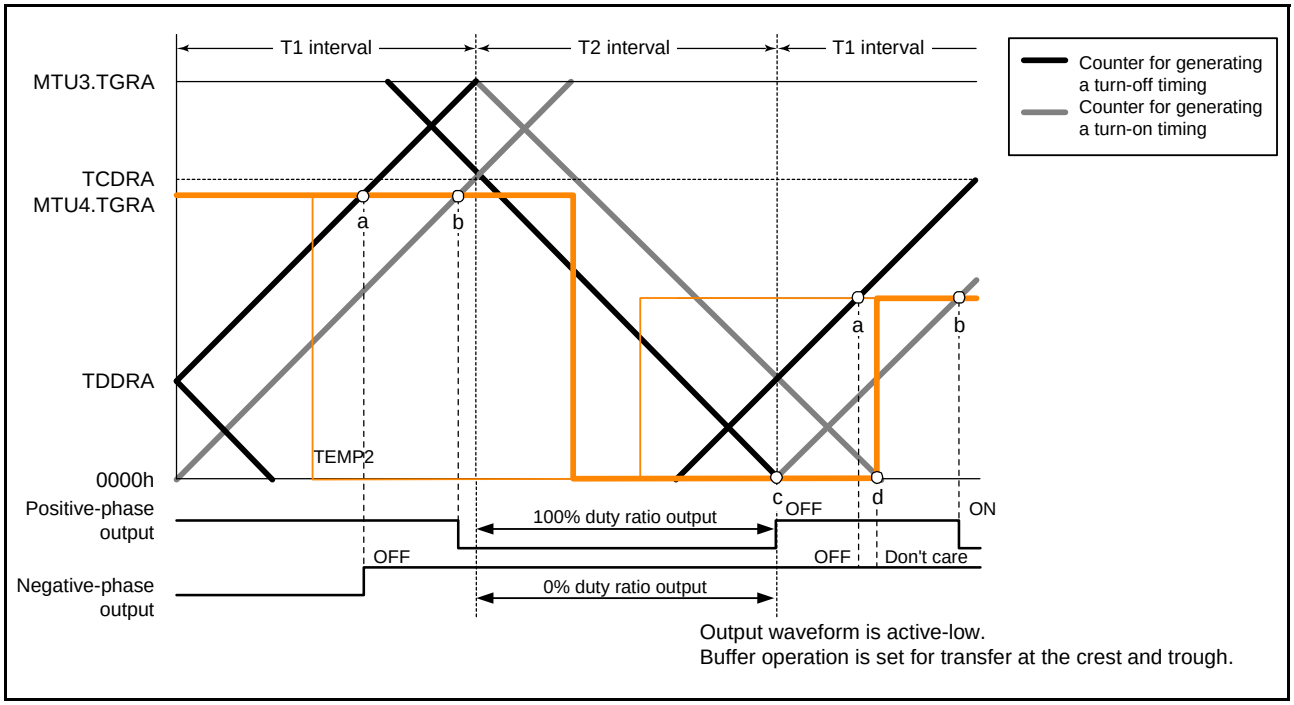


Figure 24.60 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (2)

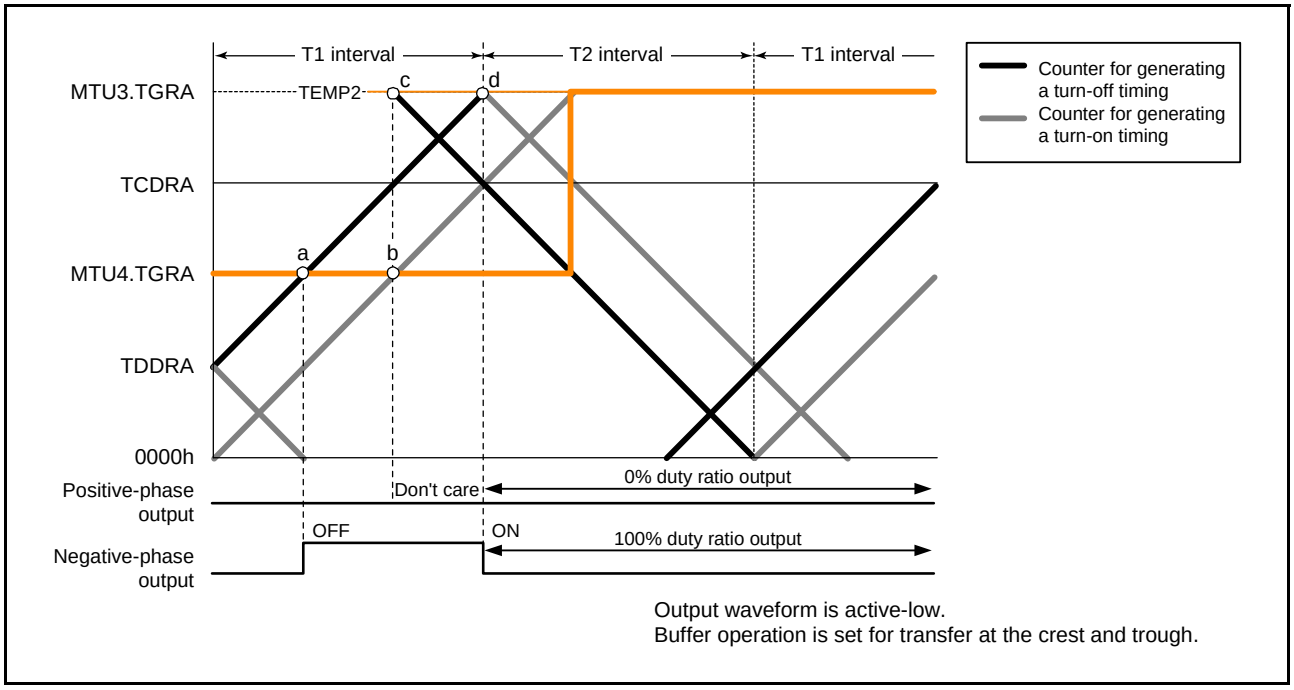


Figure 24.61 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (3)

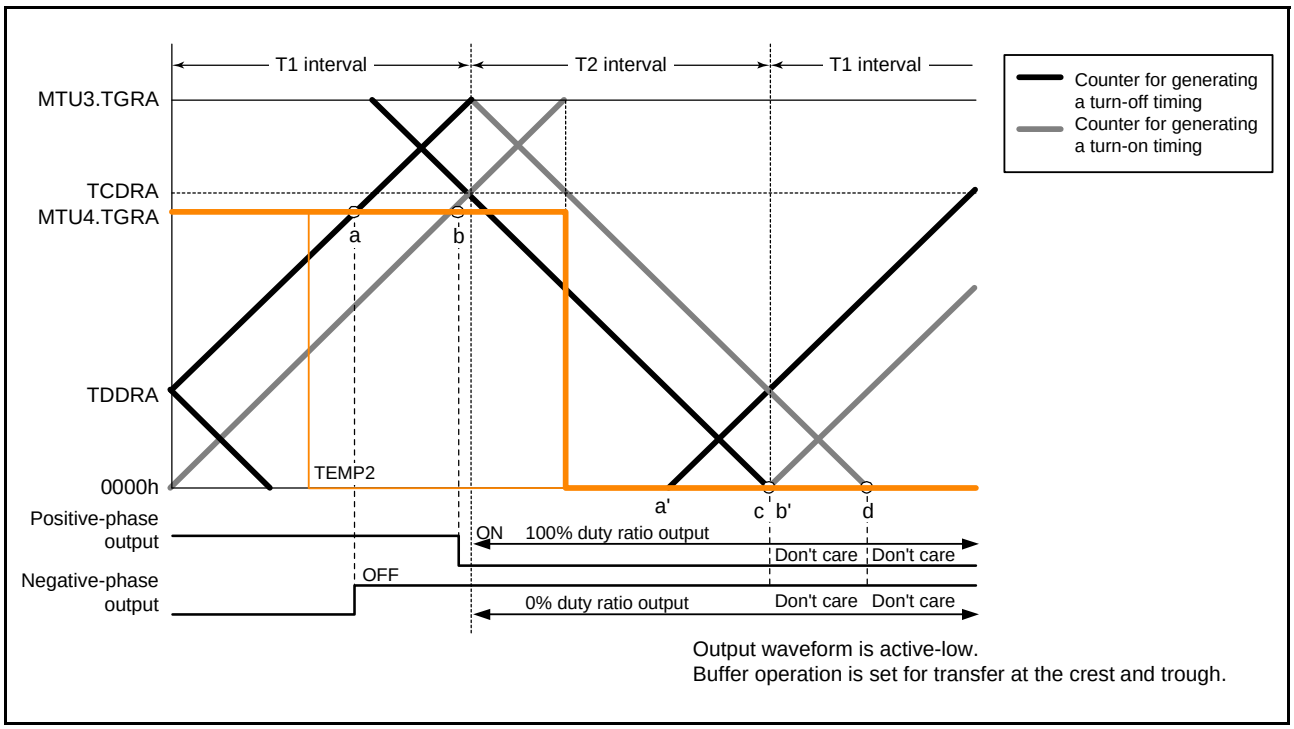


Figure 24.62 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (4)

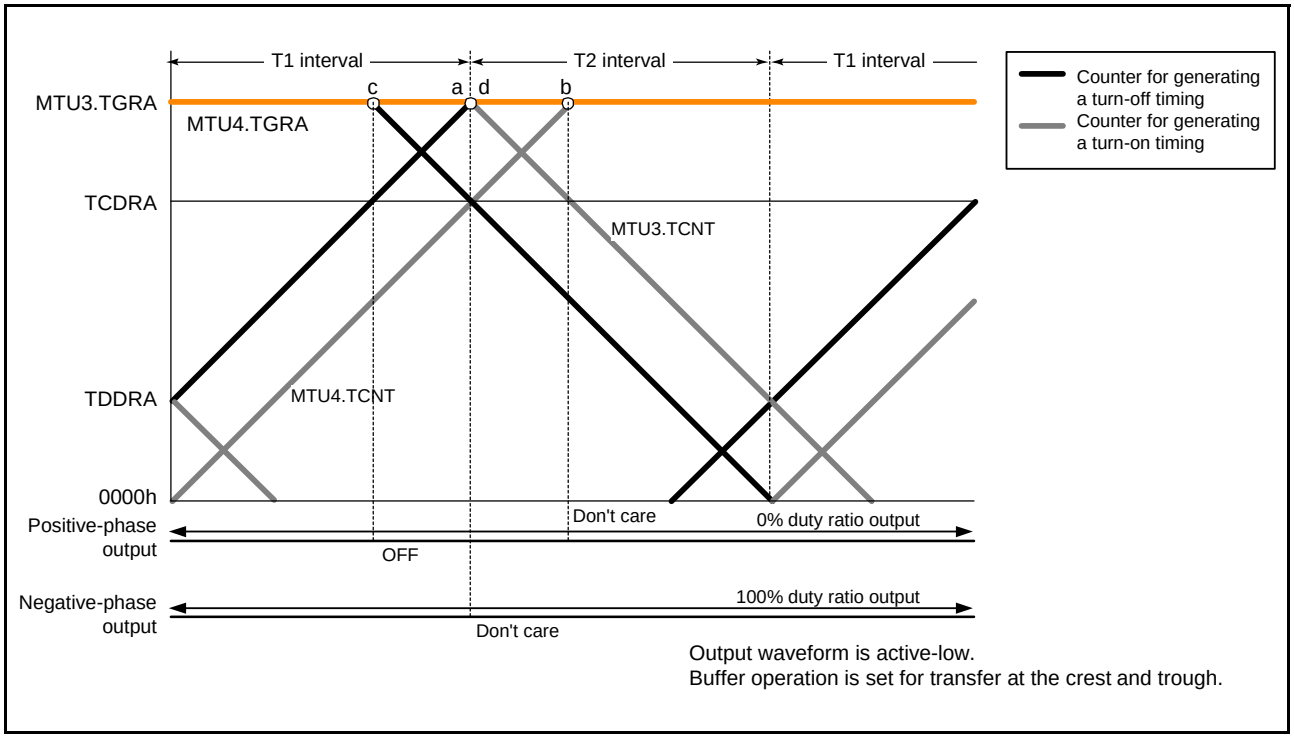


Figure 24.63 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (5)

(l) Toggle Output Synchronized with PWM Period

In complementary PWM mode, toggle output from the PWM output pin in synchronization with the PWM period can be enabled by setting the PSYE bit in the TOCR1A (TOCR1B) register to 1. An example of a toggle output waveform is shown in Figure 24.64.

This output is toggled by a compare match between MTU3.TCNT and MTU3.TGRA (MTU6.TCNT and MTU6.TGRA) and a compare match between MTU4.TCNT (MTU7.TCNT) and 0000h.

The MTIOC3A (MTIOC6A) pin is assigned for this toggle output. The initial output is high-level output.

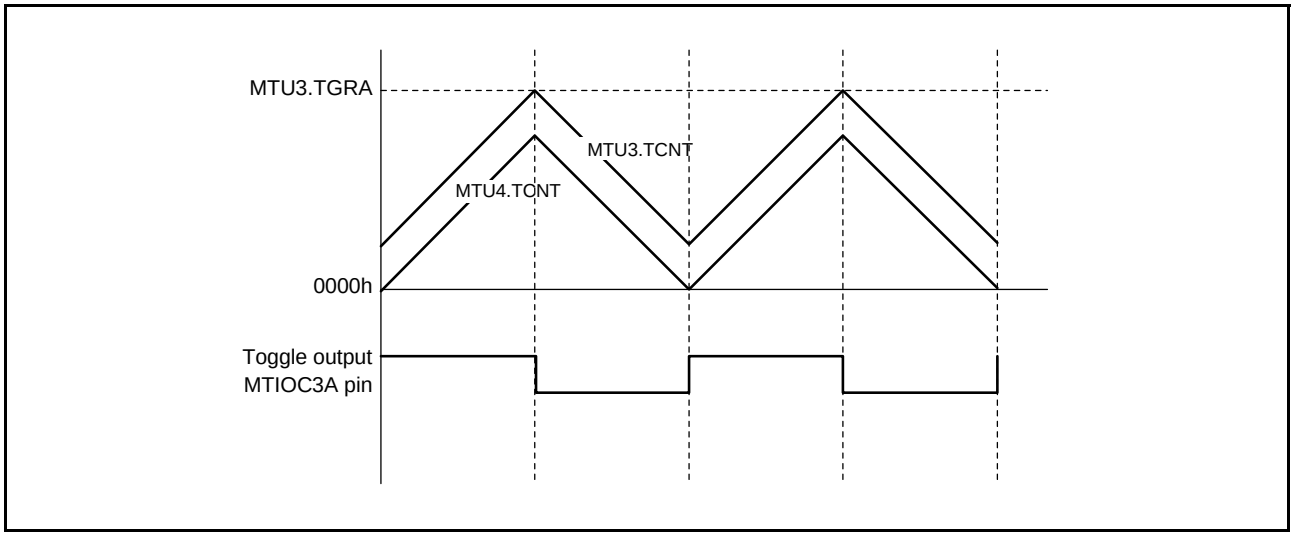


Figure 24.64 Example of Toggle Output Waveform Synchronized with PWM Output (MTU3 and MTU4)

(m) Counter Clearing by Another Channel

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTSA (MTU6.TCNT, MTU7.TCNT, and TCNTSB) can be cleared by another channel source when a mode for synchronization with another channel is specified through the TSYRA (TSYRB) register and synchronous clearing is selected with MTU3.TCR.CCLR[2:0] (MTU6.TCR.CCLR[2:0]) bits.

Figure 24.65 illustrates an example of this operation.

Use of this function enables a counter to be cleared and restarted through an external signal.

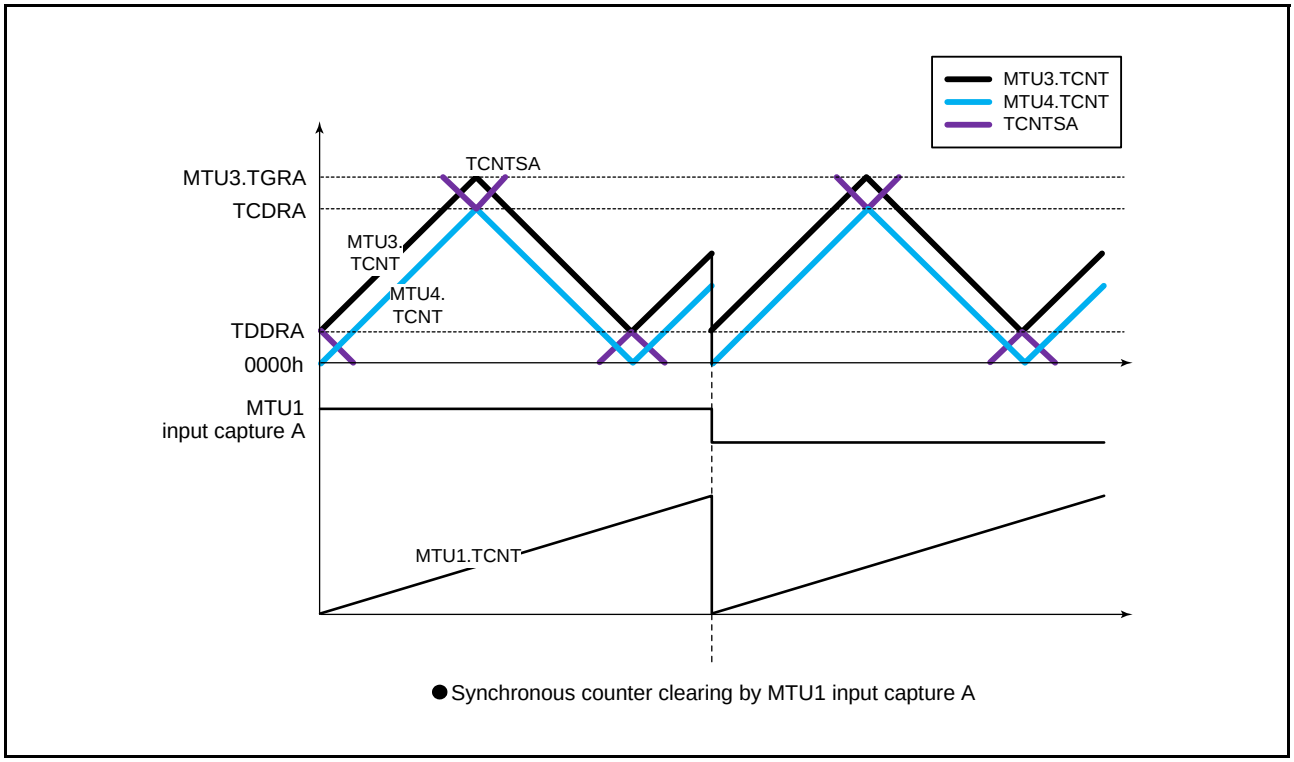


Figure 24.65 Counter Clearing Synchronized with Another Channel (MTU3 and MTU4)

(n) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Setting the WRE bit in TWCRA (TWCRB) to 1 suppresses initial output when synchronous counter clearing occurs in the Tb interval (Tb2 interval) at the trough in complementary PWM mode and controls abrupt change in duty cycle at synchronous counter clearing.

Initial output suppression through the WRE bit = 1 is applicable only when synchronous clearing occurs in the Tb2 interval as indicated by (10) or (11) in Figure 24.66. When synchronous clearing occurs outside that interval, the initial value specified by the OLSN and OLSP bits in TOCR1A (TOCR1B) is output. Even in the Tb2 interval, if synchronous clearing occurs in the initial value output period (indicated by (1) in Figure 24.66) immediately after the counters start operation, initial value output is not suppressed.

This function can be used in both channel combinations of MTU3 and MTU4, and MTU6 and MTU7. In MTU3 and MTU4, synchronous clearing in any of MTU0 to MTU2 can cause counter clearing; in MTU6 and MTU7,compare match or input capture in any of MTU0 to MTU2 can cause counter clearing.

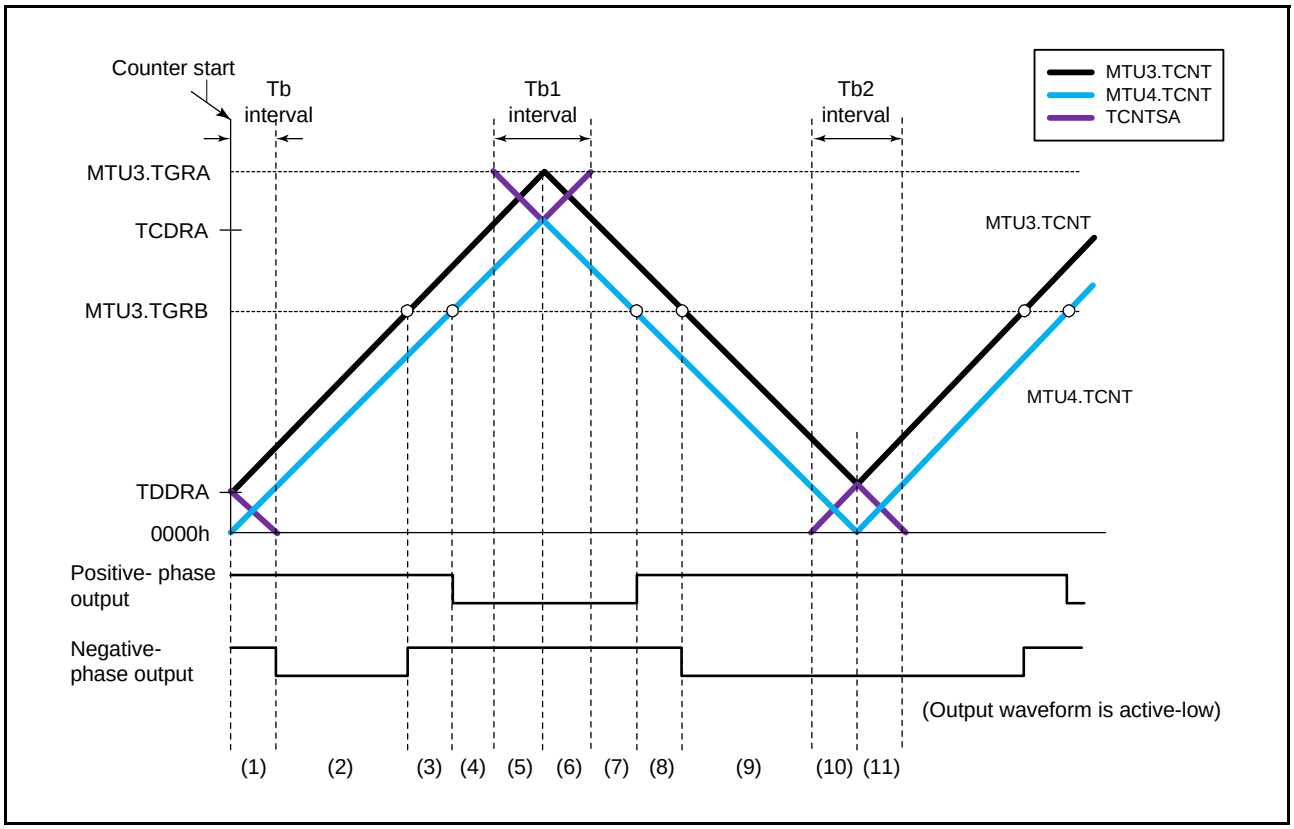


Figure 24.66 Timing for Synchronous Counter Clearing (MTU3 and MTU4)

- Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode.

An example of the procedure for setting output waveform control at synchronous counter clearing in complementary PWM mode is shown in Figure 24.67.

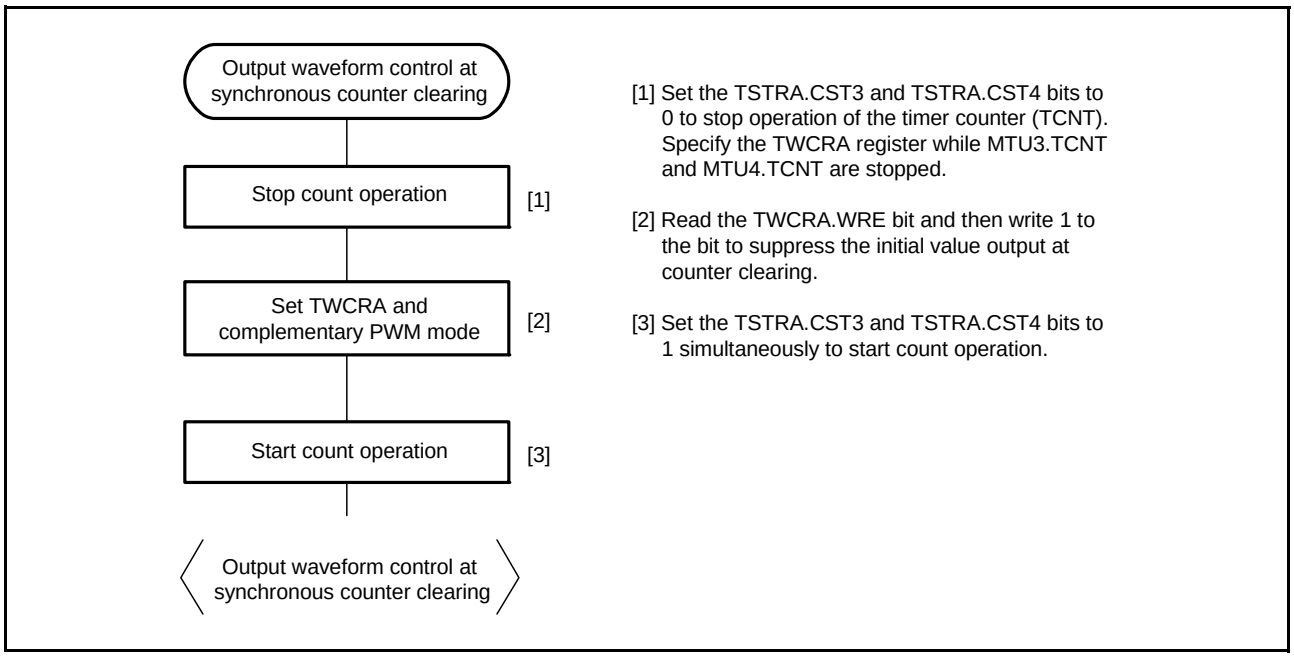


Figure 24.67 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode (MTU3 and MTU4)

- Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Figure 24.68 to Figure 24.71 show examples of output waveform control in which MTU3 and MTU4 operate in complementary PWM mode and synchronous counter clearing is generated while the WRE bit in TWCRA is set to 1. In the examples shown in Figure 24.68 to Figure 24.71, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in Figure 24.66, respectively.

In MTU6 and MTU7, these examples are equivalent to the cases when MTU6 and MTU7 operate in complementary PWM mode and synchronous counter clearing is generated while the SCC bit is set to 0 and the WRE bit is set to 1 in TWCRB.

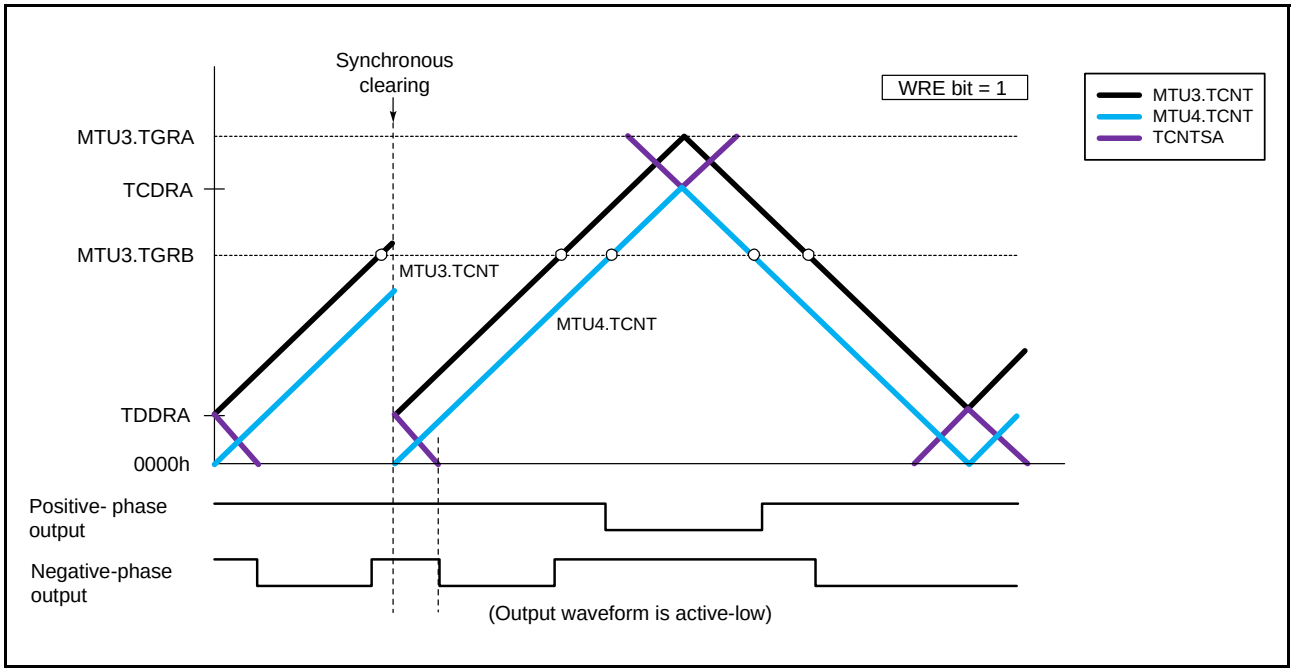


Figure 24.68 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 24.66; TWCRA.WRE Bit is 1)

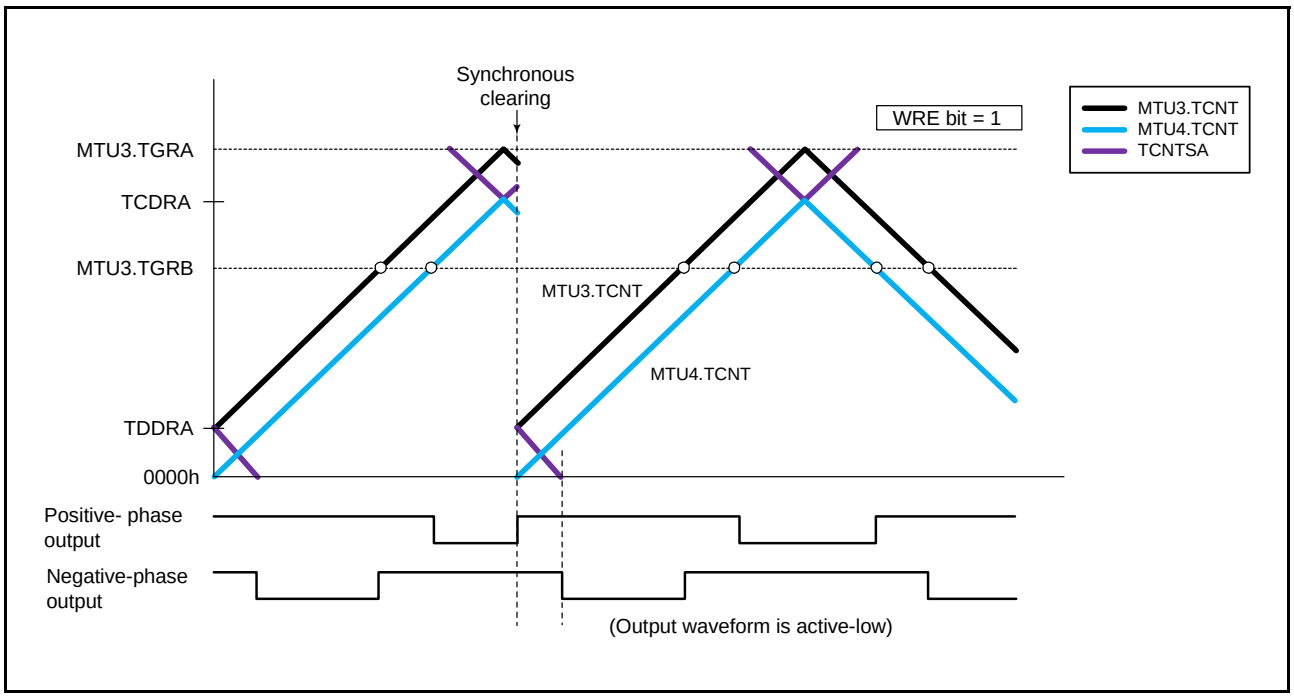


Figure 24.69 Example of Synchronous Clearing in Tb1 interval (Timing (6) in Figure 24.66; TWCRA.WRE Bit is 1)

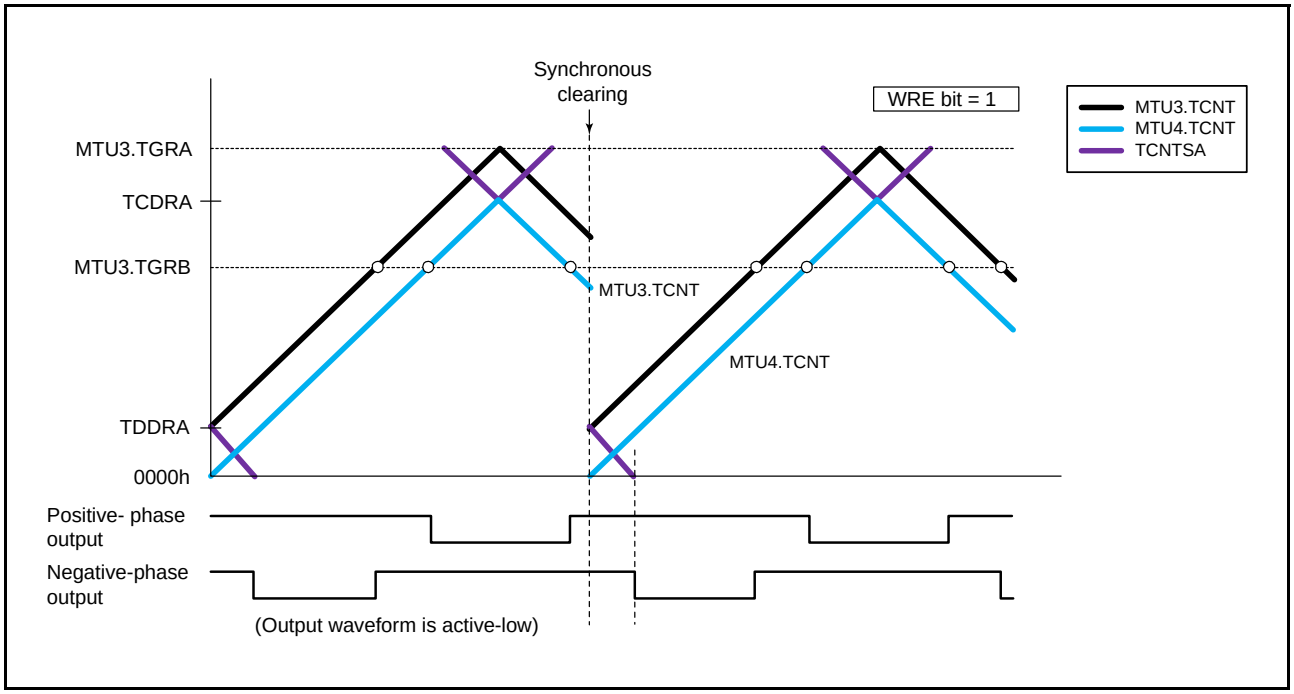


Figure 24.70 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 24.66; TWCRA.WRE Bit is 1)

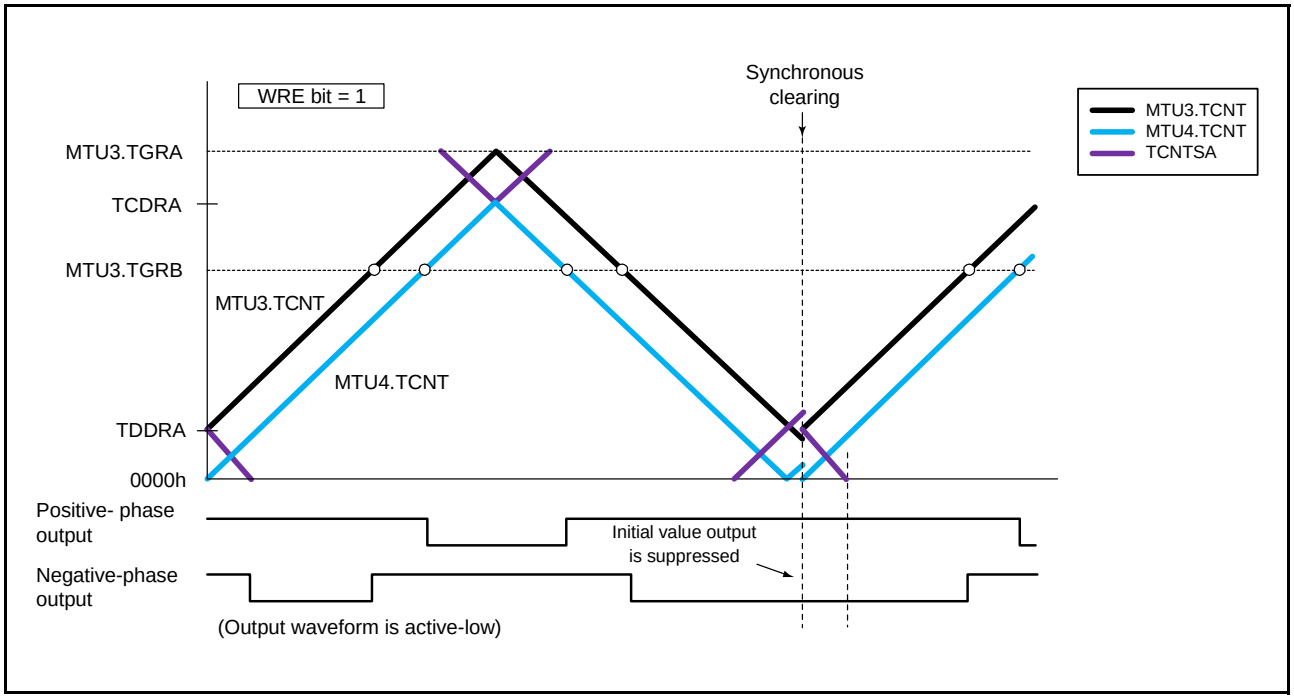


Figure 24.71 Example of Synchronous Clearing in Tb2 interval (Timing (11) in Figure 24.66; TWCRA.WRE Bit is 1)

(o) Suppressing Synchronous Counter Clearing for MTU0 to MTU2, and MTU6 and MTU7

In MTU6 and MTU7, setting the SCC bit in TWCRB to 1 suppresses synchronous counter clearing caused by MTU0 to MTU2.

Synchronous counter clearing is suppressed only within the interval shown in Figure 24.72. When using this function, MTU6 and MTU7 should be set to complementary PWM mode.

For details of synchronous clearing caused by MTU0 to MTU2, refer to section 24.3.10 (2), Synchronous Counter Clearing for MTU6 and MTU7.

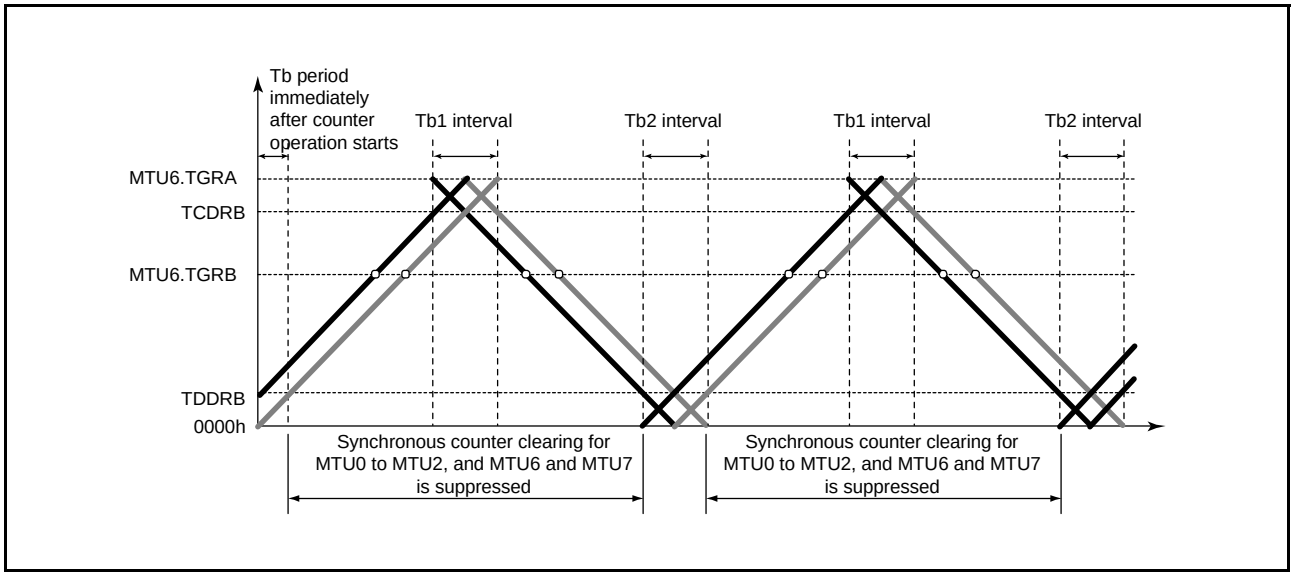


Figure 24.72 Synchronous Clearing-Suppressed Interval Specified by TWCRB.SCC Bit for MTU0 to MTU2, and MTU6 and MTU7

- Example of Procedure for Suppressing Synchronous Counter Clearing for MTU0 to MTU2, and MTU6 and MTU7

An example of the procedure for suppressing synchronous counter clearing for MTU0 to MTU2, and MTU6 and MTU7 is shown in Figure 24.73.

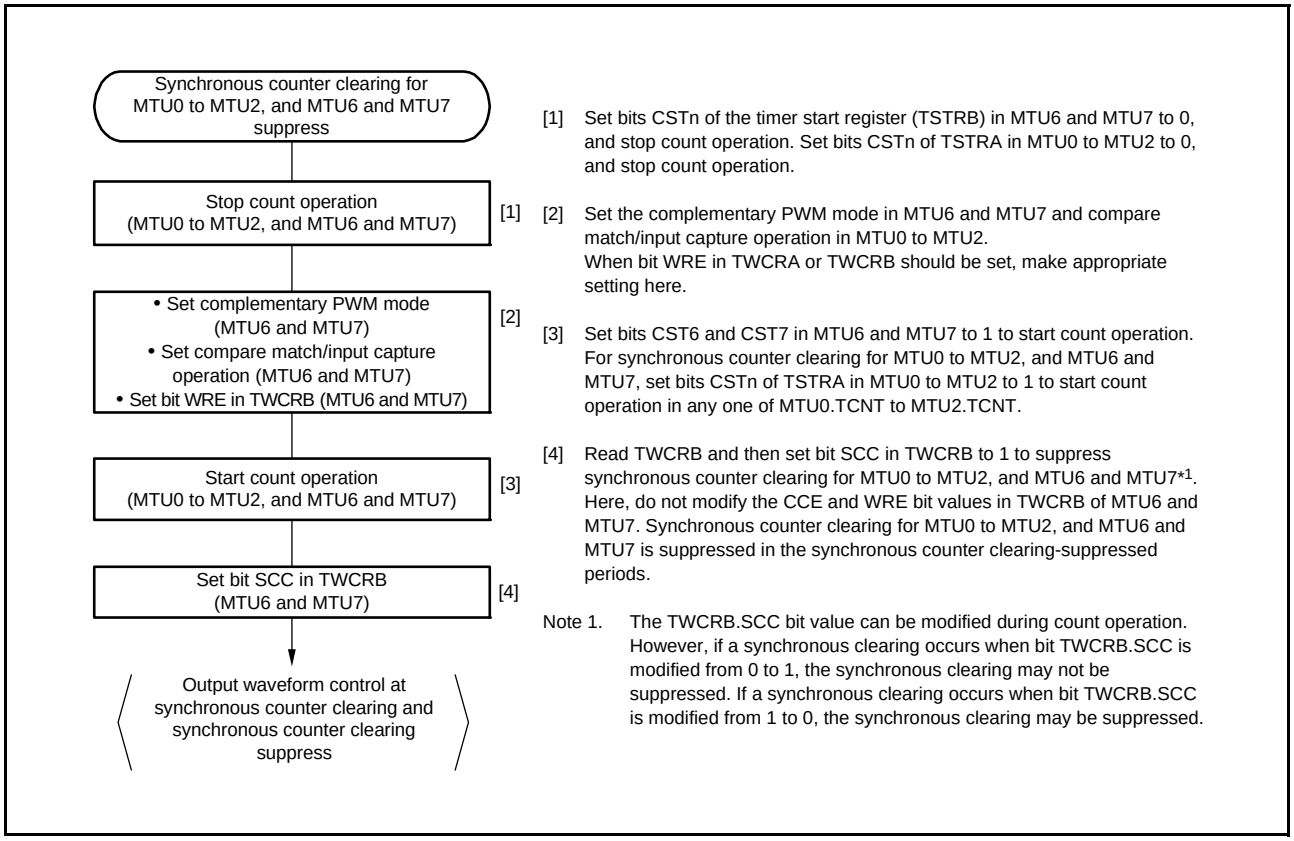


Figure 24.73 Example of Procedure for Suppressing Synchronous Counter Clearing for MTU0 to MTU2, and MTU6 and MTU7

- Examples of Suppression of Synchronous Counter Clearing for MTU 0 to MTU2, and MTU6 and MTU7

Figure 24.74 to Figure 24.77 show examples of operation in which MTU6 and MTU7 operate in complementary PWM mode and synchronous counter clearing for MTU 0 to MTU2, and MTU6 and MTU7 is suppressed by setting the SCC bit in TWCRB in MTU6 and MTU7 to 1. In the examples shown in Figure 24.74 to Figure 24.77, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in Figure 24.66, respectively. In these examples, the WRE bit in TWCRB in MTU6 and MTU7 is set to 1.

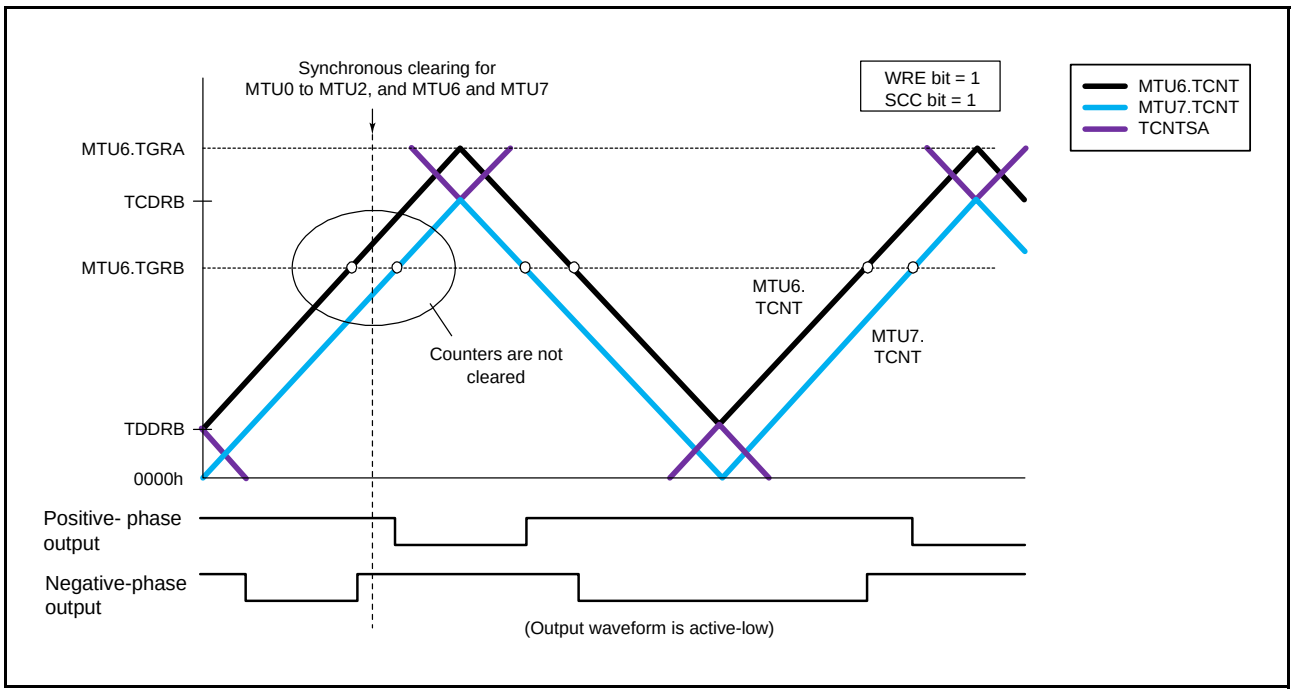


Figure 24.74 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 24.66; TWCRB.WRE Bit is 1 and SCC Bit is 1 in MTU6 and MTU7)

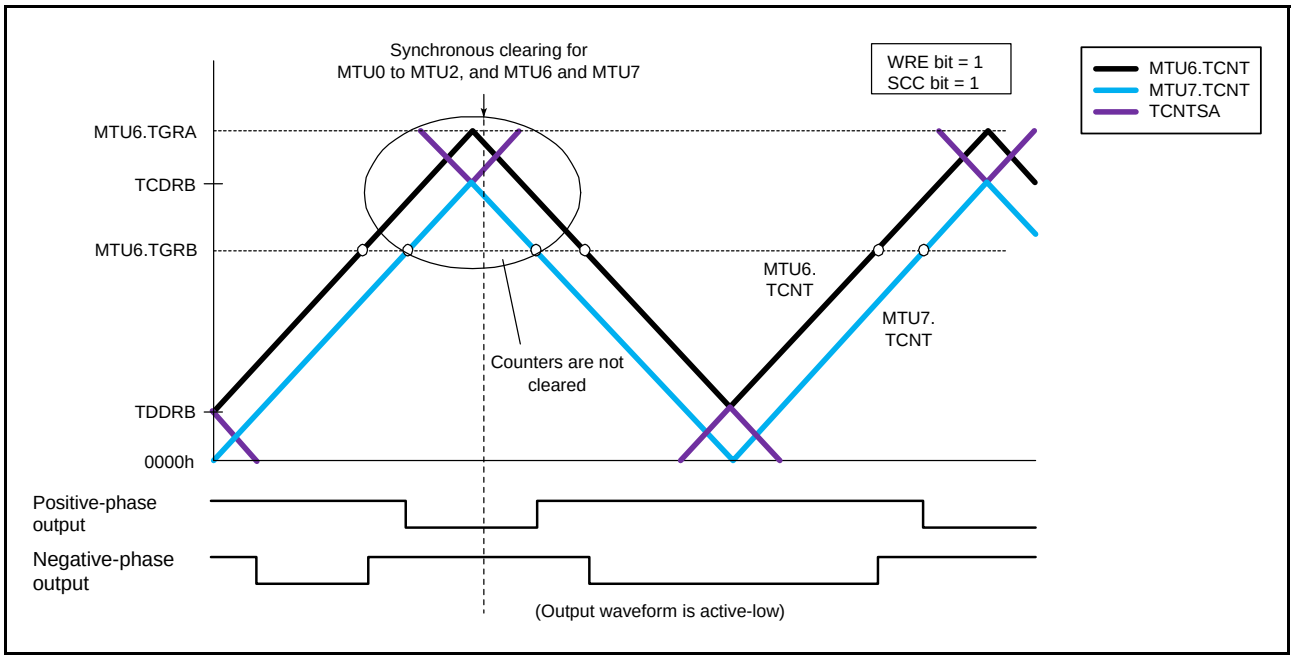


Figure 24.75 Example of Synchronous Clearing in Tb1 interval (Timing (6) in Figure 24.66; TWCRB.WRE Bit is 1 and SCC Bit is 1 in MTU6 and MTU7)

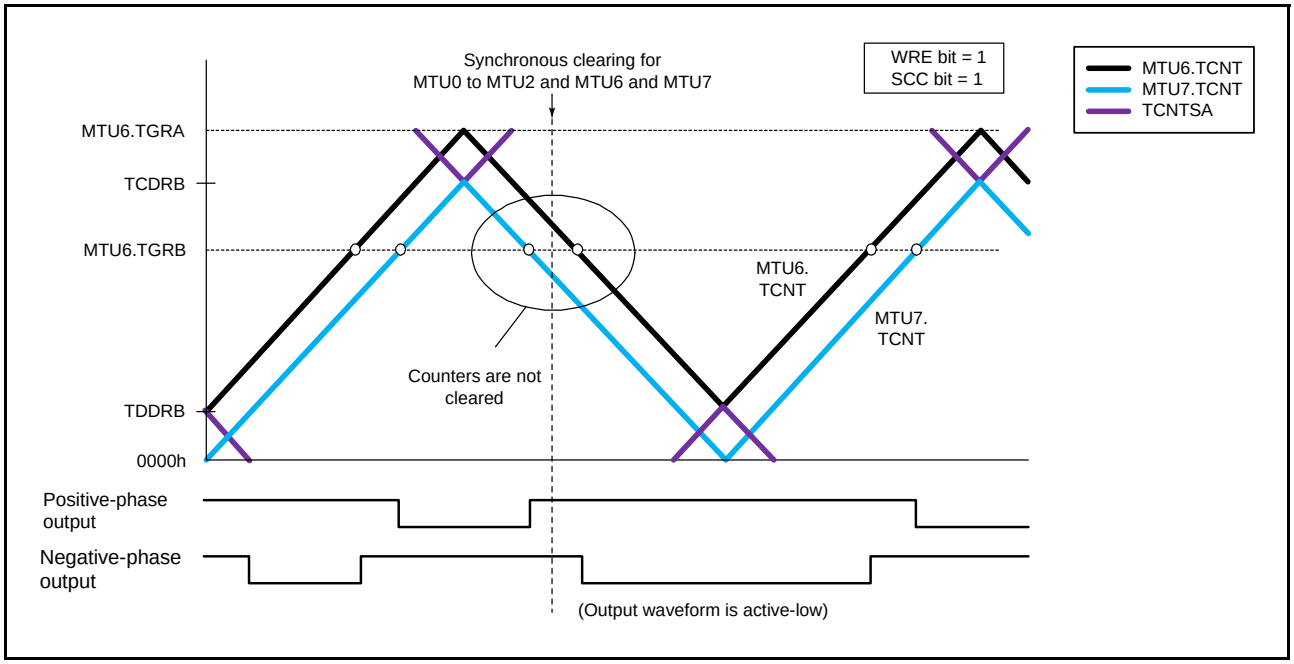


Figure 24.76 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 24.66; TWCRB.WRE Bit is 1 and SCC Bit is 1 in MTU6 and MTU7)

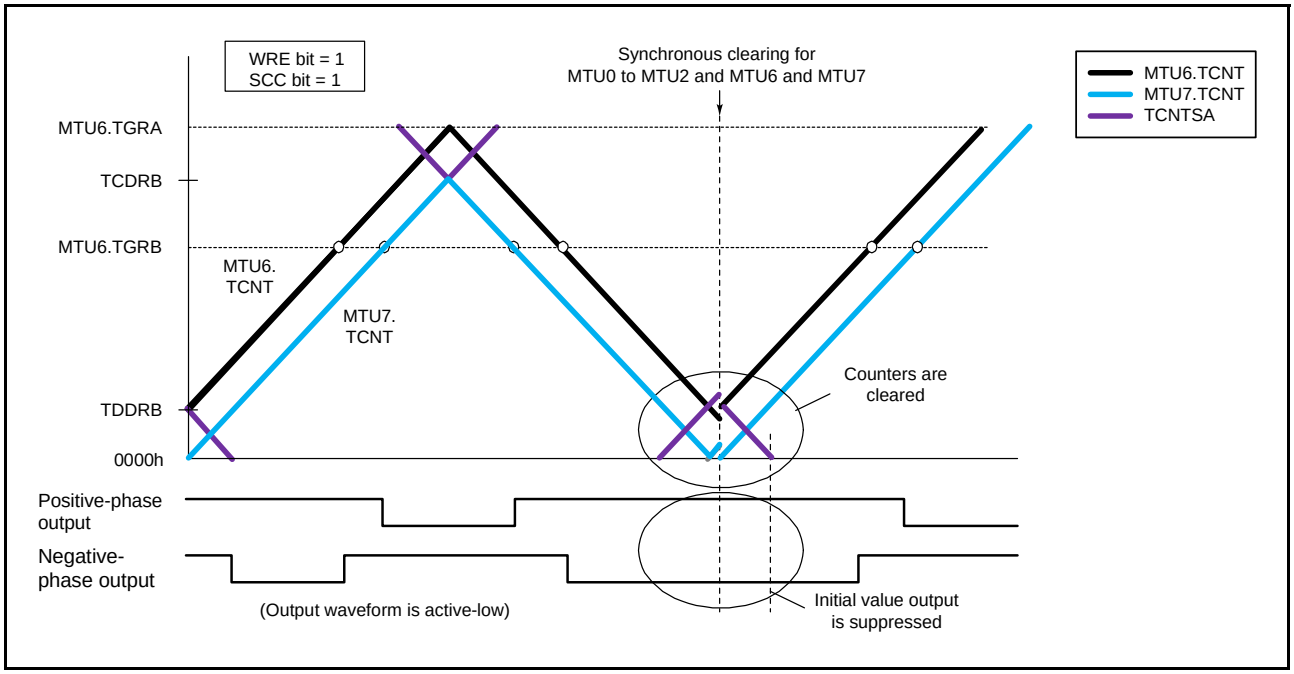


Figure 24.77 Example of Synchronous Clearing in Tb2 interval (Timing (11) in Figure 24.66; TWCRB.WRE Bit is 1 and SCC Bit is 1 in MTU6 and MTU7)

(p) Counter Clearing by MTU3.TGRA (MTU6.TGRA) Compare Match

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTSA (MTU6.TCNT, MTU7.TCNT, and TCNTSB) can be cleared by MTU3.TGRA (MTU6.TGRA) compare match when the TWCRA.CCE (TWCRB.CCE) bit. Figure 24.78 illustrates an operation example.

Note 1. Use this function only in complementary PWM mode 1 (transfer at crest).
Note 2. Do not specify synchronous clearing by another channel (do not set 1 in the SYNC0 to SYNC4, or SYNC6 and SYNC7 bits in the timer synchronous register (TSYRA or TSYRB) and the CE0A to CE0D, CE1A, CE1B, CE2A, or CE2B bits in TSYCR).
Note 3. Do not set the PWM duty value to 0000h.
Note 4. Do not set the PSYE bit in timer output control register 1 (TOCR1A or TOCR1B) to 1.

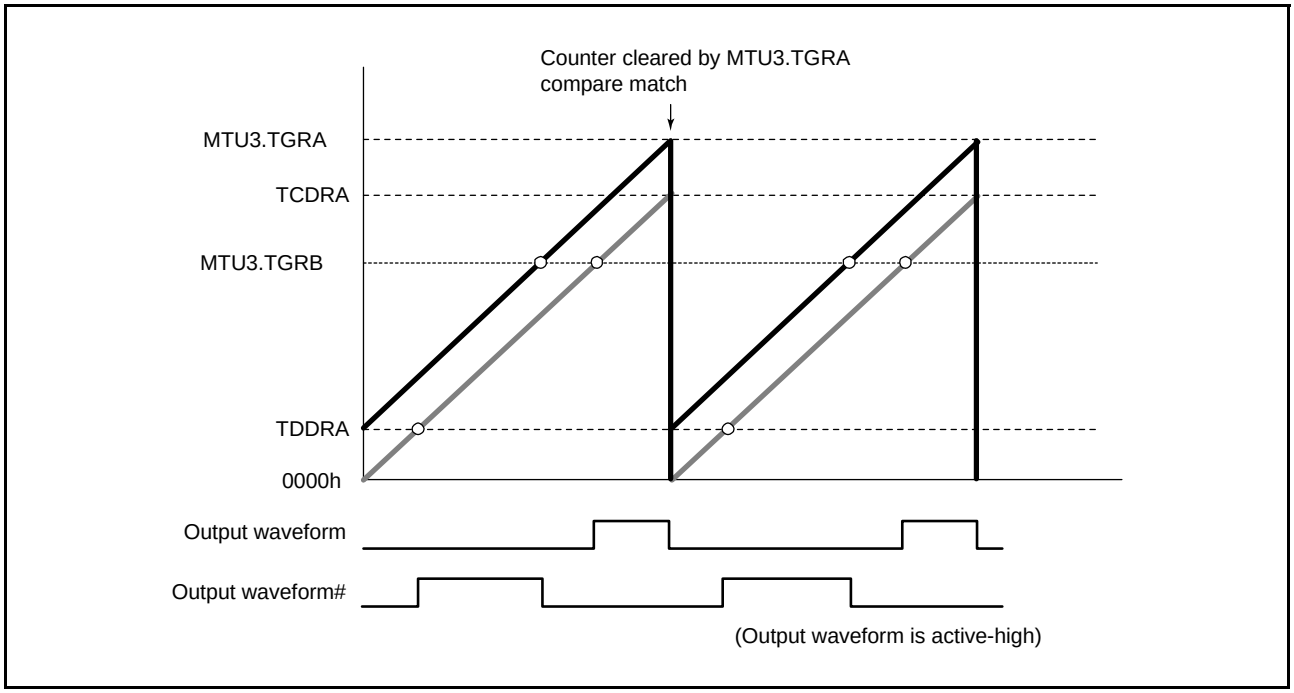


Figure 24.78 Example of Counter Clearing Operation by MTU3.TGRA Compare Match

(q) Example of Waveform Output for Driving AC Synchronous Motor (Brushless DC Motor)

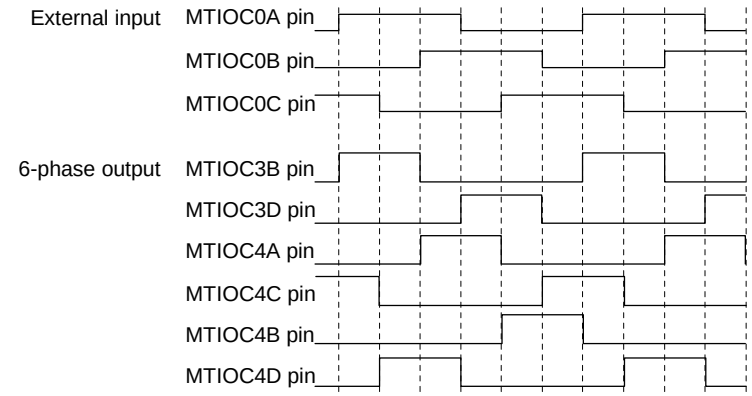
In complementary PWM mode when MTU3 and MTU4 are used, a brushless DC motor can easily be controlled using the TGCRA register. Figure 24.79 to Figure 24.82 show examples of brushless DC motor driving waveforms created using TGCRA.

To switch the output phases for a 3-phase brushless DC motor by means of external signals detected with a Hall element, etc., set the TGCRA.FB bit to 0. In this case, the external signals indicating the magnetic pole position should be input to timer input pins MTIOC0A, MTIOC0B, and MTIOC0C in MTU0 (make appropriate settings with the MPC and port mode registers (PMR) of the I/O ports). When an edge is detected at pin MTIOC0A, MTIOC0B, or MTIOC0C, the output on/off state is switched automatically.

When the TGCRA.FB bit is 1, the output on/off state is switched when the UF, VF, or WF bit in TGCRA is set to 0 or 1. The driving waveforms are output from the 6-phase PWM output pins for complementary PWM mode.

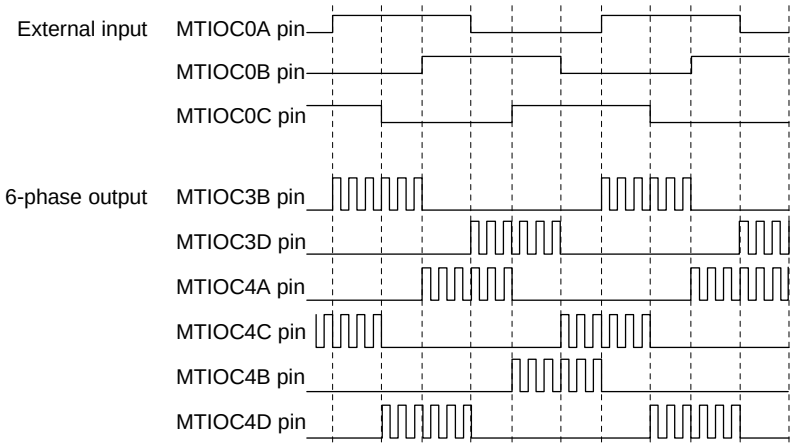
With this 6-phase output, while the output is turned on, chopping output is available through complementary PWM mode output function by setting the N bit or P bit in TGCRA to 1. When the N bit or P bit is 0, the level output is selected.

The active level of the 6-phase output (on output level) can be set with the TOCR1A.OLSN and TOCR1A.OLSP bits regardless of the setting of the N and P bits.



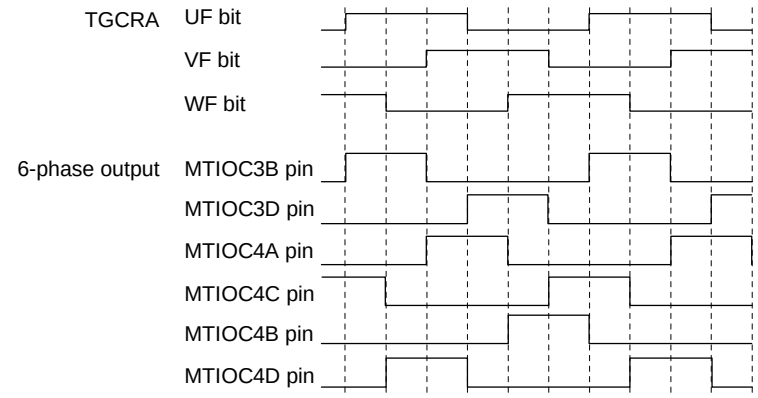
- When TGCRA.BDC = 1, TGCRA.N = 0, TGCRA.P = 0, TGCRA.FB = 0, and output active level = high

Figure 24.79 Example of Output Phase Switching by External Input (1)



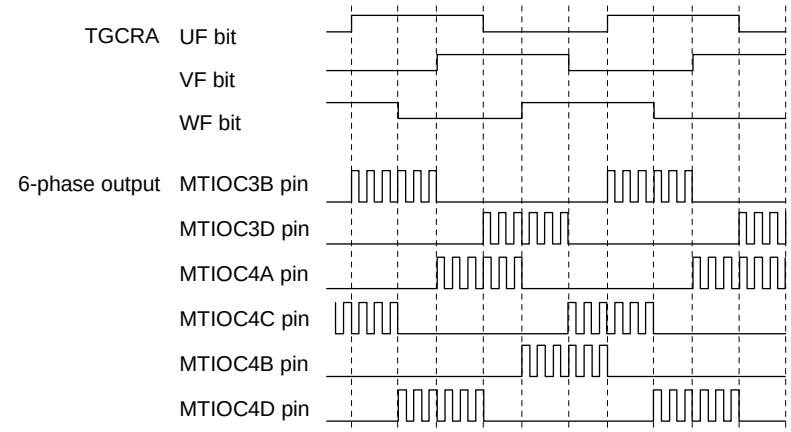
- When TGCRA.BDC = 1, TGCRA.N = 1, TGCRA.P = 1, TGCRA.FB = 0, and output active level = high

Figure 24.80 Example of Output Phase Switching by External Input (2)



- When TGCRA.BDC = 1, TGCRA.N = 0, TGCRA.P = 0, TGCRA.FB = 1, and output active level = high

Figure 24.81 Example of Output Phase Switching through UF, VF, and WF Bit Settings (1)



- When TGCRA.BDC = 1, TGCRA.N = 1, TGCRA.P = 1, TGCRA.FB = 1, and output active level = high

Figure 24.82 Example of Output Phase Switching through UF, VF, and WF Bit Settings (2)

(r) A/D Conversion Start Request Setting

In complementary PWM mode, an A/D conversion start request can be issued using MTU3.TGRA (MTU6.TGRA) compare match, MTU4.TCNT (MTU7.TCNT) underflow (trough), or compare match on a channel other than MTU3 and MTU4 (MTU6 and MTU7).
When start requests using MTU3.TGRA (MTU6.TGRA) compare match are specified, A/D conversion can be started at the crest of the MTU3.TCNT (MTU6.TCNT) count.
A/D conversion start requests can be specified by setting the TIER.TTGE bit. To issue an A/D conversion start request at an MTU4.TCNT (MTU7.TCNT) underflow (trough), set the MTU4.TIER.TTGE2 (MTU7.TIER.TTGE2) bit to 1.

(s) Double Buffer Function in Complementary PWM Mode

In complementary PWM mode 3 (transfer at the crest and trough), the PWM output setting resolution can be improved from ±2 to ±1 by setting the TMDR2A.DRS (TMDR2B.DRS) bit to 1.
When setting buffer registers A (MTU3.TGRD, MTU4.TGRC, MTU4.TGRD, MTU6.TGRD, MTU7.TGRC, and MTU7.TGRD), set also buffer registers B (MTU3.TGRE, MTU4.TGRE, MTU4.TGRF, MTU6.TGRE, MTU7.TGRE, and MTU7.TGRF) at the same time. Each buffer register B should be set to the buffer register A value or (buffer register A value – 1). For details of the setting procedure, refer to section 24.3.8 (1), Example of Complementary PWM Mode Setting Procedure

Note: When a buffer register B is set to the buffer register A value, symmetric PWM waveforms are output. When a buffer register B is set to (buffer register A value – 1), asymmetric PWM waveforms are output.

Figure 24.83 shows an example of double buffer operation.
Each register data is transferred as follows.

- After MTU4.TGRD or MTU7.TGRD (buffer A) is written to, data is transferred from MTU4.TGRD or MTU7.TGRD (buffer A) to Temp3A or Temp6A (temporary A) and from MTU4.TGRF or MTU7.TGRF (buffer B) to Temp3B or Temp6B (temporary B).
- With timing (1) in the figure, data is transferred from Temp3A or Temp6A (temporary A) to MTU4.TGRB or MTU7.TGRB (compare).
- With timing (2) in the figure, data is transferred from Temp3B or Temp6B (temporary B) to MTU4.TGRB or MTU7.TGRB (compare).

In the crest interval (Tb1 interval), the compare register and temporary register A are valid; in the trough interval (Tb2 interval), the compare register and temporary register B are valid.

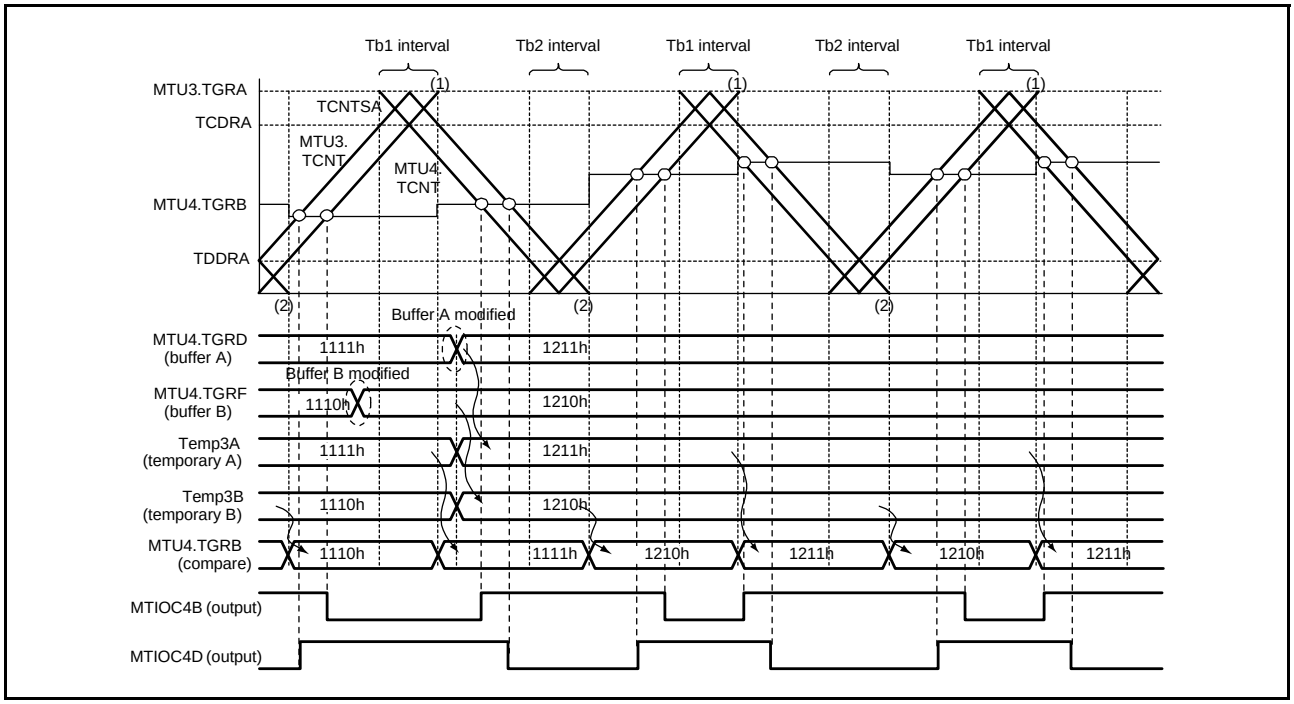


Figure 24.83 Example of Double Buffer Operation

Figure 24.84 shows an example when the buffer write value is smaller than the TDDRA (TDDRB) value, and Figure 24.85 shows an example when the write value is greater than TCDRA (TCDRB).
In the crest interval, the output is controlled according to the compare match with the compare register or temporary register A; in the trough interval, the output is controlled according to the compare match with the compare register or temporary register B.

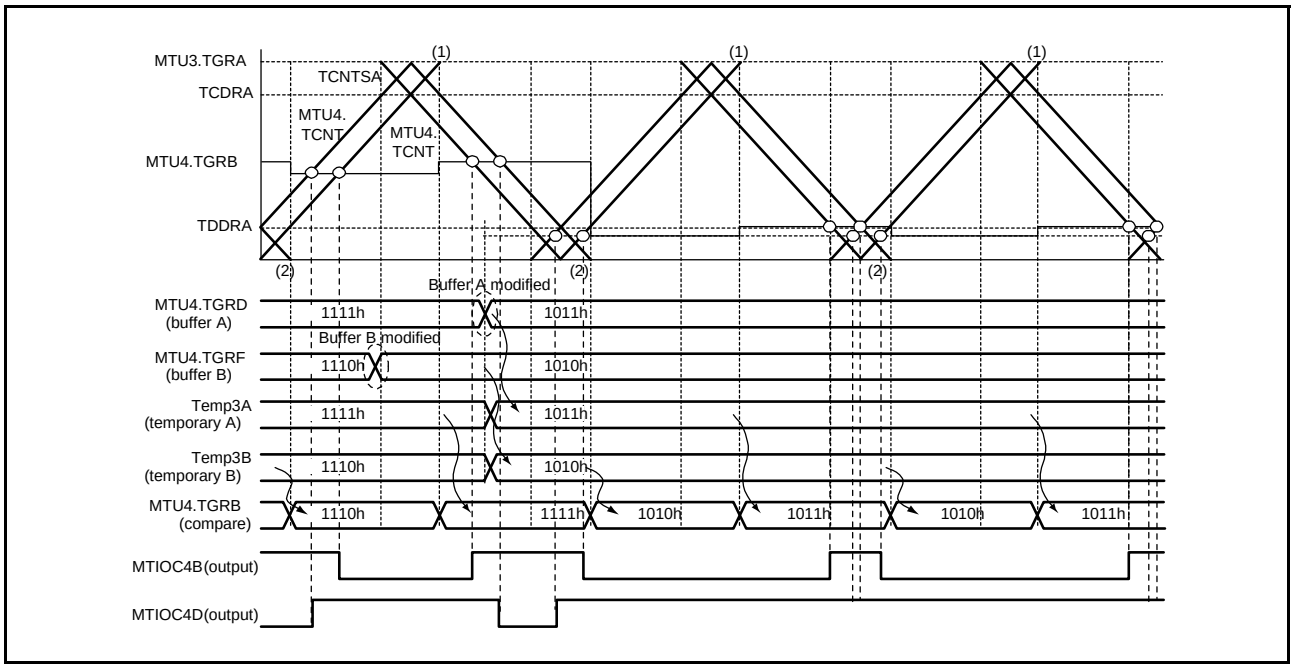


Figure 24.84 Example of Double Buffer Operation (Buffer Write Value is Smaller than TDDRA)

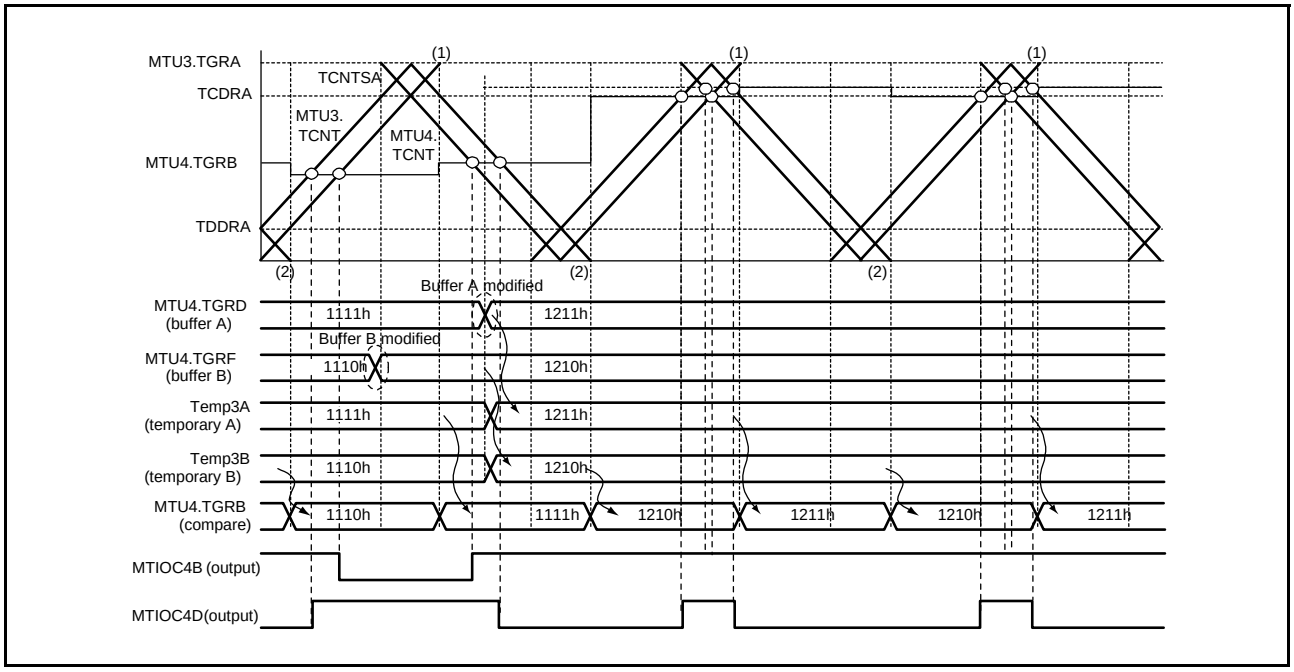


Figure 24.85 Example of Double Buffer Operation (Buffer Write Value is Greater than TCDRA)

(3) Interrupt Skipping Function 1 in Complementary PWM Mode

Interrupts TGIA3 (TGIA6) (at the crest) and TCIV4 (TCIV7) (at the trough) in MTU3 and MTU4 (MTU6 and MTU7) can be skipped up to seven times by making settings in the TITCR1A (TITCR1B) register.
Transfers from a buffer register to a temporary register or a compare register can be skipped in coordination with interrupt skipping by making settings in the TBTERA (TBTERB) register. For the linkage with buffer registers, refer to description (c), Buffer Transfer Control Linked with Interrupt Skipping, below.
A/D conversion start requests generated by the A/D conversion start request delaying function can also be skipped in coordination with interrupt skipping by making settings in the MTU4.TADCR (MTU7.TADCR) register. For the linkage with the A/D conversion start request delaying function, refer to section 24.3.9, A/D Conversion Start Request Delaying Function.
The TITCR1A (TITCR1B) register should be set while interrupt skipping function 1 is selected by setting the TITM bit in the timer interrupt skipping mode register (TITMRA or TITMRB) to 0, TGIA3 (TGIA6) interrupt requests are disabled by setting the MTU3.TIER (MTU6.TIER) register, TCIV4 (TCIV7) interrupt requests are disabled by setting the MTU4.TIER (MTU7.TIER) register, and a compare match is not generated. Before changing the skipping count, be sure to set the T3AEN (T6AEN) and T4VEN (T7VEN) bits to 0 to clear the skipping counter.

(a) Example of Interrupt Skipping Function 1 Setting Procedure

Figure 24.86 shows an example of the interrupt skipping function 1 setting procedure. Figure 24.87 shows the periods during which interrupt skipping count can be changed.

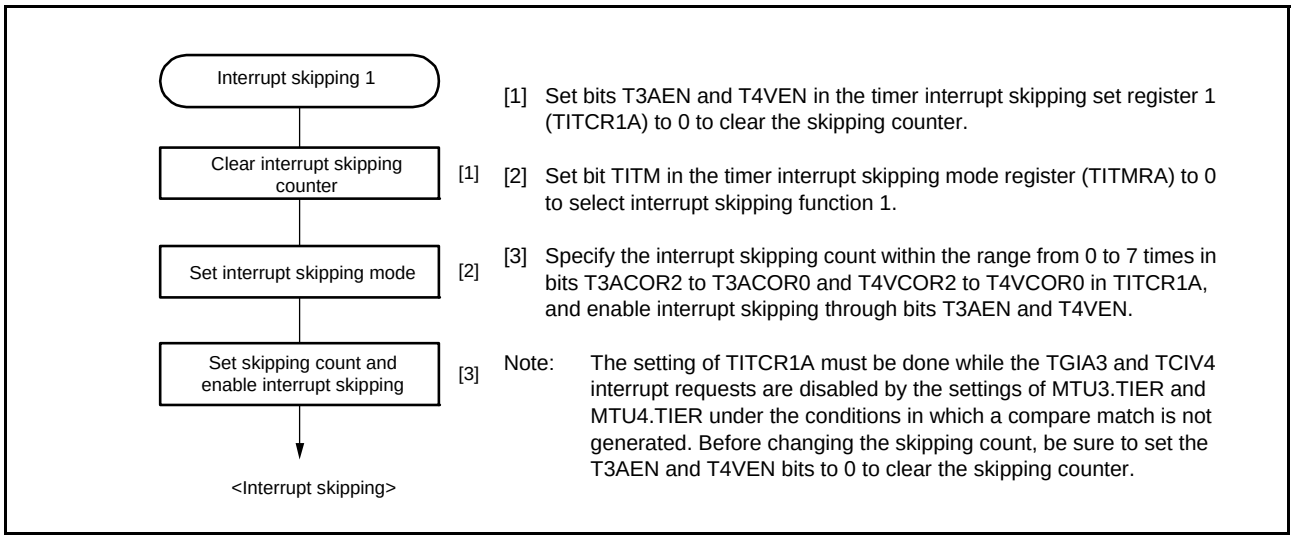


Figure 24.86 Example of Interrupt Skipping Function 1 Setting Procedure

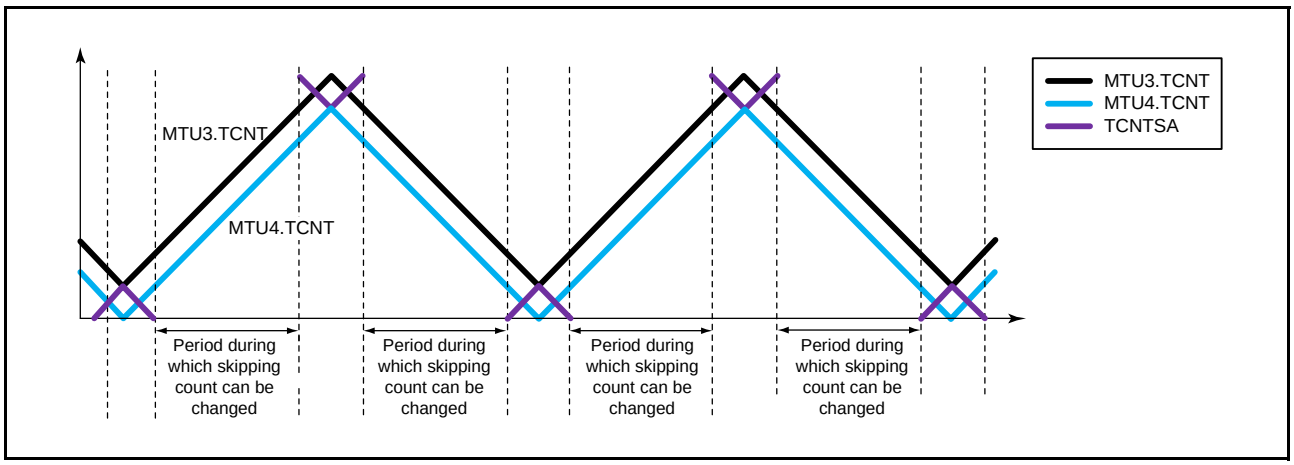


Figure 24.87 Periods during which Interrupt Skipping Count can be Changed

(b) Example of Interrupt Skipping Function 1

Figure 24.88 shows an example of TGIA3 (TGIA6) interrupt skipping in which the interrupt skipping count is set to three by the T3ACOR (T6ACOR) bits and the T3AEN (T6AEN) bit is set to 1 in the TITCR1A (TITCR1B) register.

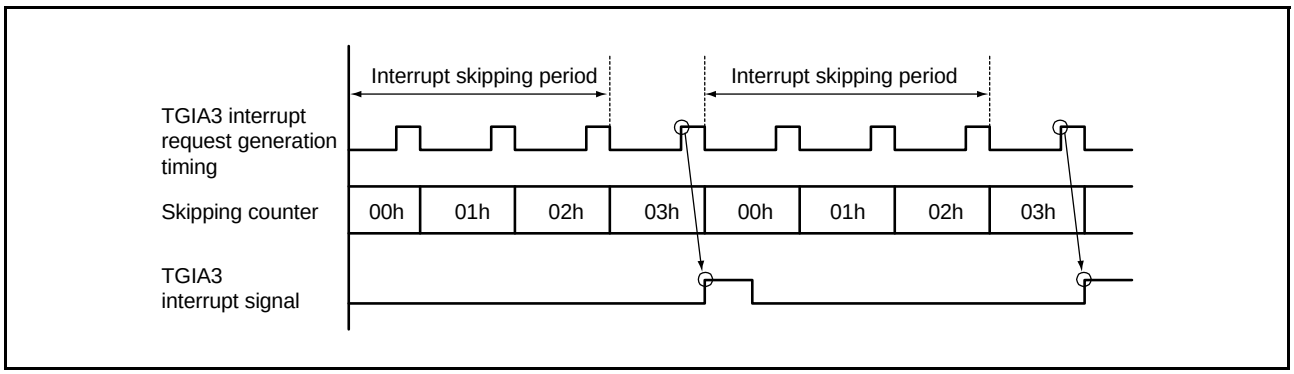
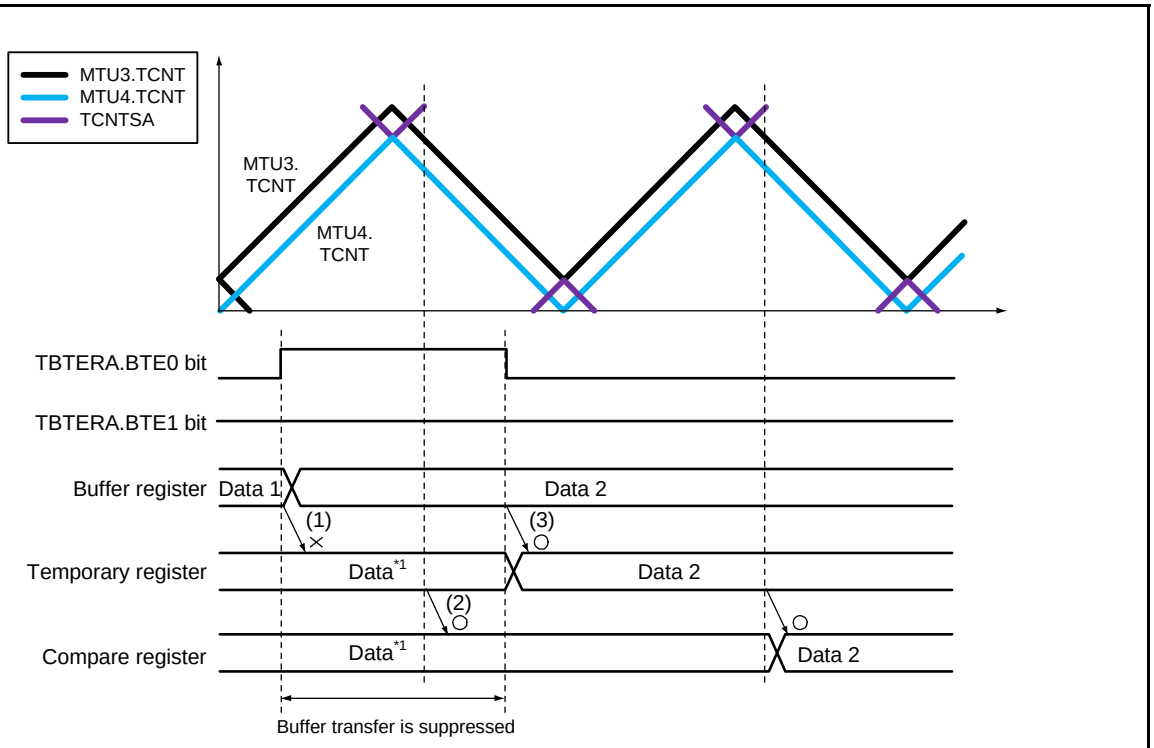


Figure 24.88 Example of Interrupt Skipping Function 1

(c) Buffer Transfer Control Linked with Interrupt Skipping

In complementary PWM mode, whether to transfer data from a buffer register to a temporary register and whether to link the transfer with interrupt skipping can be specified with the BTE[1:0] bits in the TBTERA (TBTERB) register.
Figure 24.89 shows an example of operation when buffer transfer is disabled (BTE[1:0] = 01b). While this setting is valid, data is not transferred from the buffer register to the temporary register.
Figure 24.90 shows an example of operation when buffer transfer is linked with interrupt skipping (BTE[1:0] = 10b). While this setting is valid, data is not transferred from the buffer register outside the buffer transfer-enabled period. Note that the buffer transfer-enabled period differs depending on whether only the T3AEN (T6AEN) bit in the TITCR1A (TITCR1B) register is set to 1, only the T4VEN (T7VEN) bit in the TITCR1A (TITCR1B) register is set to 1, or both the T3AEN and T4VEN (T6AEN and T7VEN) bits are set to 1. Figure 24.91 shows the relationship between the T3AEN (T6AEN) and T4VEN (T7VEN) bit settings in TITCR1A (TITCR1B) and buffer transfer-enabled period.

Note: This function must be used in combination with interrupt skipping function 1.
When interrupt skipping is disabled (the T3AEN and T4VEN (T6AEN and T7VEN) bits in the timer interrupt skipping set register 1 (TITCR1A or TITCR1B) are set to 0 or the skipping count set bits (T3ACOR and T4VCOR (T6ACOR and T7VCOR)) in TITCR1A (TITCR1B) are set to 0), make sure that buffer transfer is not linked with interrupt skipping (set the BTE1 bit in TBTERA or TBTERB to 0).
If buffer transfer is linked with interrupt skipping while interrupt skipping is disabled, buffer transfer is never performed.



(1) No data is transferred from the buffer register to the temporary register in the buffer transfer-disabled period (bits BTE[1:0] in TBTERA are set to 01b).
(2) Data is transferred from the temporary register to the compare register even in the buffer transfer-disabled period.
(3) After buffer transfer is enabled, data is transferred from the buffer register to the temporary register.

Note 1. When buffer transfer at the crest is selected.

Figure 24.89 Example of Operation When Buffer Transfer is Disabled (BTE[1:0] = 01b)

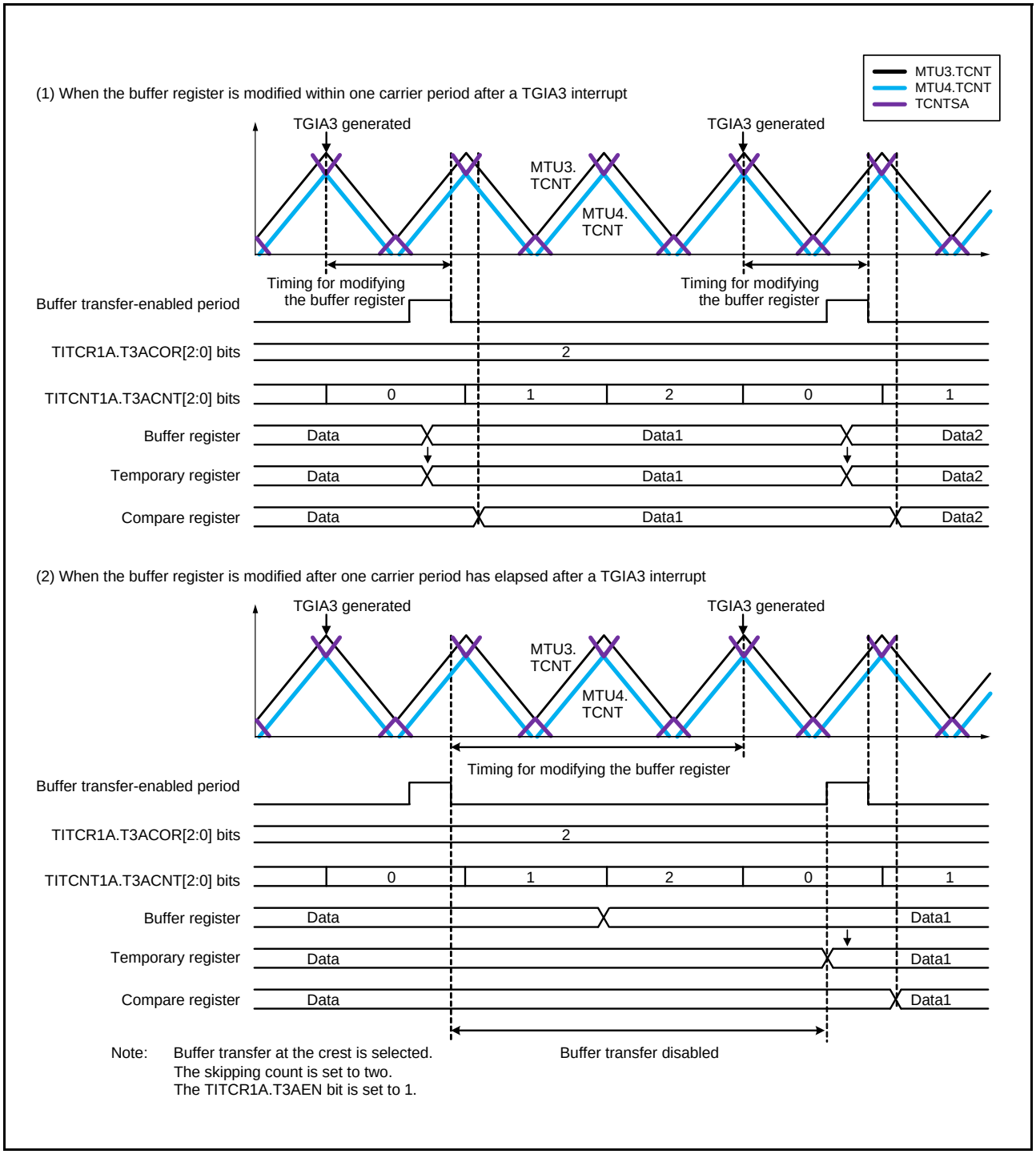


Figure 24.90 Example of Operation When Buffer Transfer is Linked with Interrupt Skipping (BTE[1:0] = 10b)

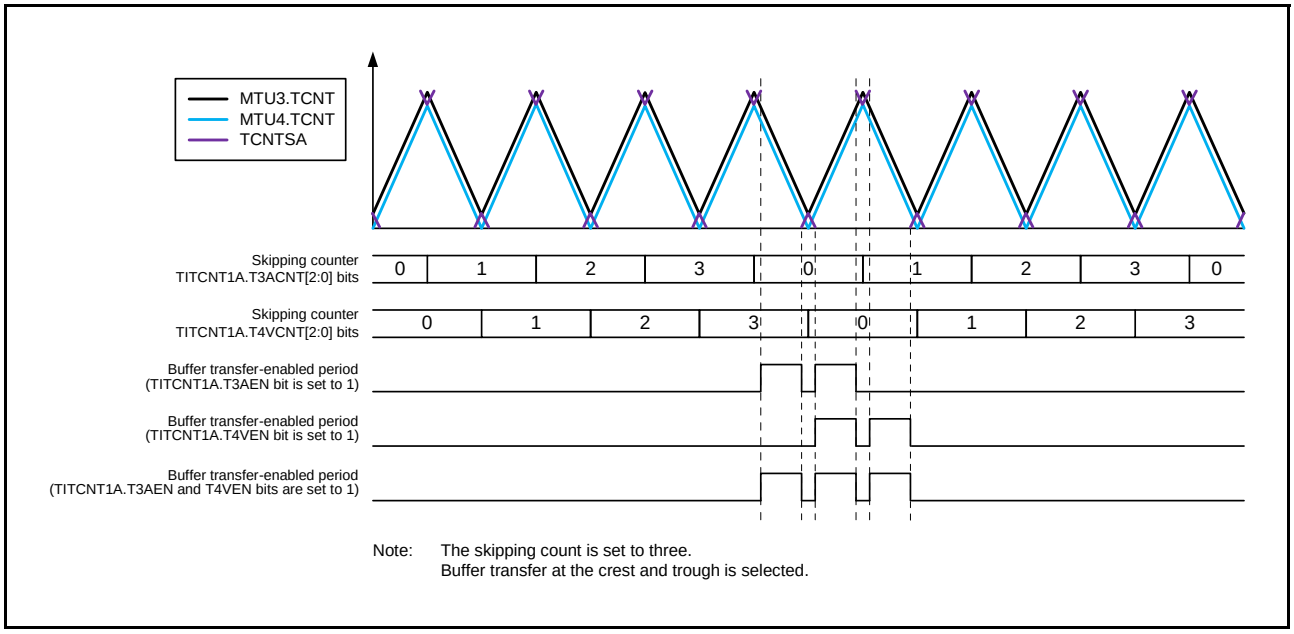


Figure 24.91 Relationship between Bits T3AEN and T4VEN in TITCR1A and Buffer Transfer-Enabled Period

(4) Complementary PWM Mode Output Protection Functions

The following output protection functions are provided for complementary PWM mode.

(a) Register and Counter Miswrite Prevention Function

Access from the CPU to the mode registers, control registers, compare registers, and counters can be enabled or disabled by setting the RWE bit in the TRWERA (TRWERB) register. The applicable registers are some of the registers in MTU3, MTU4, MTU6, and MTU7 shown below:
47 registers in total
MTU3.TCR, MTU4.TCR, MTU3.TCR2, MTU4.TCR2, MTU3.TMDR1, MTU4.TMDR1, MTU3.TIORH, MTU4.TIORH, MTU3.TIORL, MTU4.TIORL, MTU3.TIER, MTU4.TIER, MTU3.TCNT, MTU4.TCNT, MTU3.TGRA, MTU4.TGRA, MTU3.TGRB, MTU4.TGRB, MTU.TOERA, MTU.TOCR1A, MTU.TOCR2A, MTU.TGCRA, MTU.TCDRA, MTU.TDDRA,
MTU6.TCR, MTU7.TCR, MTU6.TCR2, MTU7.TCR2, MTU6.TMDR1, MTU7.TMDR1, MTU6.TIORH, MTU7.TIORH, MTU6.TIORL, MTU7.TIORL, MTU6.TIER, MTU7.TIER, MTU6.TCNT, MTU7.TCNT, MTU6.TGRA, MTU7.TGRA, MTU6.TGRB, MTU7.TGRB, MTU.TOERB, MTU.TOCR1B, MTU.TOCR2B, MTU.TCDRB, and MTU.TDDRB
This function can disable CPU access to the mode registers, control registers, and counters to prevent miswriting due to CPU runaway. In the access-disabled state, the applicable registers are read as undefined and writing to these registers is ignored.

(b) Halting of PWM Output by External Signal

The PWM output pins of MTU0, MTU3, MTU4, MTU6, and MTU7 can be set to the high-impedance state automatically.
Refer to section 25, Port Output Enable 3 (POE3a), for details.

24.3.9 A/D Conversion Start Request Delaying Function

A/D conversion start requests can be issued in MTU4 or MTU7 by making settings in the timer A/D conversion start request control register (MTU4.TADCR or MTU7.TADCR), timer A/D conversion start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB, or MTU7.TADCORA and MTU7.TADCORB), and timer A/D conversion start request cycle set buffer registers (MTU4.TADCOBRA and MTU4.TADCOBRB, or MTU7.TADCOBRA and MTU7.TADCOBRB).

The A/D conversion start request delaying function compares MTU4.TCNT with MTU4.TADCORA or MTU4.TADCORB (MTU7.TCNT with MTU7.TADCORA or MTU7.TADCORB), and when their values match, the function issues a respective A/D conversion start request (TRG4AN or TRG4BN (TRG7AN or TRG7BN)).

A/D conversion start requests (TRG4AN and TRG4BN (TRG7AN and TRG7BN)) can be skipped in coordination with interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in MTU4.TADCR (the ITA6AE, ITA7VE, ITB6AE, and ITB7VE bits in MTU7.TADCR).

(1) Example of Procedure for Specifying A/D Conversion Start Request Delaying Function

Figure 24.92 shows an example of procedure for specifying the A/D conversion start request delaying function.

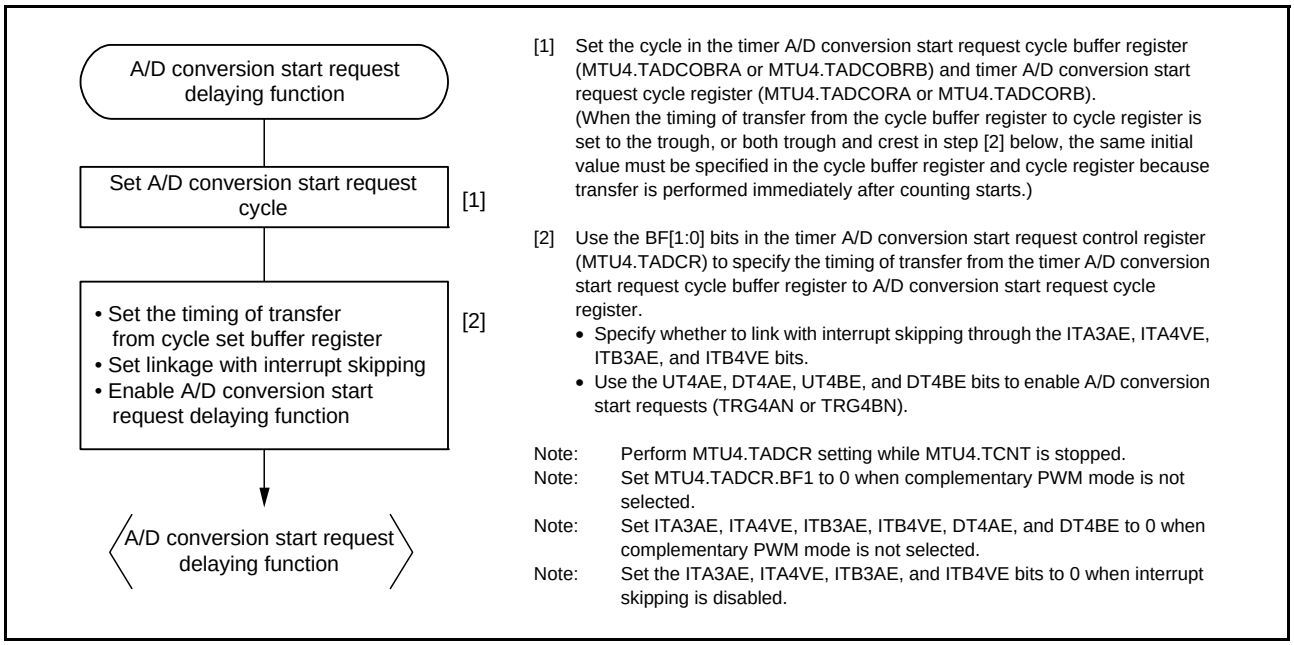
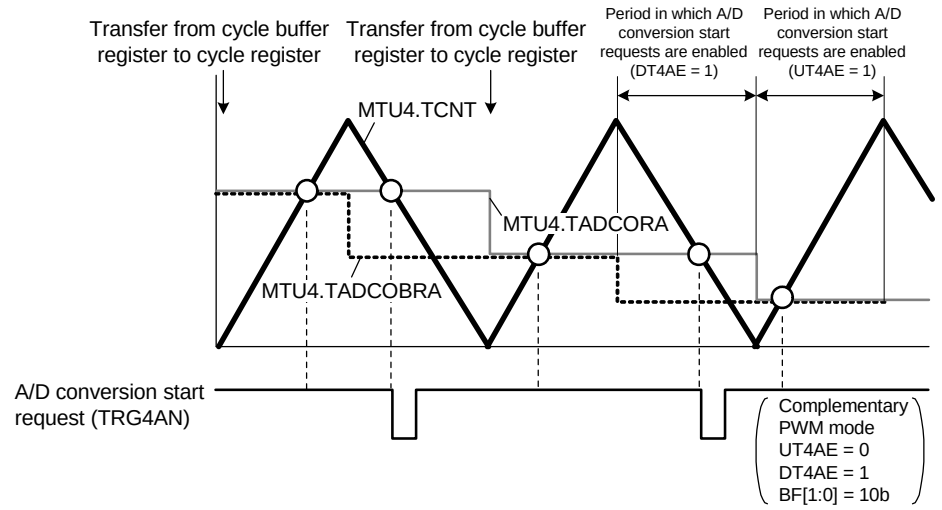


Figure 24.92 Example of Procedure for Specifying A/D Conversion Start Request Delaying Function (MTU3 and MTU4)

(2) Basic Example of A/D Conversion Start Request Delaying Function Operation

Figure 24.93 shows a basic example of A/D conversion start request signal (TRG4AN (TRG7AN)) operation when the trough of MTU4.TCNT (MTU7.TCNT) is specified for the buffer transfer timing and an A/D conversion start request signal is output during MTU4.TCNT (MTU7.TCNT) down-counting.



Period in which A/D conversion start requests are enabled (UT4AE = 1): 0 ≤ MTU4.TCNT ≤ TCDR – 1
Period in which A/D conversion start requests are enabled (DT4AE = 1): TCDR ≥ MTU4.TCNT ≥ 1

UT4AE, DT4AE, BF[1:0]: Bits in TADCR

Figure 24.93 Basic Example of A/D Conversion Start Request Signal (TRG4AN) Operation

(3) Period in Which A/D Conversion Start Requests are Enabled

When the MTU4.TCNT (MTU7.TCNT) counter and the MTU4.TADCORA or MTU4.TADCORB (MTU7.TADCORA or MTU7.TADCORB) register matches within the period enabled by the UT4AE and UT4BE (UT7AE and UT7BE) bits, the corresponding A/D conversion start request (TRG4AN or TRG4BN) is issued.

When the UT4AE and UT4BE (UT7AE and UT7BE) bits in the MTU4.TADCR (MTU7.TADCR) register are set to 1 in complementary PWM mode, A/D conversion start requests are enabled during the MTU4.TCNT (MTU7.TCNT) up-counting (0 ≤ MTU4.TCNT (MTU7.TCNT) ≤ TCDR – 1). When the DT4AE and DT4BE (DT7AE and DT7BE) bits in the MTU4.TADCR (MTU7.TADCR) register are set to 1, A/D conversion start requests are enabled during MTU4.TCNT (MTU7.TCNT) down-counting (TCDR ≥ MTU4.TCNT (MTU7.TCNT) ≥ 1). Refer to Figure 24.93.

(4) Buffer Transfer

The data in the timer A/D conversion start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB, or MTU7.TADCORA and MTU7.TADCORB) is updated by writing data to the timer A/D conversion start request cycle set buffer registers (MTU4.TADCOBRA and MTU4.TADCOBRB, or MTU7.TADCOBRA and MTU7.TADCOBRB). Data is transferred from the buffer registers to the respective cycle set registers at the timing selected with the BF[1:0] bits in the MTU4.TADCR (MTU7.TADCR) register.

In complementary PWM mode, data is also transferred from the timer A/D conversion start request cycle set buffer registers to the timer A/D conversion start request cycle set registers when MTU4.TGRD (MTU7.TGRD) register is updated.

There are notes on the timing for transferring data when using buffer transfer in complementary PWM mode.

For details, section 24.6.28, Usage Notes on A/D Conversion Start Request Delaying Function in Complementary PWM Mode.

In modes other than complementary PWM mode, set the BF1 bit in the MTU4.TADCR (MTU7.TADCR) register to 0.

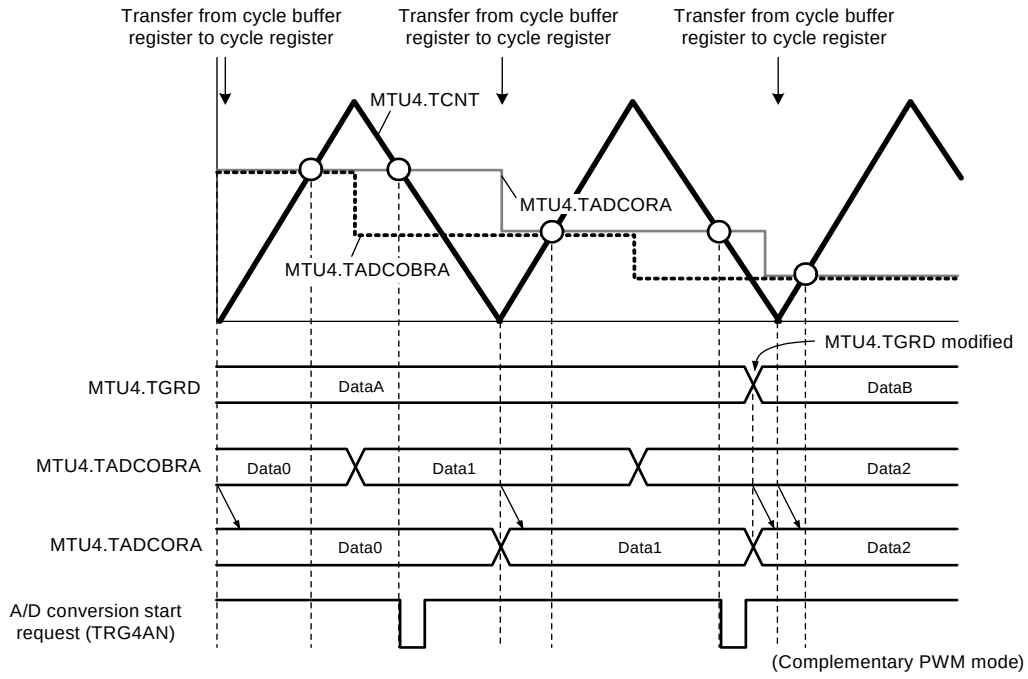


Figure 24.94 Example of A/D Conversion Start Request Signal (TRG4AN) and Buffer Transfer Operation

(5) A/D Conversion Start Request Delaying Function Linked with Interrupt Skipping Function 1

In complementary PWM mode, A/D conversion start requests (TRG4AN and TRG4BN (TRG7AN and TRG7BN)) can be issued in coordination with interrupt skipping 1 by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE (ITA6AE, ITA7VE, ITB6AE, and ITB7VE) bits in the MTU4.TADCR (MTU7.TADCR) register.

Figure 24.95 shows an example of A/D conversion start request signal (TRG4AN (TRG7AN)) operation when TRG4AN (TRG7AN) output is enabled during MTU4.TCNT (MTU7.TCNT) up-counting and down-counting and A/D conversion start requests are linked with interrupt skipping 1.

Figure 24.96 shows another example of A/D conversion start request signal (TRG4AN (TRG7AN)) operation when TRG4AN (TRG7AN) output is enabled during MTU4.TCNT (MTU7.TCNT) up-counting and A/D conversion start requests are linked with interrupt skipping 1.

In modes other than complementary PWM mode, do not use the A/D conversion start request delaying function linked with the interrupt skipping function 1.

Set the ITA3AE, ITA4VE, ITB3AE, and ITB4VE (ITA6AE, ITA7VE, ITB6AE, and ITB7VE) bits in the MTU4.TADCR (MTU7.TADCR) register to 0.

Note: This function should be used in combination with interrupt skipping 1.
When interrupt skipping is disabled (the T3AEN and T4VEN (T6AEN and T7VEN) bits in the timer interrupt skipping set register (TITCR1A (TITCR1B)) are set to 0 or the skipping count set bits (T3ACOR and T4VCOR (T6ACOR and T7VCOR)) in TITCR1A (TITCR1B) are set to 0), make sure that A/D conversion start requests are not linked with interrupt skipping 1 (set the ITA3AE, ITA4VE, ITB3AE, and ITB4VE (ITA6AE, ITA7VE, ITB6AE, and ITB7VE) bits in the timer A/D conversion start request control register (MTU4.TADCR (MTU7.TADCR)) to 0).
When this function is used, MTU4.TADCORA and MTU4.TADCORB (MTU7.TADCORA and MTU7.TADCORB) should be set with the value ranging 0002h to the value set in TCDRA minus 2 (value set in TCDRB minus 2).

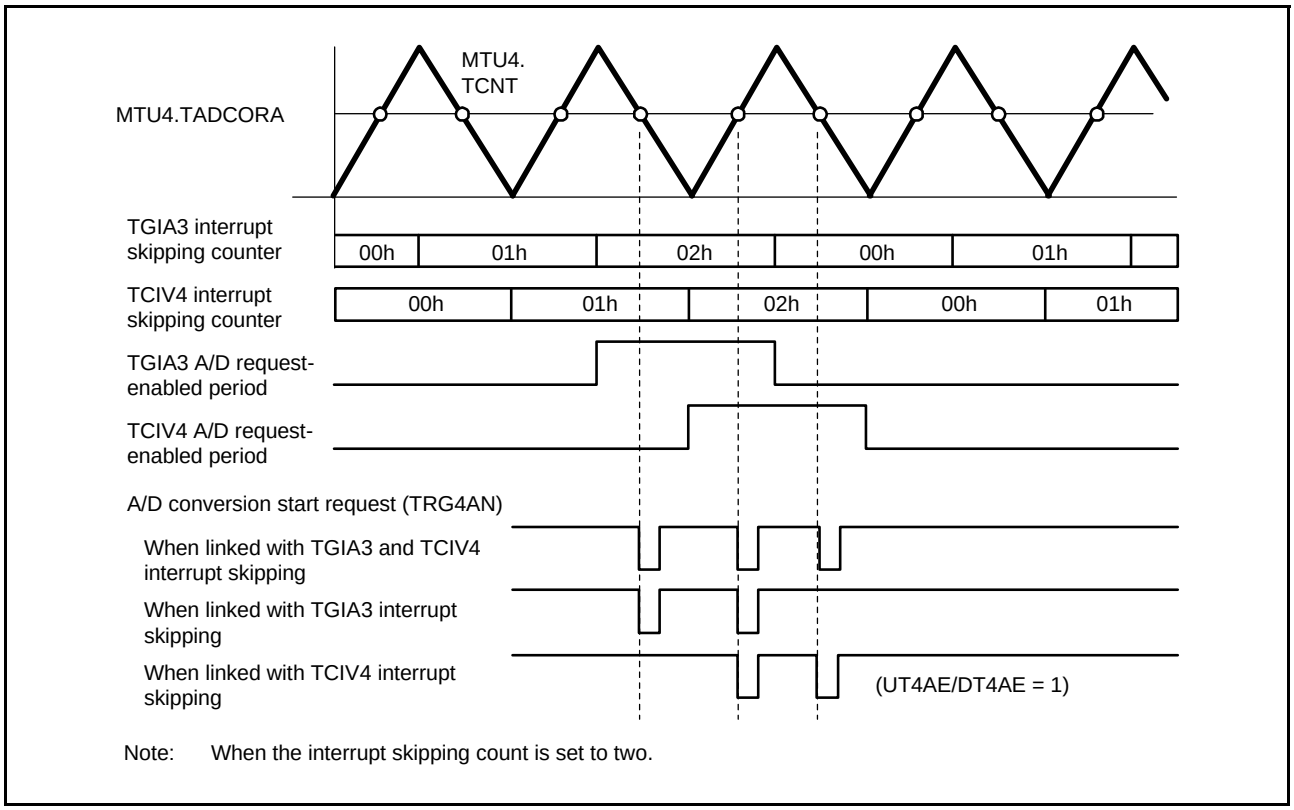


Figure 24.95 Example of A/D Conversion Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping Function 1 (UT4AE and DT4AE = 1)

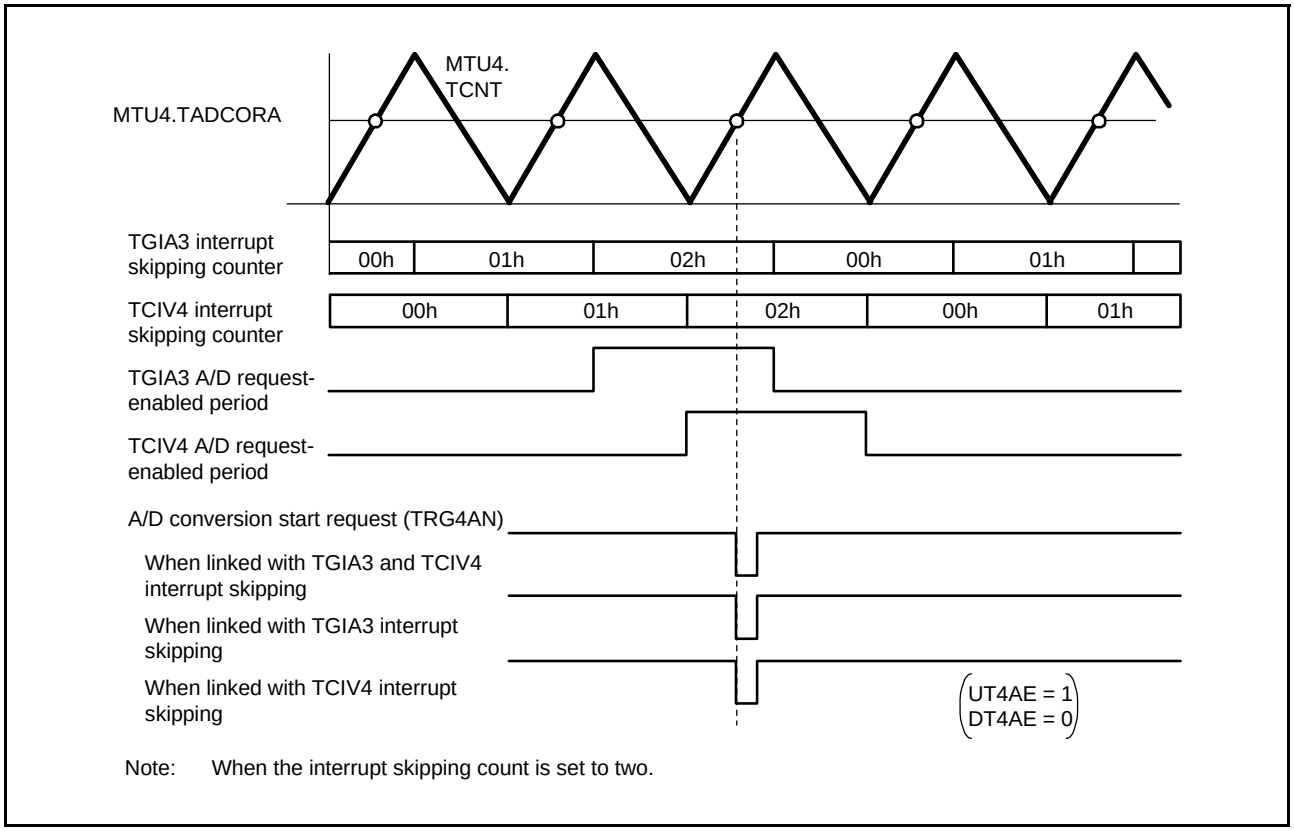


Figure 24.96 Example of A/D Conversion Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping Function 1 (UT4AE = 1, DT4AE = 0)

(6) A/D Conversion Start Request Delaying Function Linked with Interrupt Skipping Function 2

By setting the TITM bit to 1 in the TITMRA (TITMRB) register, the counter starts down-counting from the value (0 to 7) set in the TRG4COR[2:0] (TRG7COR[2:0]) bits in TITCR2A (TITCR2B) register every time an A/D conversion start trigger (TRG4AN or TRG4BN (TRG7AN or TRG7BN)) is generated. When the counter value reaches 0 and is reloaded, the TRG4AN and TRG4BN (TRG7AN and TRG7BN) interrupts become valid and an A/D conversion start request signal (TRG4ABN (TRG7ABN)) is output.
This function is valid only when the A/D conversion start request delaying function is enabled.

(a) Example of Procedure for Setting Interrupt Skipping Function 2

Figure 24.97 shows an example of procedure for setting interrupt skipping function 2.

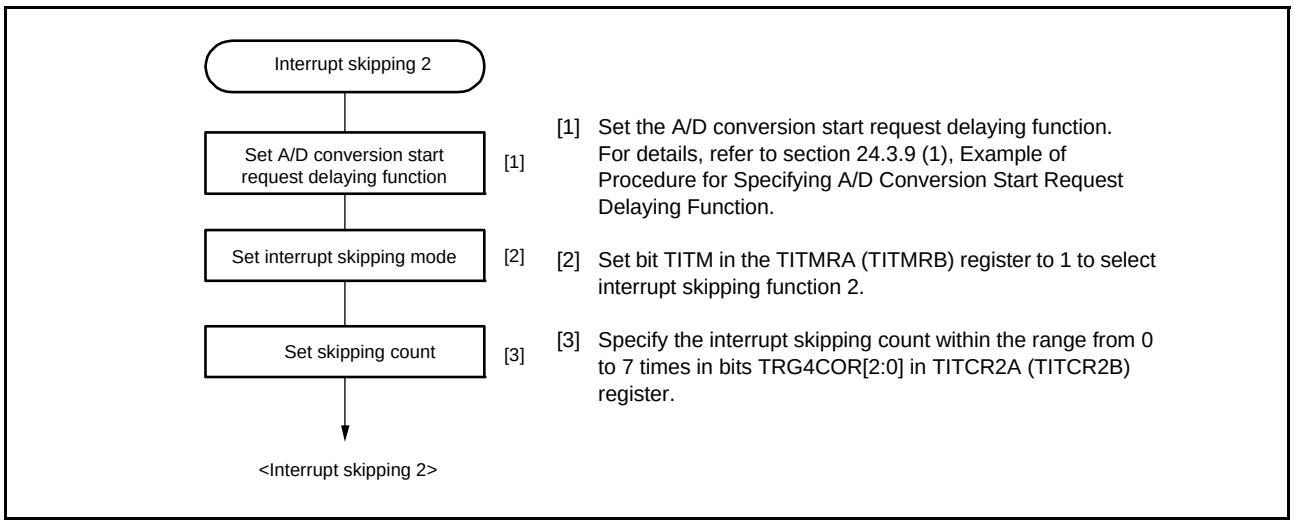


Figure 24.97 Example of Procedure for Setting Interrupt Skipping Function 2

(b) Example of Interrupt Skipping Function 2 Operation

Figure 24.98 shows an example of interrupt skipping function 2 operation.

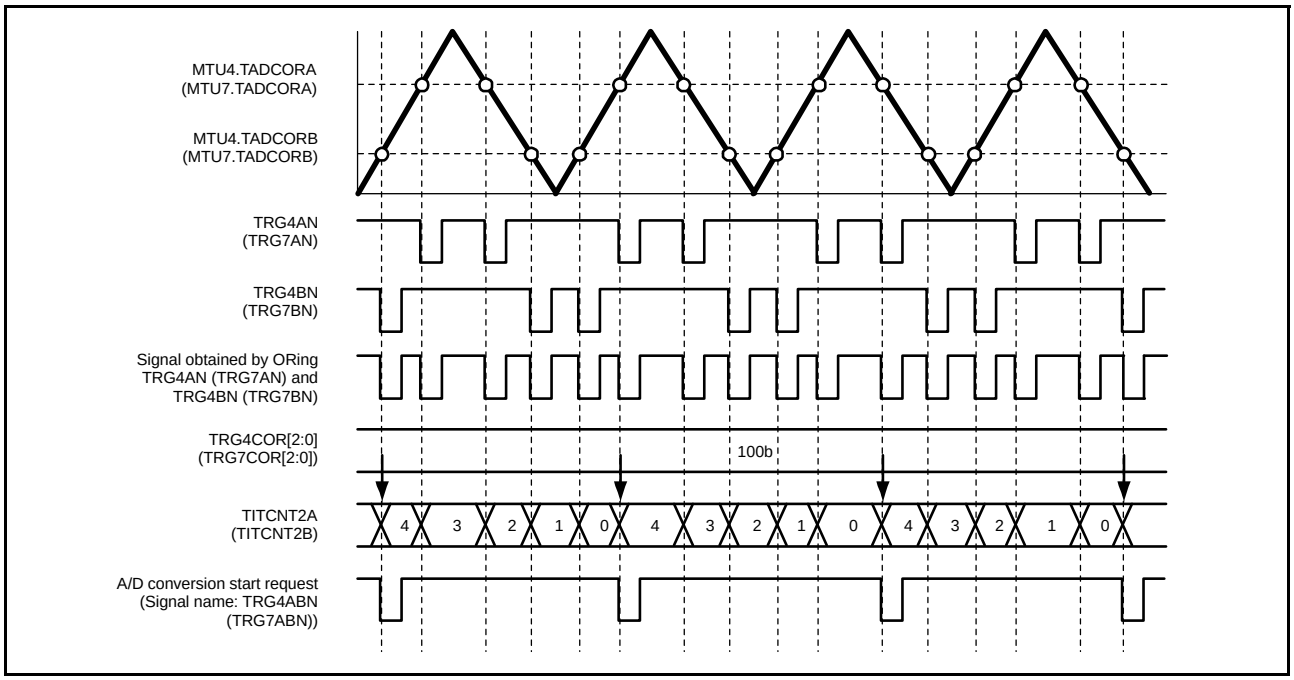


Figure 24.98 Example of Interrupt Skipping Function 2 Operation (Skipping Count is Set to Four)

24.3.10 Synchronous Operation of MTU0 to MTU4, MTU6, and MTU7

(1) Synchronous Counter Start for MTU0 to MTU4, MTU6, and MTU7

The counters in MTU0 to MTU4, MTU6, and MTU7 can be started synchronously by making the TCSYSTR settings.

(a) Example of Procedure for Setting Synchronous Counter Start for MTU0 to MTU4, MTU6, and MTU7

Figure 24.99 shows an example of procedure for setting synchronous counter start for MTU0 to MTU4, MTU6, and MTU7.

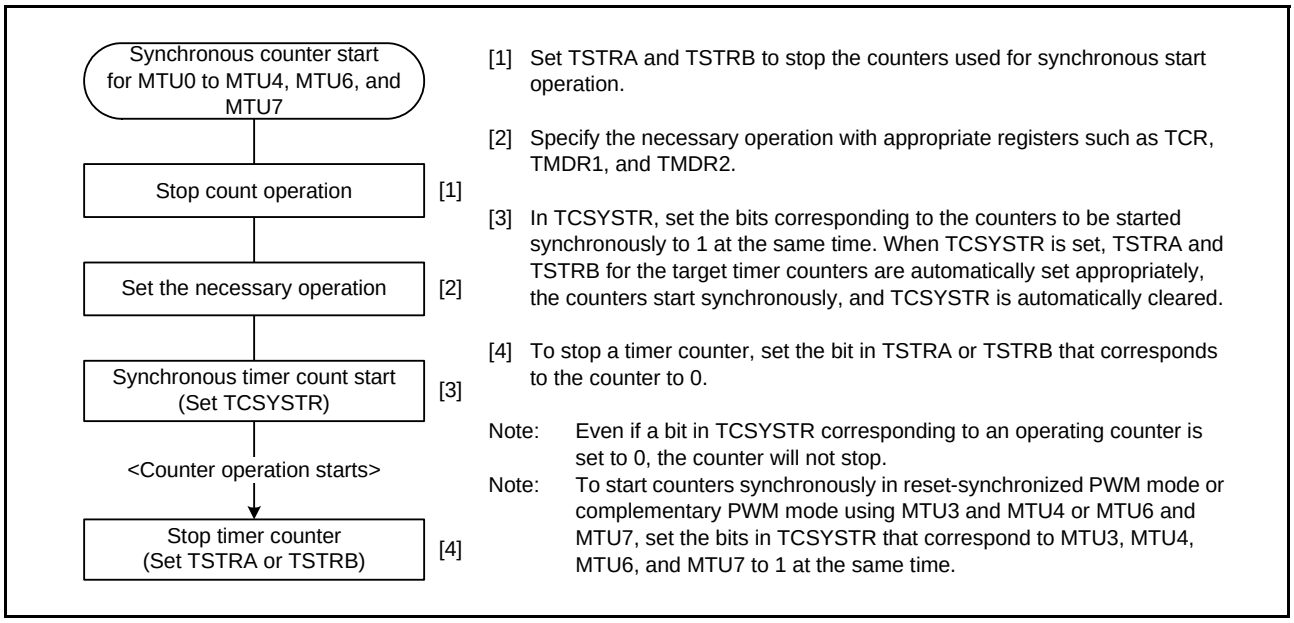


Figure 24.99 Example of Procedure for Setting Synchronous Counter Start for MTU0 to MTU4, MTU6, and MTU7

(b) Examples of Synchronous Counter Start Operation

Figure 24.100 shows an examples of synchronous counter start operation for MTU0 to MTU4, MTU6, and MTU7.

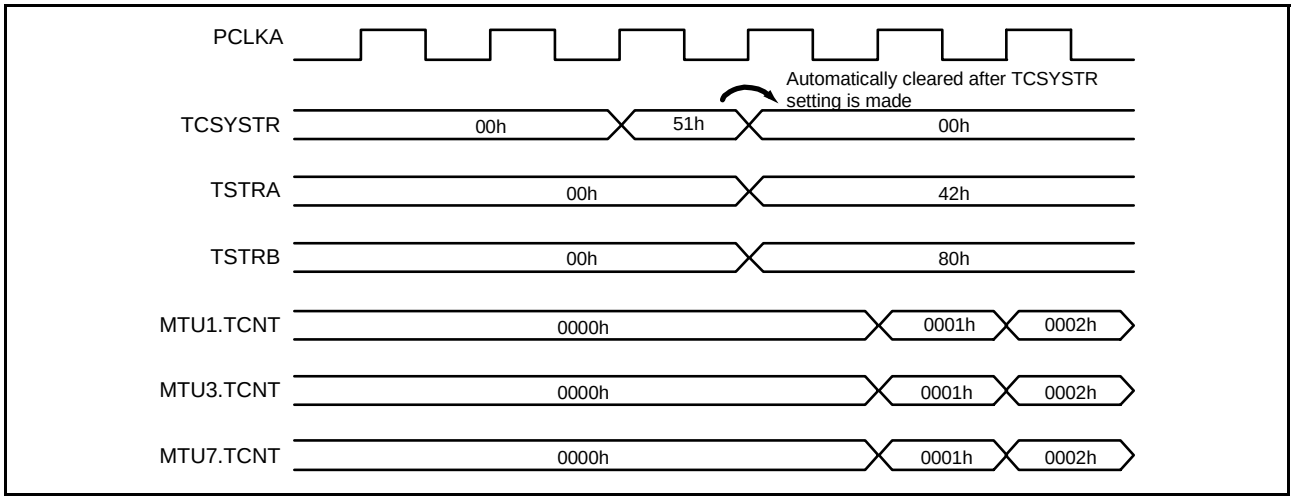


Figure 24.100 Examples of Synchronous Counter Start Operation for MTU0 to MTU4, MTU6, and MTU7

(2) Synchronous Counter Clearing for MTU6 and MTU7

The counters in MTU6 and MTU7 can be cleared by the TGImn interrupt generation timing (m = A to D; n = 0 to 2) through the TSYCR setting.

(a) Example of Procedure for Specifying Synchronous Counter Clearing for MTU6 and MTU7

Figure 24.101 shows an example of procedure for specifying synchronous counter clearing for MTU6 and MTU7 by interrupt generation timing.

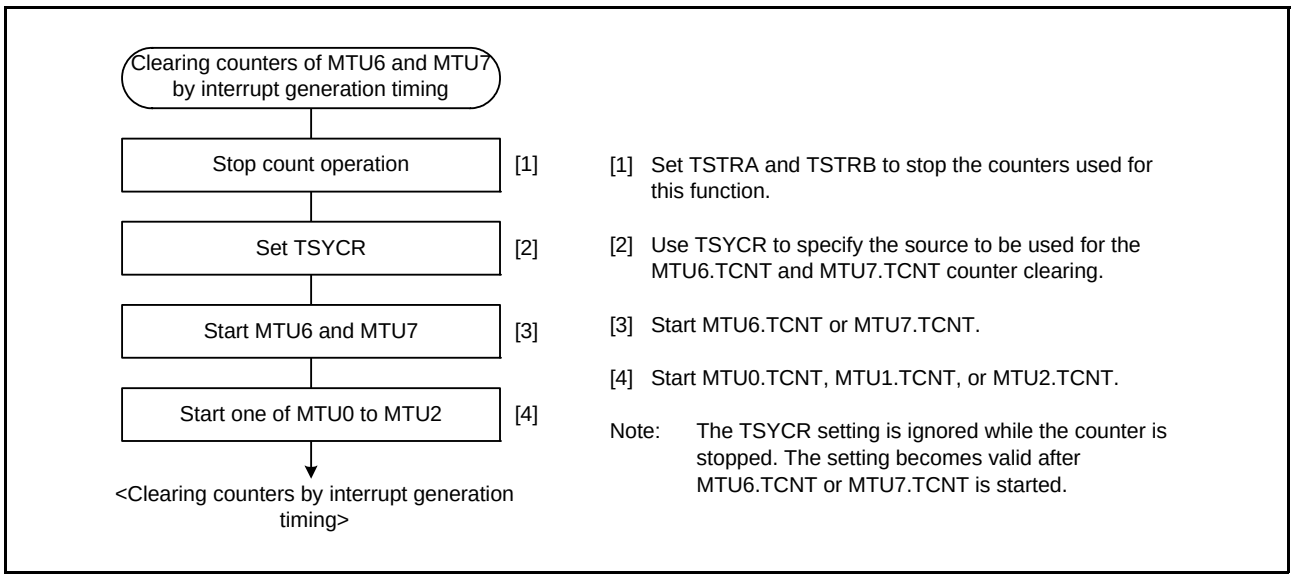


Figure 24.101 Example of Procedure for Specifying Synchronous Counter Clearing for MTU6 and MTU7

(b) Examples of Synchronous Counter Clearing for MTU6 and MTU7

Figure 24.102 and Figure 24.103 show examples of synchronous counter clearing for MTU6 and MTU7 by interrupt generation timing.

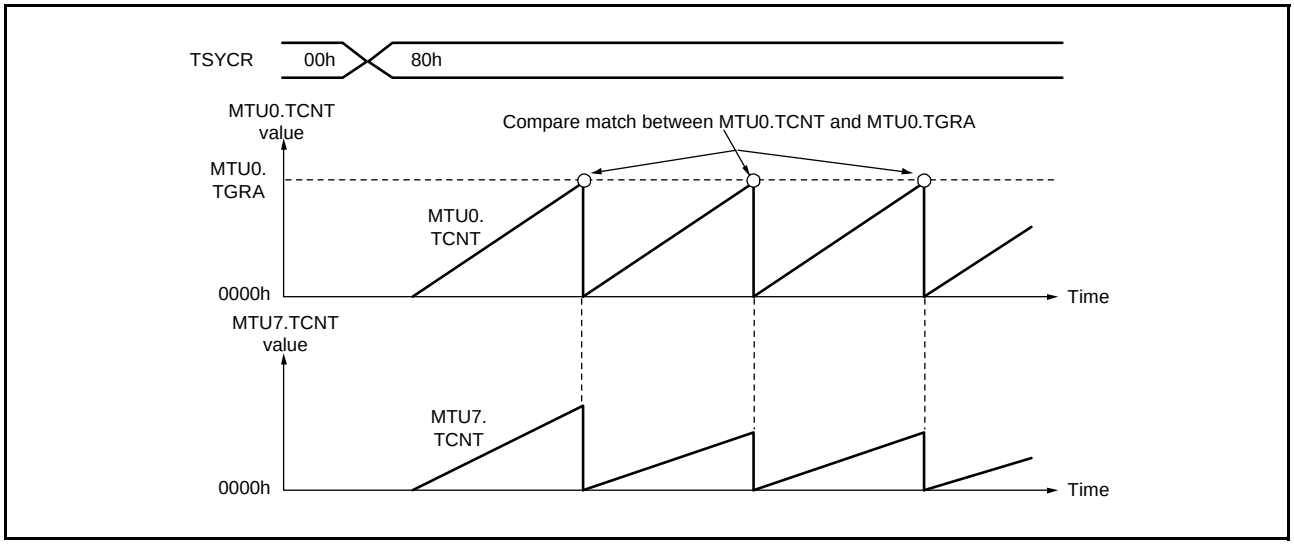


Figure 24.102 Example of Synchronous Counter Clearing for MTU6 and MTU7 (1)

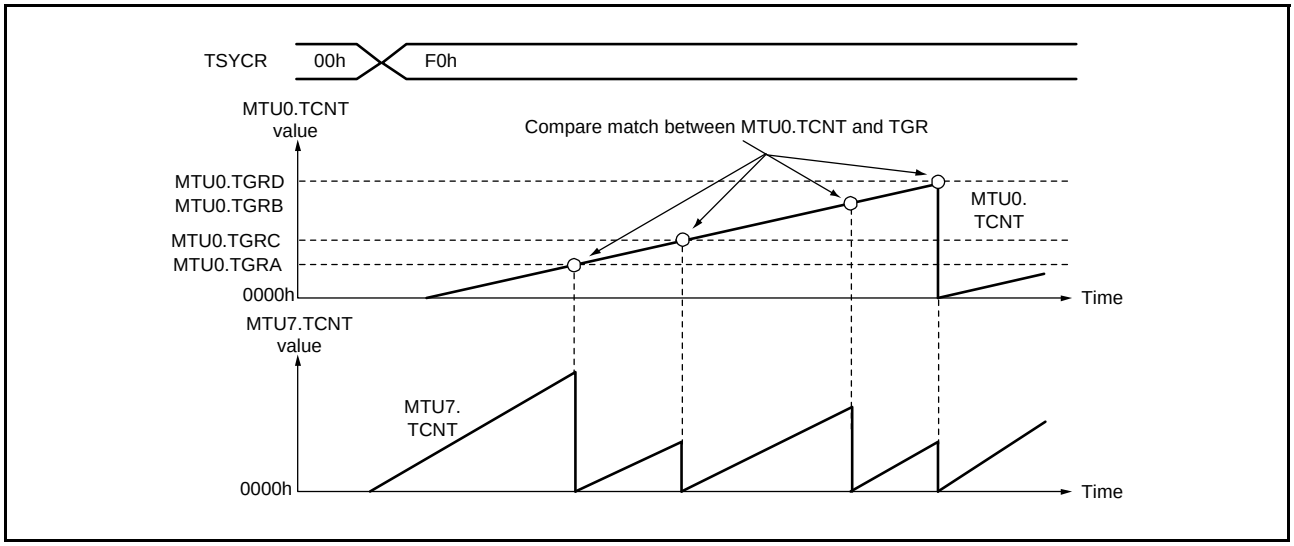


Figure 24.103 Example of Synchronous Counter Clearing for MTU6 and MTU7 (2)

24.3.11 External Pulse Width Measurement

The pulse widths of up to three external input lines can be measured in MTU5.
When the IOC[4:0] bits in MTU5.TIORU, MTU5.TIORV, MTU5.TIORW are set for pulse width measurement, the pulse width of the signal input to the MTIC5U, MTIC5V, and MTIC5W pins are measured. TCNTU, TCNTV, and TCNTW count up while the level specified by the IOC[4:0] bits is input.
Figure 24.104 shows an example of setting external pulse width measurement, and Figure 24.105 an example of external pulse width measurement.

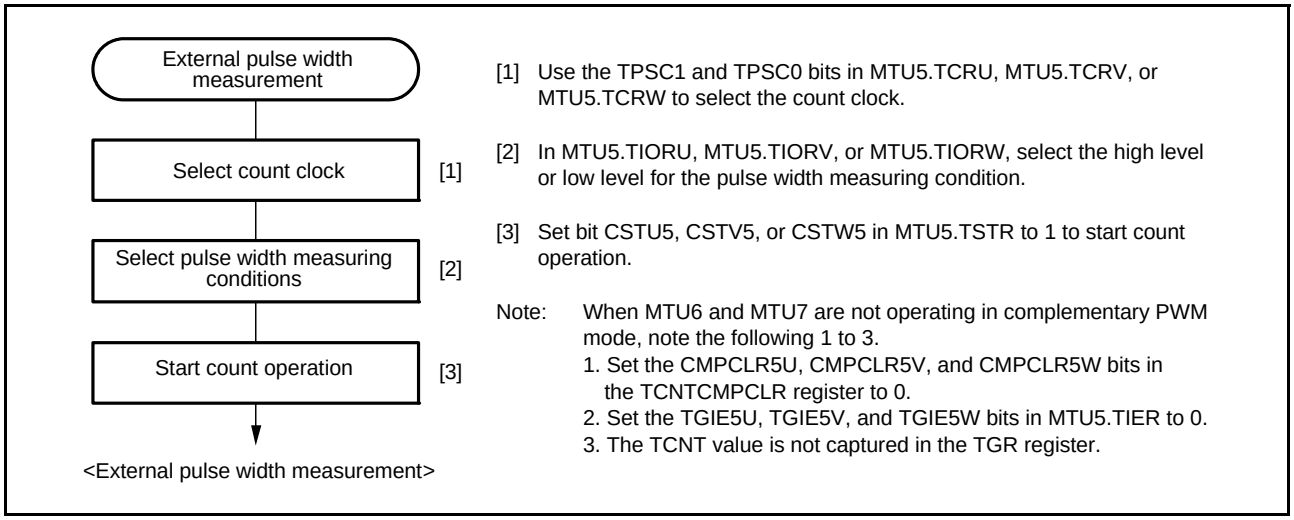


Figure 24.104 Example of External Pulse Width Measurement Setting Procedure

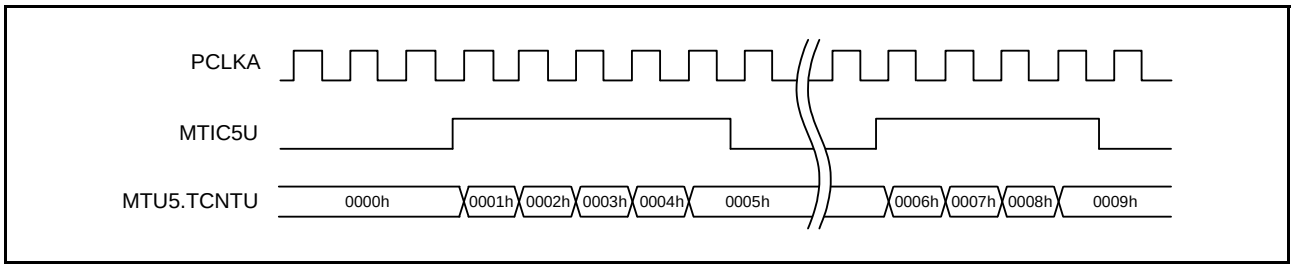


Figure 24.105 Example of External Pulse Width Measurement (Measuring High Pulse Width)

24.3.12 Dead Time Compensation

A dead time delay (a propagation delay of the inverter output from the complementary PWM output) can be compensated by combining MTU5 with MTU6 and MTU7. Figure 24.106 shows an example of the motor control circuit compensating a dead time delay by combining MTU5 with MTU6 and MTU7. A dead time for the PWM output waveform during complementary PWM operation using MTU6 and MTU7 can be compensated by adjusting a duty ratio set in a compare register for the PWM output after measuring a delay of the inverter output from the complementary PWM output by an external pulse measurement function for MTU5 (Figure 24.107). Figure 24.108 shows the procedure for setting dead time compensation using MTU5 to MTU7. For details on MTU5 operation at this time, refer to section 24.3.13, TCNTU, TCNTV, and TCNTW Capture at Crest and/or Trough in Complementary PWM Mode.

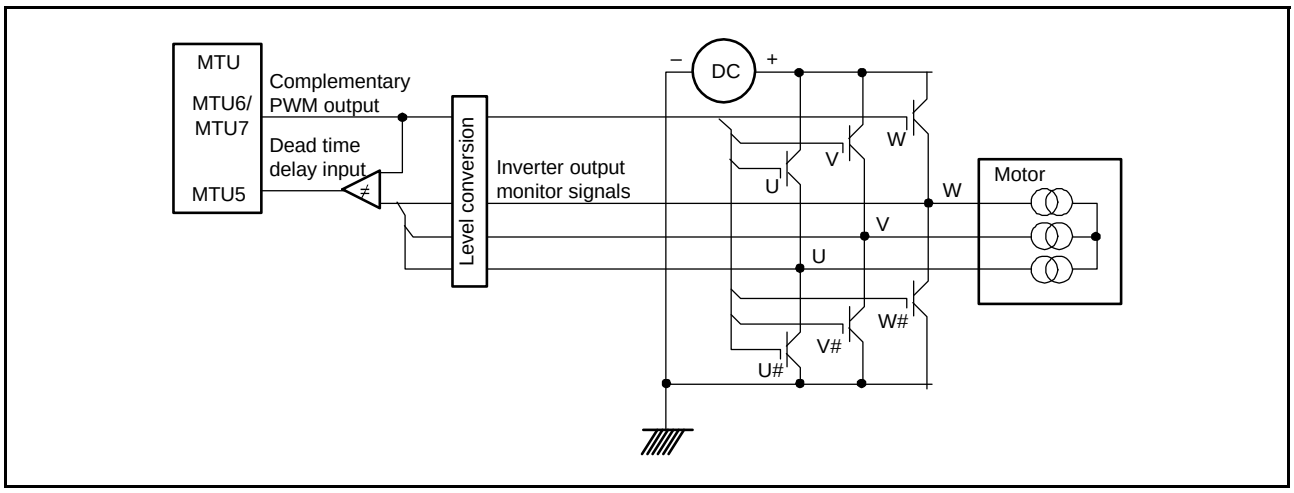


Figure 24.106 Motor Control Circuit Example

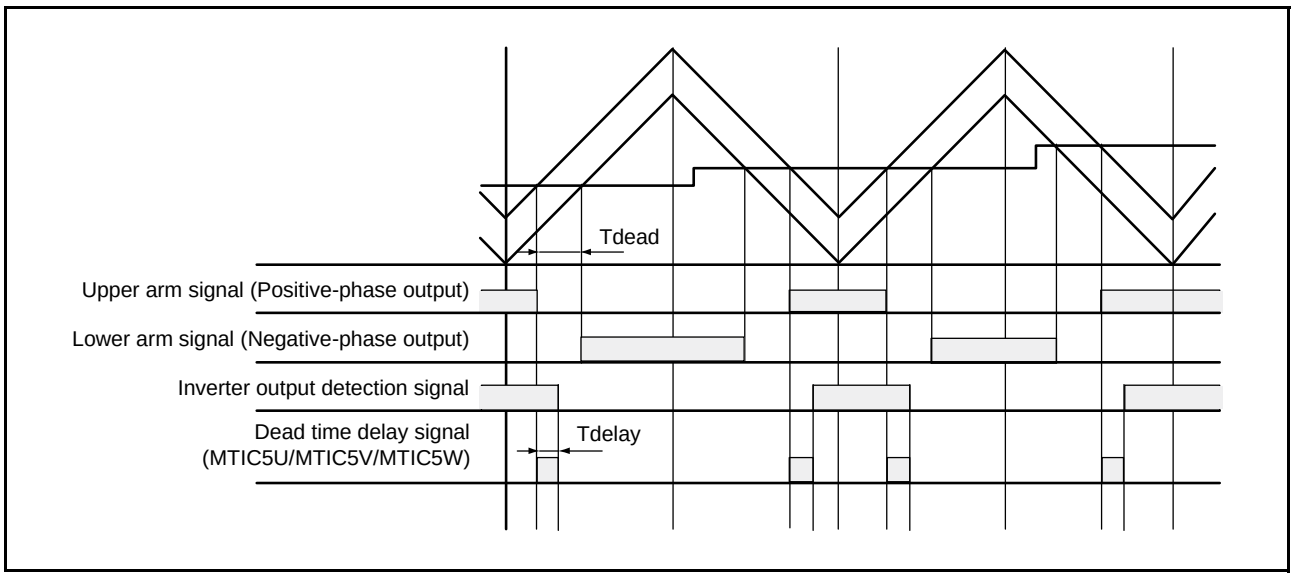
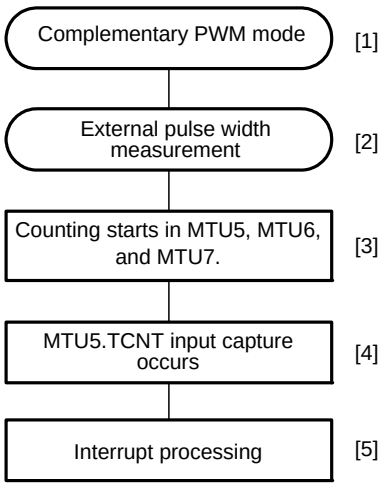


Figure 24.107 Delay in Dead Time in Complementary PWM Operation

(1) Example of Dead Time Compensation Setting Procedure

Figure 24.108 shows an example of dead time compensation setting procedure by using three counters in MTU5.



[1] Set MTU6 and MTU7 to complementary PWM mode. For details, refer to section 24.3.8, Complementary PWM Mode.

[2] Specify the external pulse width measurement function for the target TIOR in MTU5. For details, refer to section 24.3.11, External Pulse Width Measurement.

[3] Set bits CST6 and CST7 in TSTRB and bits CSTU5, CSTV5, and CSTW5 in MTU5.TSTR to 1 to start count operation.

[4] When the capture condition specified in TIOR is satisfied, the MTU5.TCNT value is captured in MTU5.TGR.
To prevent the generation of the MTU5.TGIm5 interrupt (m = U, V, W) request under the capture condition specified in TIOR, do not enable interrupt requests in MTU5.TIER.

[5] For U-phase dead time compensation, when an interrupt is generated at the crest (TGIA6) or trough (TCIV7) in complementary PWM mode, read the MTU5.TGRU value, calculate the difference in time between MTU6.TGRB and MTU5.TGRU, and write the corrected value to MTU6.TGRD in the interrupt processing.
For the V phase and W phase, read the MTU5.TGRV and MTU5.TGRW values and write the corrected values to MTU7.TGRC and TGRD, respectively, in the same way as for U-phase compensation.
The MTU5.TCNT value should be cleared through the TCNTCMPCLR setting or by software.

Note: The MPC and port mode registers (PMR) of the I/O ports should be set up in advance.
As the interrupt flag is set under the capture condition specified in TIOR, do not enable interrupt requests in MTU5.TIER.

Figure 24.108 Example of Dead Time Compensation Setting Procedure

24.3.13 TCNTU, TCNTV, and TCNTW Capture at Crest and/or Trough in Complementary PWM Mode

The MTU5 external pulse width measurement function allows to transfer the value in TCNTU, TCNTV, and TCNTW to TGRU, TGRV, and TGRW at the crest, or trough, or crest and trough when MTU6 and MTU7 operate in complementary PWM mode. The transfer timing is set in TIORU, TIORV, and TIORW. When the CMPCLR5U, CMPCLR5V, and CMPCLR5W bits in the TCNTCMPCLR register are set to 1, TCNTU, TCNTV, and TCNTW become 0000h at the transfer timing for TGRU, TGRV, and TGRW.

When MTU3 and MTU4 operate in complementary PWM mode, MTU5 cannot operate a capture operation of TCNTU, TCNTV, and TCNTW at the crest, or trough, or crest and trough in complementary PWM mode.

Figure 24.109 shows an operation example in which TCNTU is used as a free-running counter without being cleared, and the value is captured in TGRU at the crest and trough in complementary PWM mode.

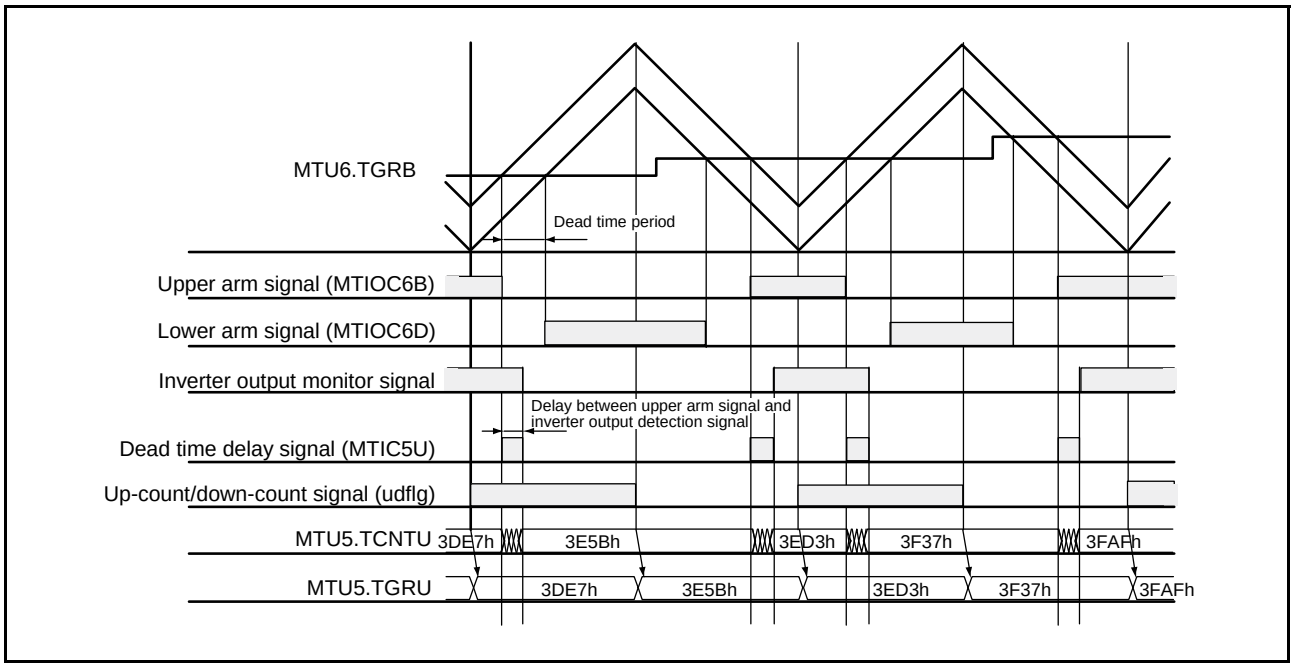


Figure 24.109 TCNTU Capture at Crest and Trough in Complementary PWM Operation

24.3.14 Noise Filter Function

The input capture input pins and external pulse input pins have a noise filter function.
Set the NFCRn register (n = 0 to 7, C) to enable or disable the noise filter function and set the sampling clock. The noise filter for each pin can be enabled or disabled individually, and the sampling clock can be set for each channel.
Figure 24.110 shows the timing of noise filtering.

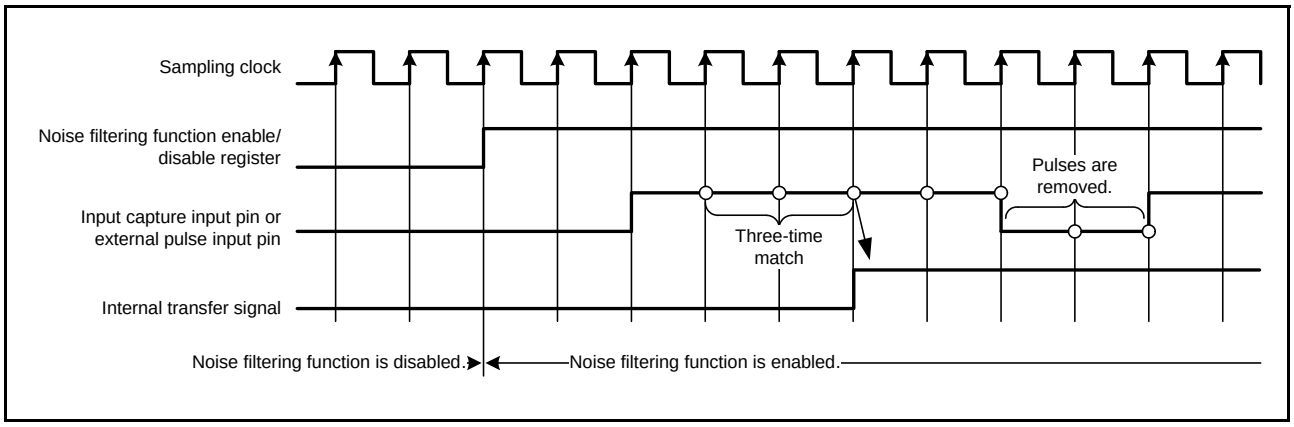


Figure 24.110 Timing of Noise Filtering

24.4 Interrupt Sources

24.4.1 Interrupt Sources and Priorities

There are three kinds of interrupt source; TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own enable/disable bit, allowing the generation of interrupt request signals to be enabled or disabled individually.

When an interrupt source is generated, if the corresponding enable/disable bit in TIER is set to 1, an interrupt is requested.

Relative channel priorities can be changed by the interrupt controller; however the priority within a channel is fixed. For details, refer to section 15, Interrupt Controller (ICUE). Table 24.77 lists the MTU interrupt sources.

Table 24.77 MTU Interrupt Sources

Channel	Name	Interrupt Source	DMAC/DTC Activation
MTU0	TGIA0	MTU0.TGRA input capture/compare match	Possible
	TGIB0	MTU0.TGRB input capture/compare match	Possible
	TGIC0	MTU0.TGRC input capture/compare match	Possible
	TGID0	MTU0.TGRD input capture/compare match	Possible
	TCIV0	MTU0.TCNT overflow	Not possible
	TGIE0	MTU0.TGRE compare match	Not possible
	TGIF0	MTU0.TGRF compare match	Not possible
MTU1	TGIA1	MTU1.TGRA input capture/compare match	Possible
	TGIB1	MTU1.TGRB input capture/compare match	Possible
	TCIV1	MTU1.TCNT overflow	Not possible
	TCIU1	MTU1.TCNT underflow	Not possible
MTU2	TGIA2	MTU2.TGRA input capture/compare match	Possible
	TGIB2	MTU2.TGRB input capture/compare match	Possible
	TCIV2	MTU2.TCNT overflow	Not possible
	TCIU2	MTU2.TCNT underflow	Not possible
MTU3	TGIA3	MTU3.TGRA input capture/compare match	Possible
	TGIB3	MTU3.TGRB input capture/compare match	Possible
	TGIC3	MTU3.TGRC input capture/compare match	Possible
	TGID3	MTU3.TGRD input capture/compare match	Possible
	TCIV3	MTU3.TCNT overflow	Not possible
MTU4	TGIA4	MTU4.TGRA input capture/compare match	Possible
	TGIB4	MTU4.TGRB input capture/compare match	Possible
	TGIC4	MTU4.TGRC input capture/compare match	Possible
	TGID4	MTU4.TGRD input capture/compare match	Possible
	TCIV4	MTU4.TCNT overflow/underflow*1	Possible
MTU5	TGIU5	MTU5.TGRU input capture/compare match	Possible
	TGIV5	MTU5.TGRV input capture/compare match	Possible
	TGIW5	MTU5.TGRW input capture/compare match	Possible
MTU6	TGIA6	MTU6.TGRA input capture/compare match	Possible
	TGIB6	MTU6.TGRB input capture/compare match	Possible
	TGIC6	MTU6.TGRC input capture/compare match	Possible
	TGID6	MTU6.TGRD input capture/compare match	Possible
	TCIV6	MTU6.TCNT overflow	Not possible
MTU7	TGIA7	MTU7.TGRA input capture/compare match	Possible
	TGIB7	MTU7.TGRB input capture/compare match	Possible
	TGIC7	MTU7.TGRC input capture/compare match	Possible
	TGID7	MTU7.TGRD input capture/compare match	Possible
	TCIV7	MTU7.TCNT overflow/underflow*1	Possible
MTU8	TGIA8	MTU8.TGRA input capture/compare match	Possible
	TGIB8	MTU8.TGRB input capture/compare match	Possible
	TGIC8	MTU8.TGRC input capture/compare match	Possible
	TGID8	MTU8.TGRD input capture/compare match	Possible
	TCIV8	MTU8.TCNT overflow	Not possible

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

Note 1. Underflow is available only in complementary PWM mode.

(1) Input Capture/Compare Match Interrupt

If the TIER.TGIE bit is set to 1 when a TGR input capture/compare match occurs on a channel, an interrupt is requested. The MTU has 33 input capture/compare match interrupts (six for MTU0, four each for MTU3, MTU4, MTU6, MTU7, and MTU8, two each for MTU1 and MTU2, and three for MTU5).

(2) Overflow Interrupt

If the TIER.TCIEV bit is set to 1 when a TCNT overflow occurs on a channel, an interrupt is requested. The MTU has eight overflow interrupts (one for each channel except MTU5).
Note that an overflow interrupt is generated also when an underflow of the MTU4.TCNT and MTU7.TCNT occurs while operating in complementary PWM mode.

(3) Underflow Interrupt

If the TIER.TCIEU bit is set to 1 when a TCNT underflow occurs on a channel, an interrupt is requested. The MTU has two underflow interrupts (one each for MTU1, and MTU2).

24.4.2 DTC/DMAC Trigger Sources

(1) DTC Trigger Sources

The DTC can be triggered by the TGR input capture/compare match interrupt in each channel or the overflow interrupt in MTU4 and MTU7. For details, refer to section 20, Data Transfer Controller (DTCb).
The MTU provides a total of 33 input capture/compare match interrupts and overflow interrupts that can be used as DTC trigger sources: four each for MTU0, MTU3, MTU6, and MTU8, two each for MTU1 and MTU2, five each for MTU4 and MTU7, and three for MTU5.

(2) DMAC Trigger Sources

The DMAC can be triggered by the TGR input capture/compare match interrupt in each channel and the overflow interrupt in MTU4 and MTU7. For details, refer to section 18, DMA Controller (DMACAb).
The MTU provides a total of 33 input capture/compare match interrupts and overflow interrupts that can be used as DMAC trigger sources: four each for MTU0, MTU3, MTU6, and MTU8, two each for MTU1 and MTU2, five each for MTU4 and MTU7, and three for MTU5.
If a DMA transfer is initiated by the MTU, the trigger signal is cleared when the DMAC requests ownership of the internal bus. Accordingly, the state of the internal bus may lead to a wait for the start of DMA transfer.

(3) EXDMAC Initiation

The EXDMAC can be triggered by MTU interrupts. For details, refer to section 19, EXDMA Controller (EXDMACa).

24.4.3 A/D Converter Trigger Sources

The A/D converter can be triggered by one of the following three methods in the MTU. Table 24.78 shows the relationship between interrupt sources and A/D conversion start request signals.

(1) A/D Conversion Start by TGRA Input Capture/Compare Match or at Trough of MTU4.TCNT (MTU7.TCNT) in Complementary PWM Mode

The A/D converter can be triggered by the occurrence of a TGRA input capture/compare match in each channel. In addition, if complementary PWM operation is performed while the TTGE2 bit in MTU4.TIER (MTU7.TIER) is set to 1, the A/D converter can be triggered at the trough of MTU4.TCNT (MTU7.TCNT) count (MTU4.TCNT (MTU7.TCNT) = 0000h).
A/D conversion start request TRGAnN is issued to the A/D converter under either of the following conditions (n = 0 to 4, 6, 7).

- When a TGRA input capture/compare match occurs on a channel while the TIER.TTGE bit is set to 1
- When the MTU4.TCNT (MTU7.TCNT) count reaches the trough (MTU4.TCNT (MTU7.TCNT) = 0000h) during complementary PWM operation while the TTGE2 bit in MTU4.TIER (MTU7.TIER) is set to 1

When either condition is satisfied, if A/D conversion start request signal TRGAnN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

(2) A/D Conversion Start by Compare Match between MTU0.TCNT and MTU0.TGRE

A/D conversion start request TRG0N is issued to the A/D converter when a compare match occurs between MTU0.TCNT and MTU0.TGRE.
When a compare match occurs between MTU0.TCNT and MTU0.TGRE while the TTGE2 bit in MTU0.TIER2 is set to 1, A/D conversion start request TRG0N is issued to the A/D converter. If A/D conversion start request signal TRG0N from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

(3) A/D Conversion Start by A/D Conversion Start Request Delaying Function

The A/D converter can be triggered by generating A/D conversion start request signal TRG4AN or TRG4BN (TRG7AN or TRG7BN) when the MTU4.TCNT (MTU7.TCNT) count matches the MTU4.TADCORA or MTU4.TADCORB (MTU7.TADCORA or MTU7.TADCORB) value if the UT4AE, DT4AE, UT4BE, or DT4BE (UT7AE, DT7AE, UT7BE, or DT7BE) bit in the A/D conversion start request control register (MTU4.TADCR (MTU7.TADCR)) is set to 1. For details, refer to section 24.3.9, A/D Conversion Start Request Delaying Function.
A/D conversion will start when TRG4AN (TRG7AN) is generated if A/D conversion start request signal TRG4AN (TRG7AN) from the MTU is selected as the trigger in the A/D converter, when TRG4BN (TRG7BN) is generated if TRG4BN (TRG7BN) from the MTU is selected as the trigger in the A/D converter, or when TRG4ABN (TRG7ABN) is generated if TRG4ABN (TRG7ABN) from the MTU is selected as the trigger in the A/D converter.

Table 24.78 Interrupt Sources and A/D Conversion Start Request Signals

Target Registers	Interrupt Source	A/D Conversion Start Request Signal
MTU0.TGRA and MTU0.TCNT	Input capture/compare match	TRGA0N
MTU1.TGRA and MTU1.TCNT		TRGA1N
MTU2.TGRA and MTU2.TCNT		TRGA2N
MTU3.TGRA and MTU3.TCNT		TRGA3N
MTU4.TGRA and MTU4.TCNT*1		TRGA4N
MTU4.TCNT	MTU4.TCNT trough in complementary PWM mode	
MTU6.TGRA and MTU6.TCNT	Input capture/compare match	TRGA6N
MTU7.TGRA and MTU7.TCNT*1		TRGA7N
MTU7.TCNT	MTU7.TCNT trough in complementary PWM mode	
MTU0.TGRE and MTU0.TCNT	Compare match	TRG0N
MTU4.TADCORA and MTU4.TCNT		TRG4AN
MTU4.TADCORB and MTU4.TCNT		TRG4BN
MTU7.TADCORA and MTU7.TCNT		TRG7AN
MTU7.TADCORB and MTU7.TCNT		TRG7BN
MTU4.TADCORA and MTU4.TCNT, MTU4.TADCORB and MTU4.TCNT	Compare match (interrupt skipping function 2)	TRG4ABN
MTU7.TADCORA and MTU7.TCNT, MTU7.TADCORB and MTU7.TCNT		TRG7ABN

Note 1. Since PWM waveforms are generated in complementary PWM mode, MTU4.TGRA (MTU7.TGRA) compare match not only with MTU4.TCNT (MTU7.TCNT) but also with MTU3.TCNT (MTU6.TCNT) and TCNTSA (TCNTSB) is detected. Accordingly, when compare match with MTU3.TCNT (MTU6.TCNT) and TCNTSA (TCNTSB) occurs, TRGA4N (TRGA7N) is also generated. When MTU3 and MTU 4 (MTU 6 and MTU7) are made to operate in complementary PWM mode for generating an A/D conversion start request, use the A/D conversion start request by compare match between MTU4.TCNT (MTU7.TCNT) and MTU4.TADCORA/TADCORB (MTU7.TADCORA/TADCORB).

24.5 Operation Timing

24.5.1 Input/Output Timing

(1) TCNT Count Timing

Figure 24.111 and Figure 24.112 show the TCNT count timing in internal clock operation, Figure 24.113 shows the TCNT count timing in external clock operation (normal mode), and Figure 24.114 shows the TCNT count timing in external clock operation (phase counting mode).

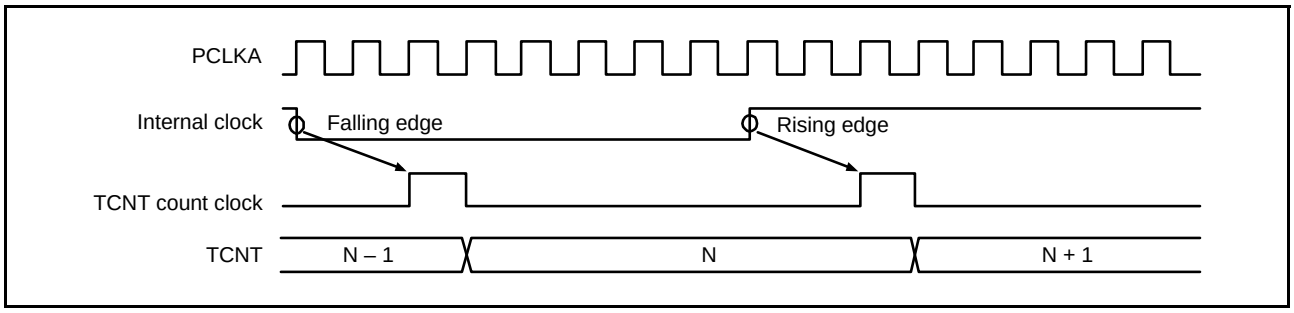


Figure 24.111 Count Timing in Internal Clock Operation (MTU0 to MTU4 and MTU6 to MTU8)

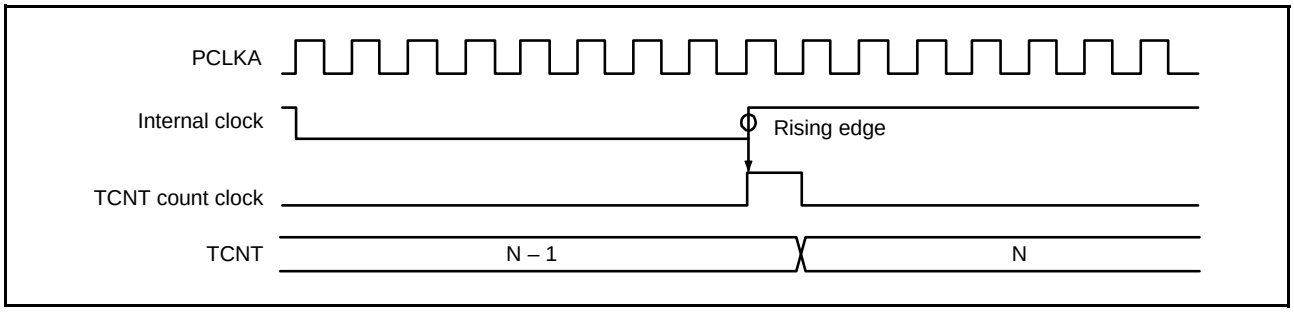


Figure 24.112 Count Timing in Internal Clock Operation (MTU5)

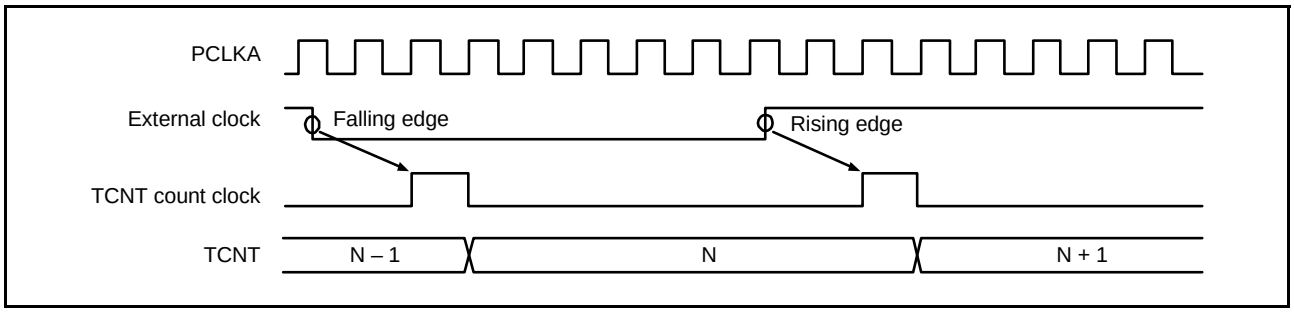


Figure 24.113 Count Timing in External Clock Operation (MTU0 to MTU4 and MTU6 to MTU8)

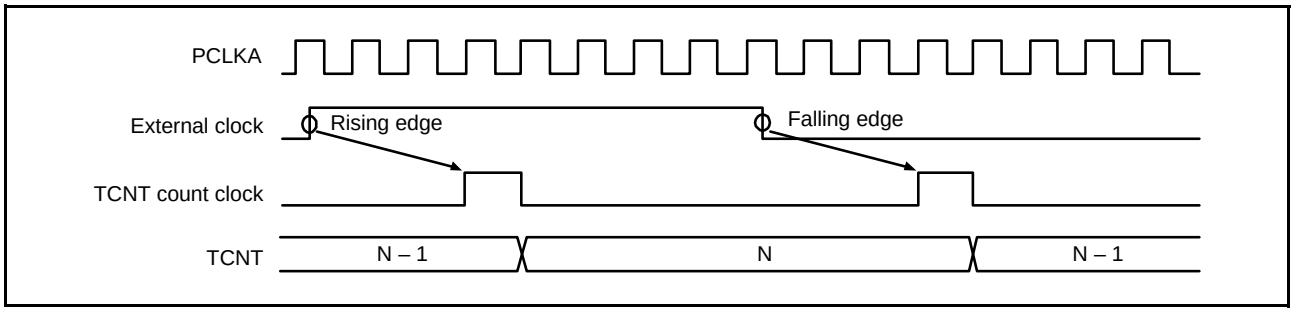


Figure 24.114 Count Timing in External Clock Operation (Phase Counting Mode)

(2) Output Compare Output Timing

A compare match signal is generated in the final state in which TCNT and TGR match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the value set in TIOR is output from MTIOCnm pin (n = 0 to 4, 6, 7, 8; m = A to D). After a match between TCNT and TGR, the compare match signal is not generated until the TCNT count clock is generated.

Figure 24.115 shows the output compare output timing (normal mode or PWM mode) and Figure 24.116 shows the output compare output timing (complementary PWM mode or reset-synchronized PWM mode).

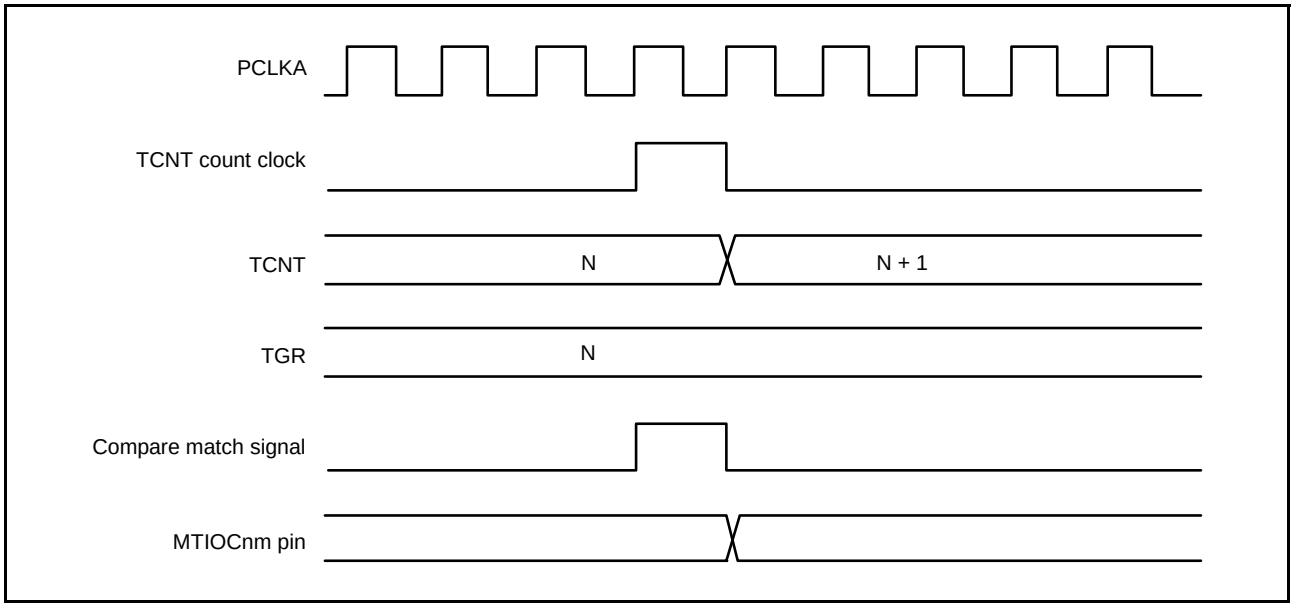


Figure 24.115 Output Compare Output Timing (Normal Mode or PWM Mode) (n = 0 to 4, 6, 7, 8; m = A to D)

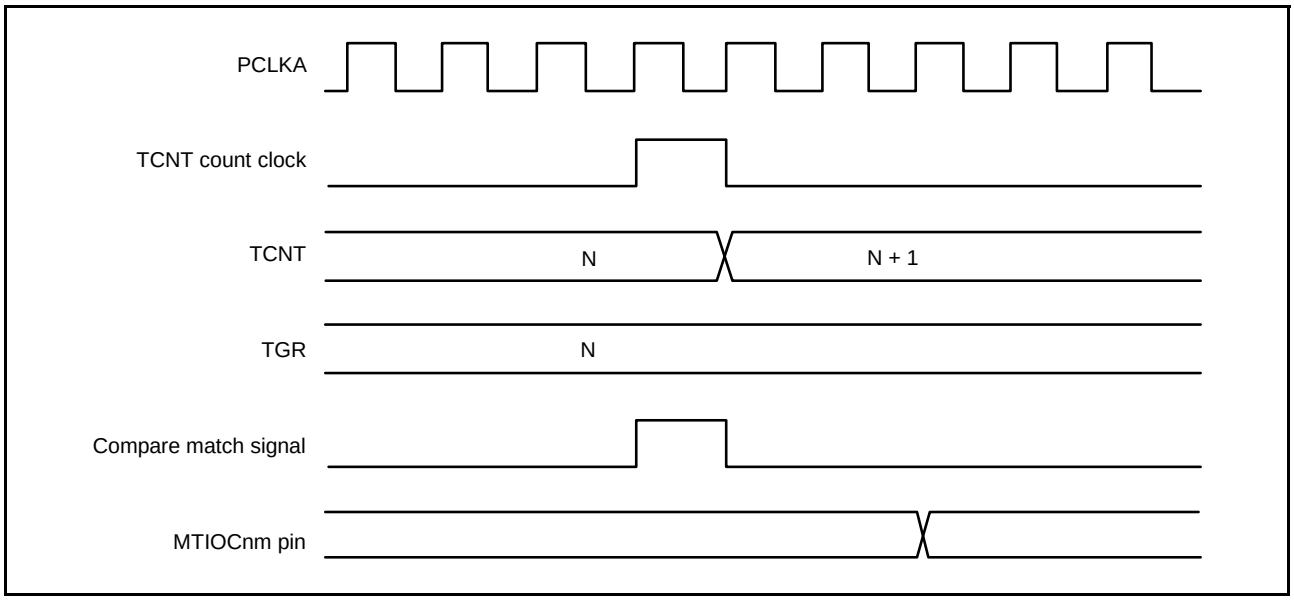


Figure 24.116 Output Compare Output Timing (Complementary PWM Mode or Reset-Synchronized PWM Mode) (n = 0 to 4, 6, 7, 8; m = A to D)

(3) Input Capture Signal Timing

Figure 24.117 shows the input capture signal timing.

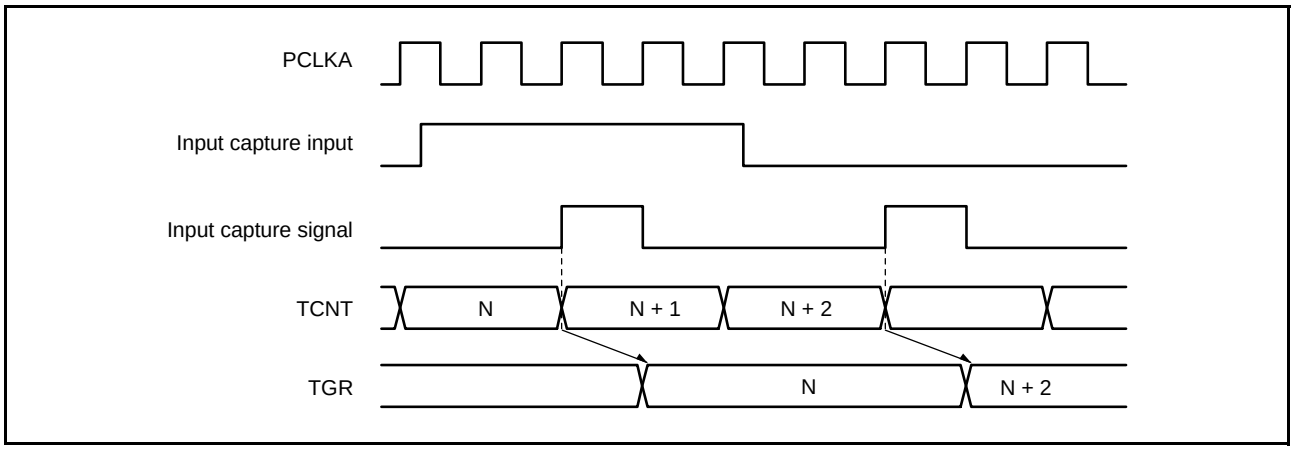


Figure 24.117 Input Capture Input Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

FFigure 24.118 and Figure 24.119 show the timing when counter clearing on compare match is specified, and Figure 24.120 shows the timing when counter clearing on input capture is specified.

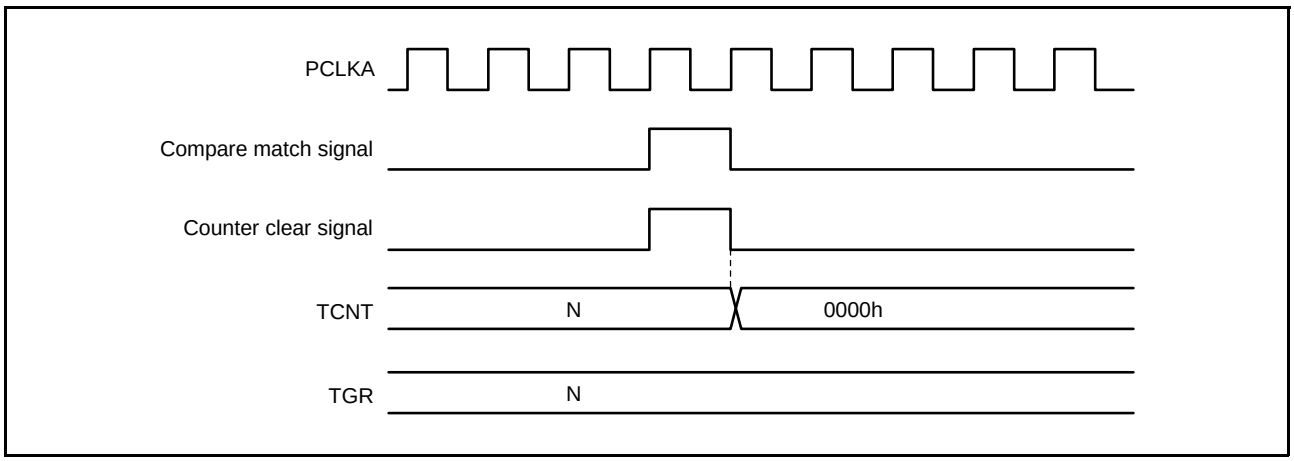


Figure 24.118 Counter Clear Timing (Compare Match) (MTU0 to MTU4 and MTU6 to MTU8)

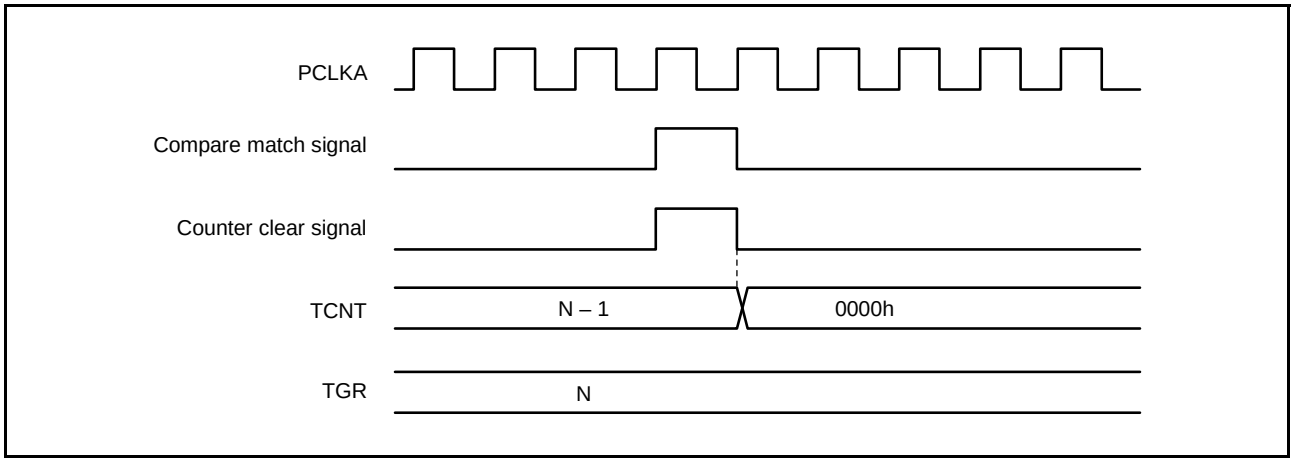


Figure 24.119 Counter Clear Timing (Compare Match) (MTU5)

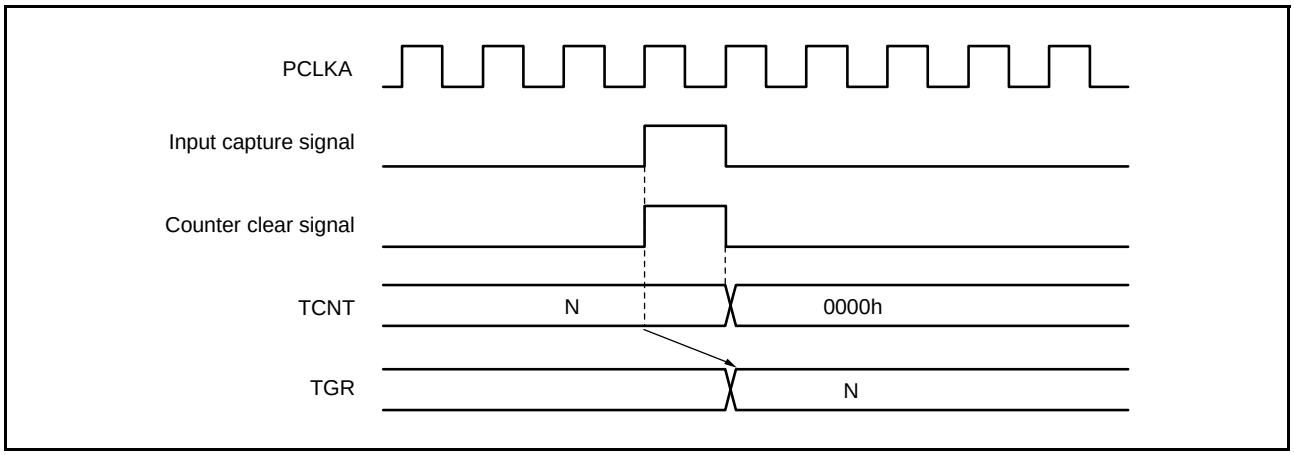


Figure 24.120 Counter Clear Timing (Input Capture) (MTU0 to MTU8)

(5) Buffer Operation Timing

Figure 24.121 to Figure 24.123 show the timing in buffer operation.

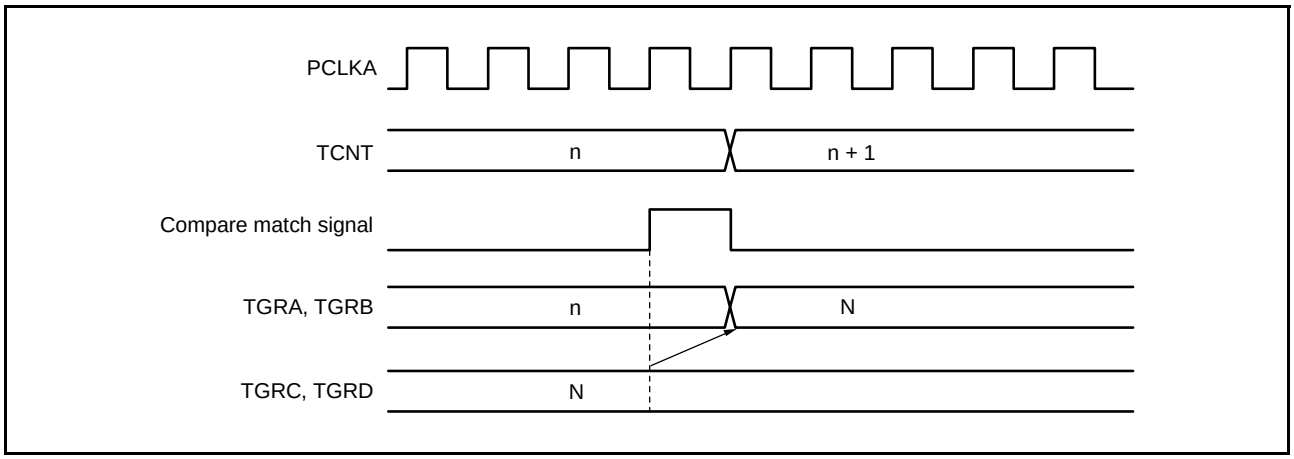


Figure 24.121 Buffer Operation Timing (Compare Match)

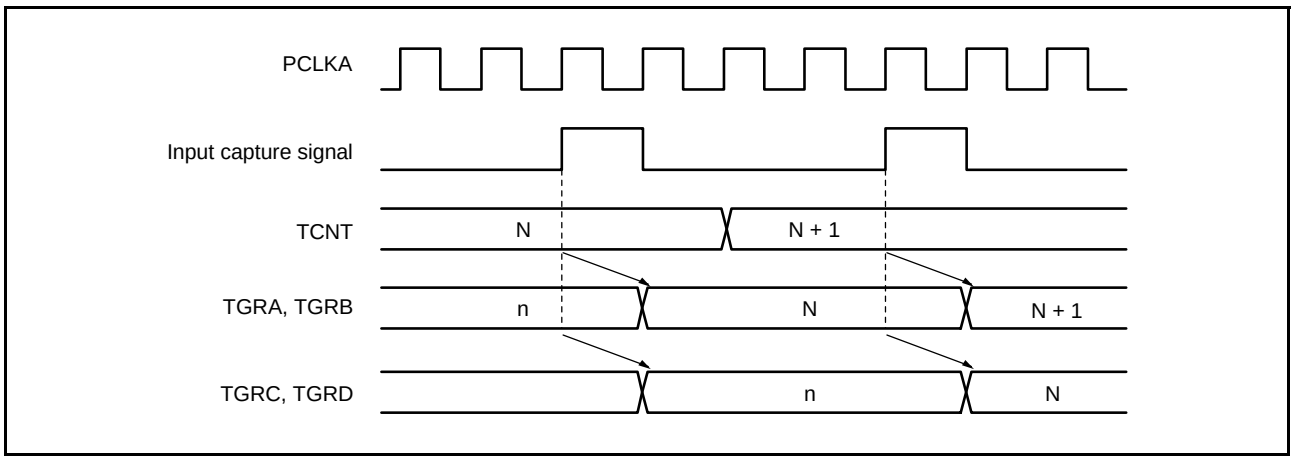


Figure 24.122 Buffer Operation Timing (Input Capture)

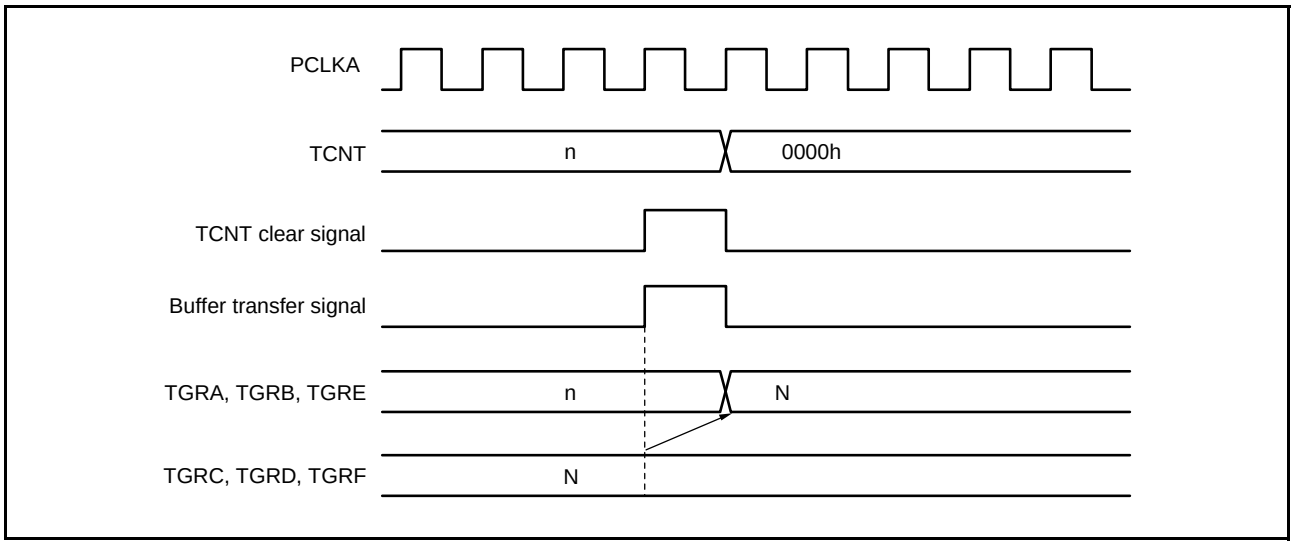


Figure 24.123 Buffer Operation Timing (When TCNT Cleared)

(6) Buffer Transfer Timing (Complementary PWM Mode)

Figure 24.124 to Figure 24.126 show the buffer transfer timing in complementary PWM mode.

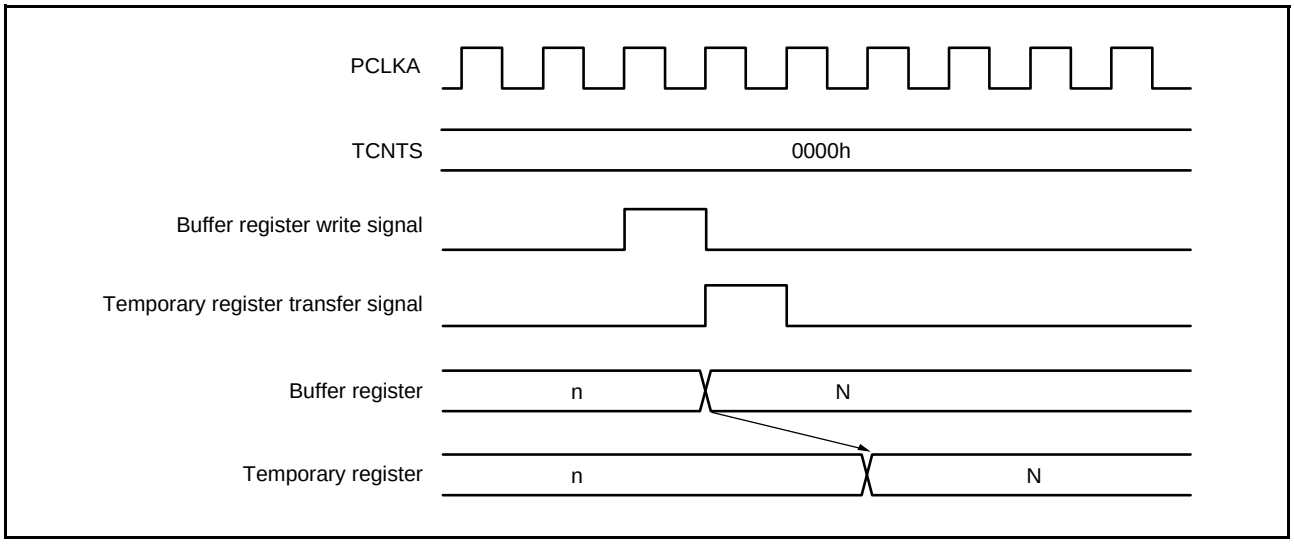


Figure 24.124 Transfer Timing from Buffer Register to Temporary Register (TCNTSA Stopped)

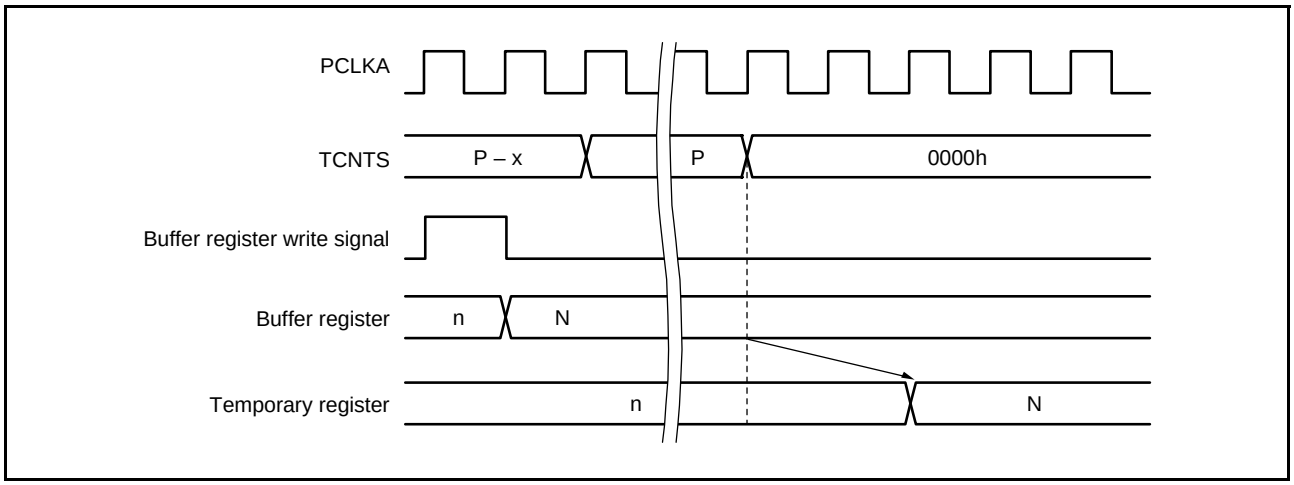


Figure 24.125 Transfer Timing from Buffer Register to Temporary Register (TCNTSA Operating)

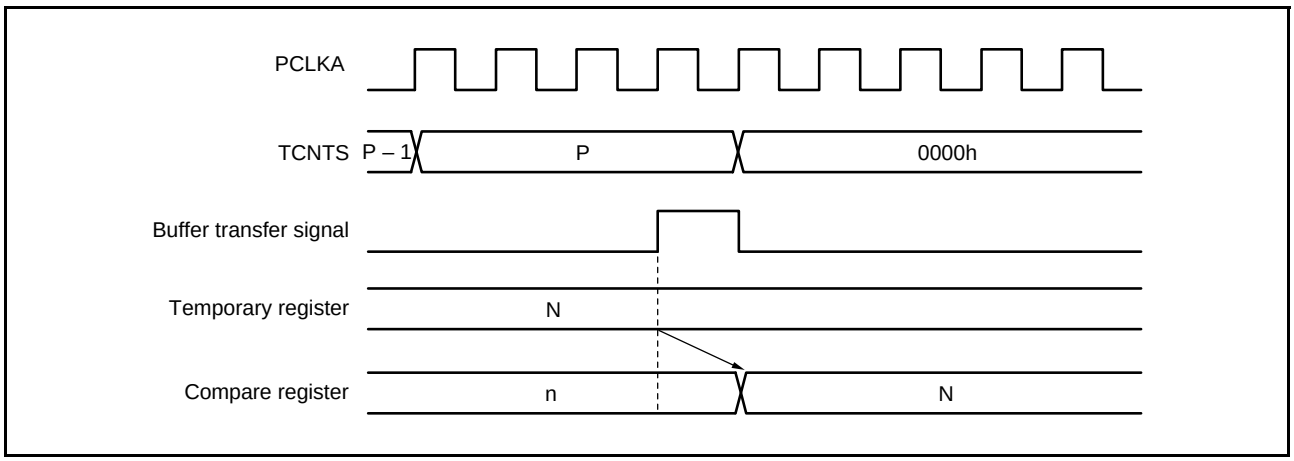


Figure 24.126 Transfer Timing from Temporary Register to Compare Register

24.5.2 Interrupt Signal Timing

(1) TGI Interrupt Timing by Compare Match

Figure 24.127 and Figure 24.128 show the TGI interrupt request signal timing when a compare match occurs.

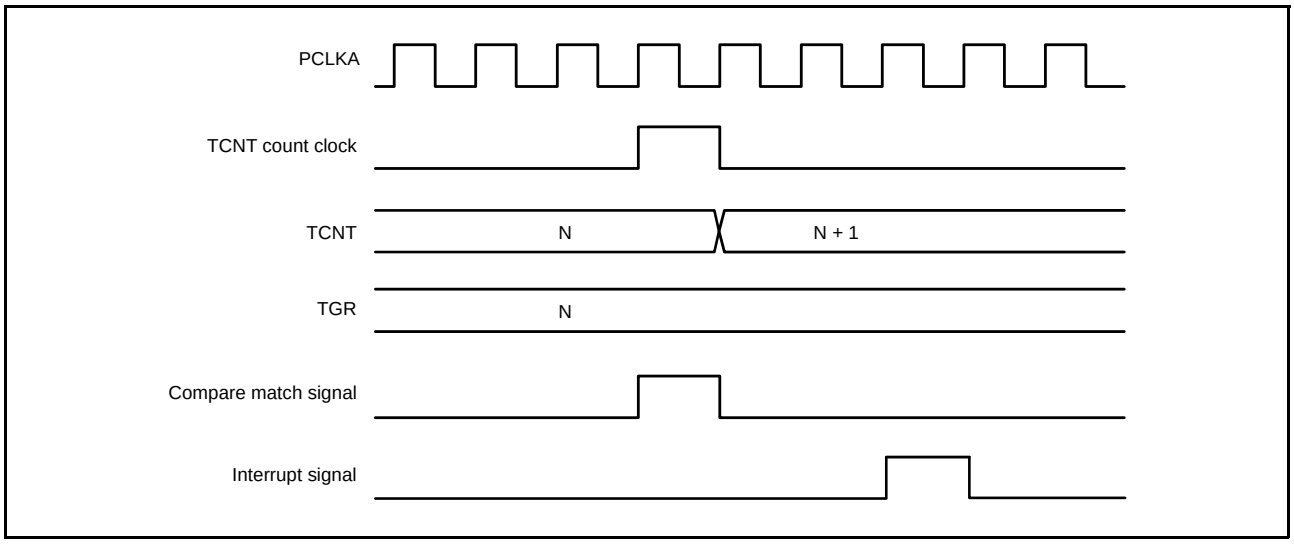


Figure 24.127 TGI Interrupt Timing (Compare Match) (MTU0 to MTU4 and MTU6 to MTU8)

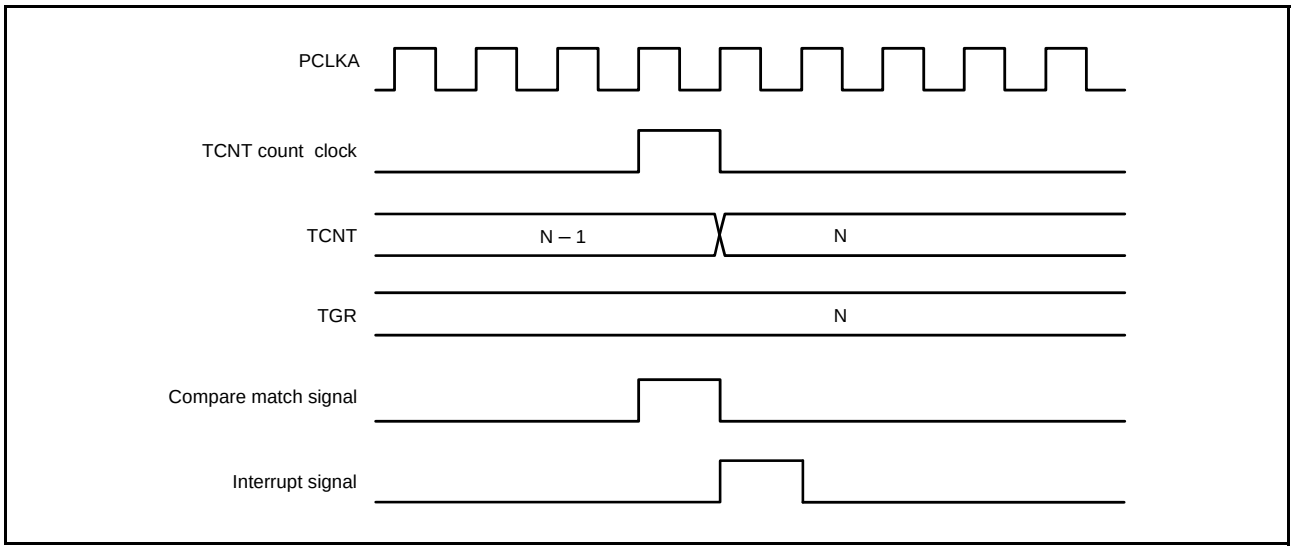


Figure 24.128 TGI Interrupt Timing (Compare Match) (MTU5)

(2) TGI Interrupt Timing by Input Capture

Figure 24.129 and Figure 24.130 show the TGI interrupt request signal timing when an input capture occurs.

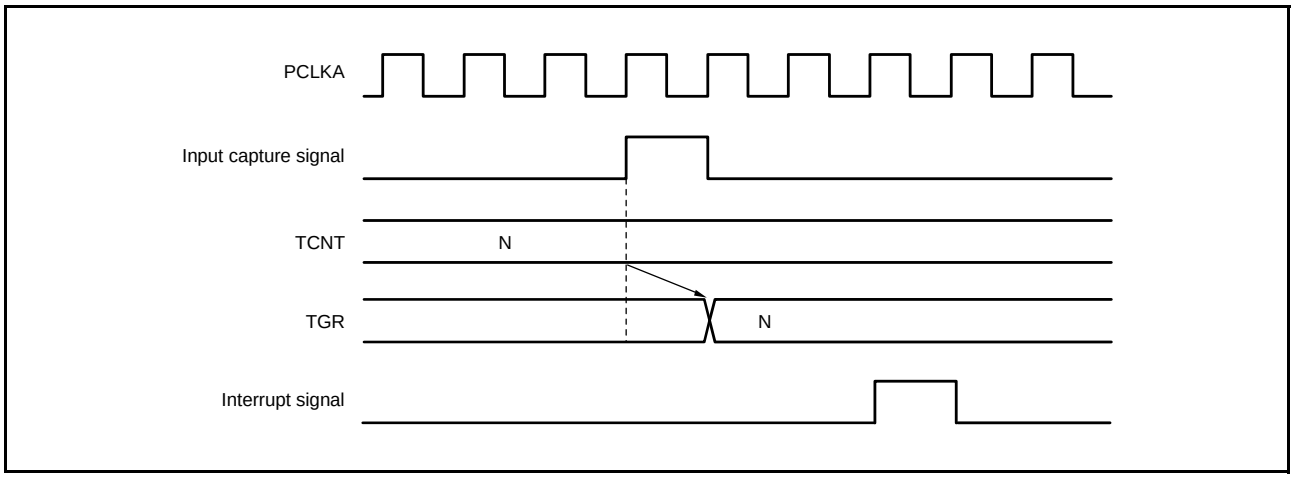


Figure 24.129 TGI Interrupt Timing (Input Capture) (MTU0 to MTU4 and MTU6 to MTU8)

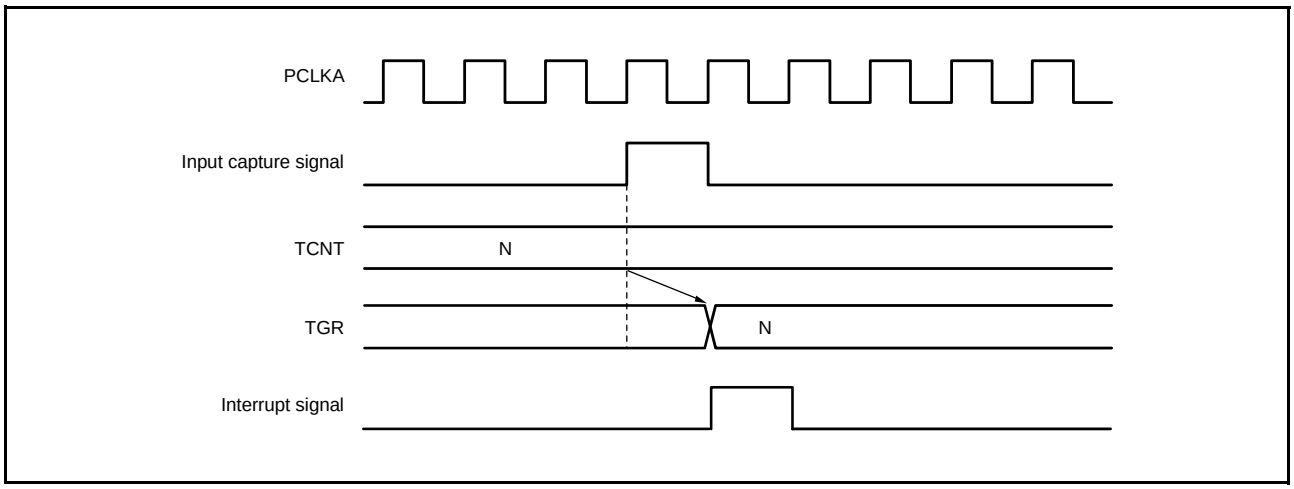


Figure 24.130 TGI Interrupt Timing (Input Capture) (MTU5)

(3) TCIV and TCIU Interrupt Timing

Figure 24.131 shows the TCIV interrupt request signal timing when an overflow is generated.
Figure 24.132 shows the TCIU interrupt request signal timing when an underflow is generated.

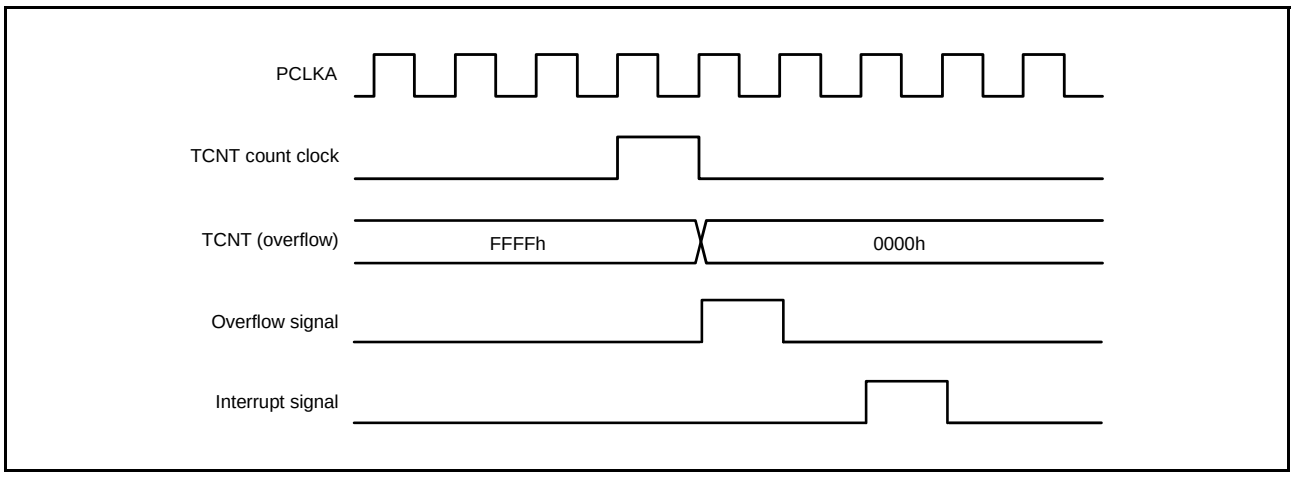


Figure 24.131 TCIV Interrupt Timing

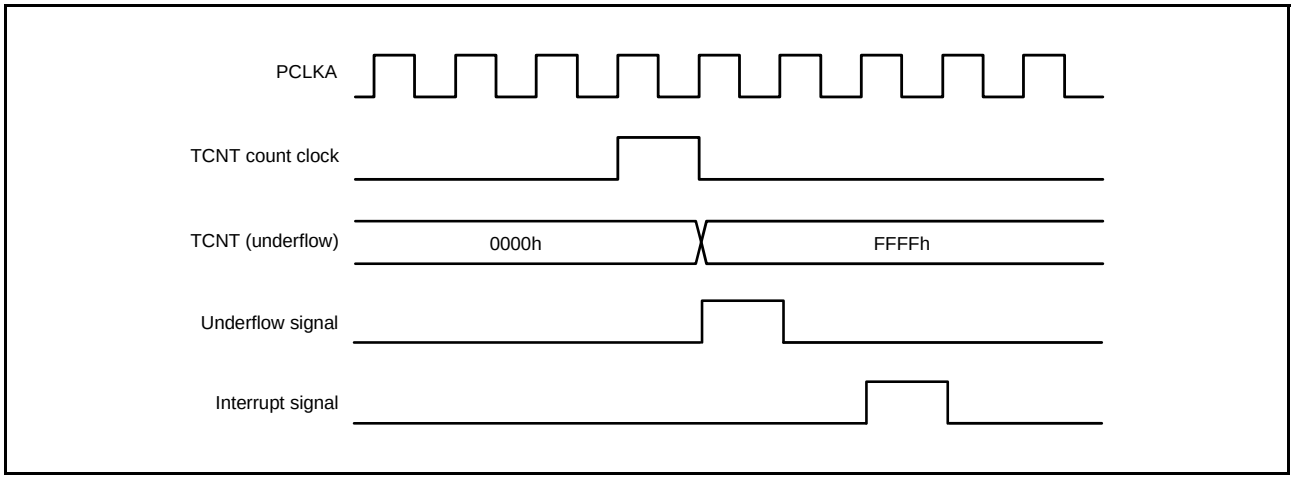


Figure 24.132 TCIU Interrupt Timing

24.6 Usage Notes

24.6.1 Module Stop Function Setting

MTU operation can be disabled or enabled using the module stop control register. MTU operation is stopped with the initial setting. Register access is enabled by releasing the module clock stop state. For details, refer to section 11, Low Power Consumption.

24.6.2 Count Clock Restrictions

The count clock source pulse width must be at least 1.5 PCLKA cycles for single-edge detection, and at least 2.5 PCLKA cycles for both-edge detection. The MTU will not operate properly at narrower pulse widths.
In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 PCLKA cycles, and the pulse width must be at least 2.5 PCLKA cycles. Figure 24.133 shows the input clock conditions in phase counting mode.

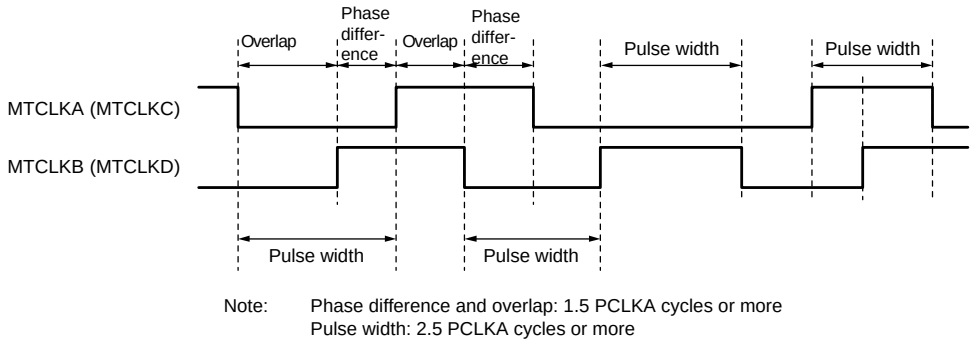


Figure 24.133 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

24.6.3 Note on Period Setting

When counter clearing on compare match is set, TCNT is cleared in the final state in which it matches the TGR value (the point at which TCNT updates the matched count value). Consequently, the actual counter frequency is given by the following formula:

- MTU0 to MTU4 and MTU6 to MTU8

$$f = \frac{CNTCLK}{N + 1}$$

- MTU5

$$f = \frac{CNTCLK}{N}$$

f: Counter frequency
CNTCLK: The count clock frequency set by TCR.TPSC[2:0] and TCR2.TPSC2[2:0]
N: TGR setting

24.6.4 Contention between TCNT Write and Clear Operations

If the counter clear signal is generated in the TCNT write cycle, TCNT clearing takes precedence and TCNT write operation is not performed.
Figure 24.134 shows the timing in this case.

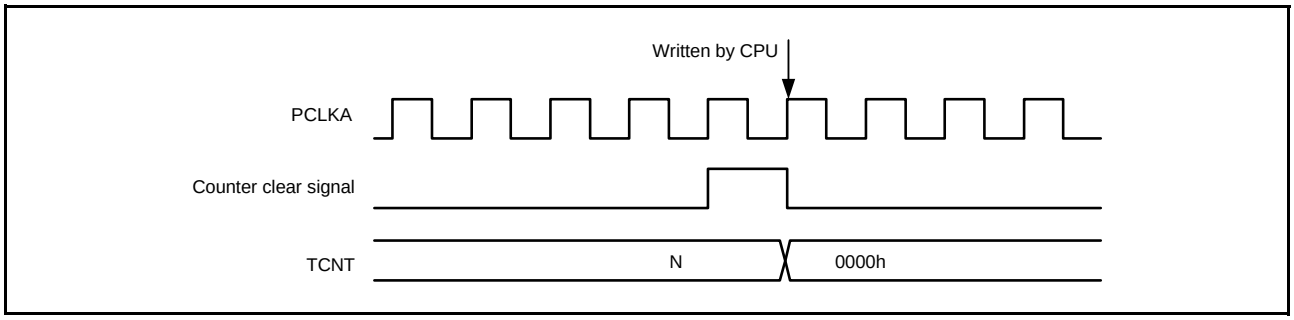


Figure 24.134 Contention between TCNT Write and Clear Operations

24.6.5 Contention between TCNT Write and Increment Operations

If incrementing occurs in a TCNT write cycle, TCNT write operation takes precedence and TCNT is not incremented.
Figure 24.135 shows the timing in this case.

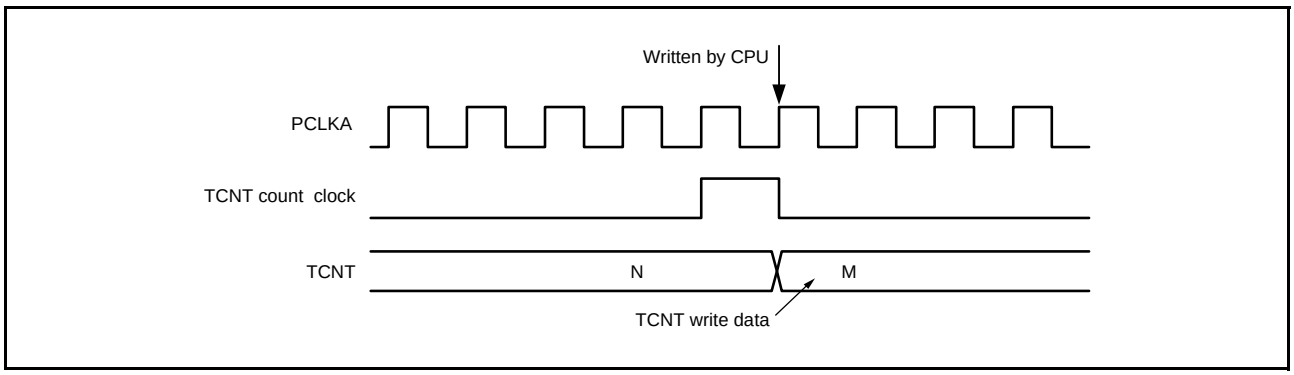


Figure 24.135 Contention between TCNT Write and Increment Operations

24.6.6 Contention between TGR Write Operation and Compare Match

If a compare match occurs in a TGR write cycle, TGR write operation is executed and the compare match signal is also generated.
Figure 24.136 shows the timing in this case.

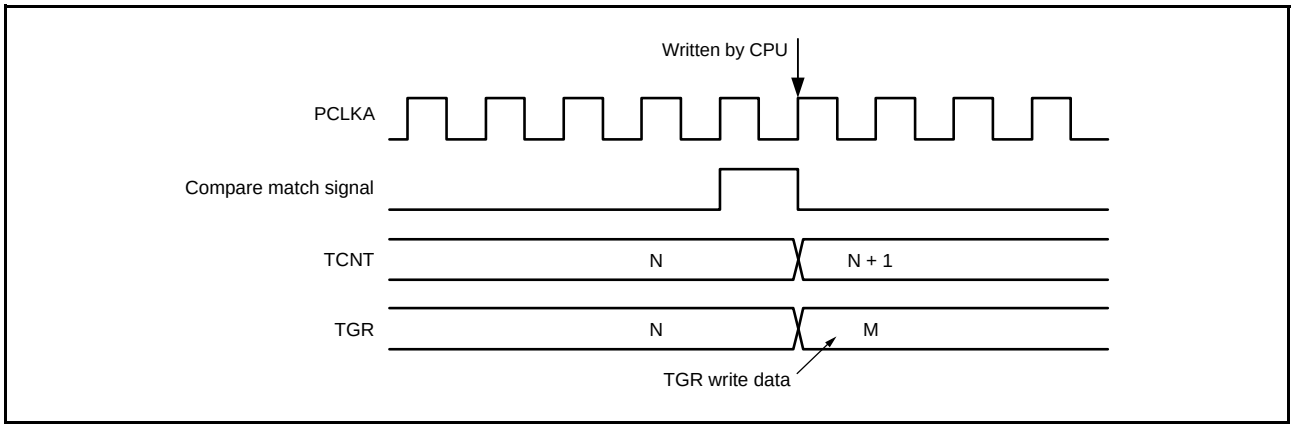


Figure 24.136 Contention between TGR Write Operation and Compare Match

24.6.7 Contention between Buffer Register Write Operation and Compare Match

If a compare match occurs in the T2 state in a TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.
Figure 24.137 shows the timing in this case.

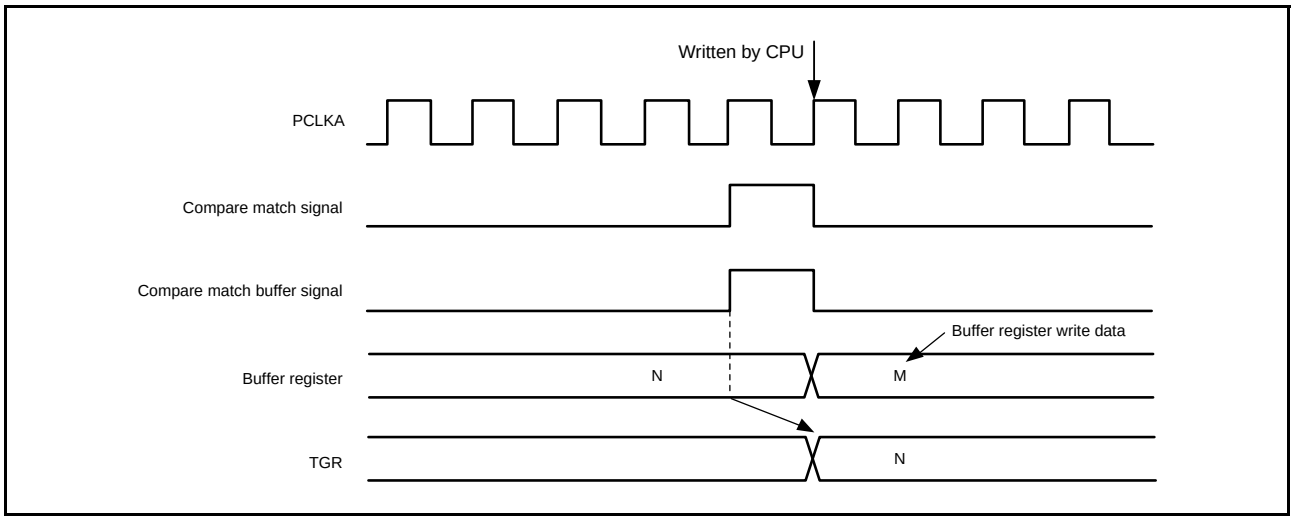


Figure 24.137 Contention between Buffer Register Write Operation and Compare Match

24.6.8 Contention between Buffer Register Write and TCNT Clear Operations

When the buffer transfer timing is set at the TCNT clear timing by the timer buffer transfer mode register (TBTM), if TCNT clearing occurs in the TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.
Figure 24.138 shows the timing in this case.

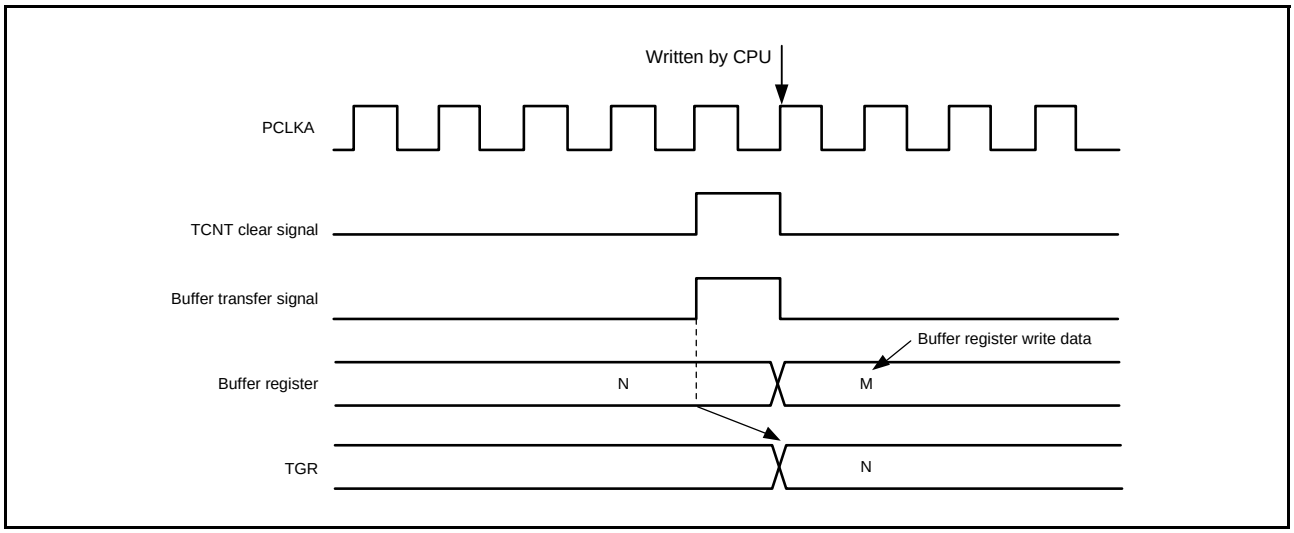


Figure 24.138 Contention between Buffer Register Write and TCNT Clear Operations

24.6.9 Contention between TGR Read Operation and Input Capture

If an input capture signal is generated in a TGR read cycle, the data before input capture transfer is read.
Figure 24.139 shows the timing in this case.

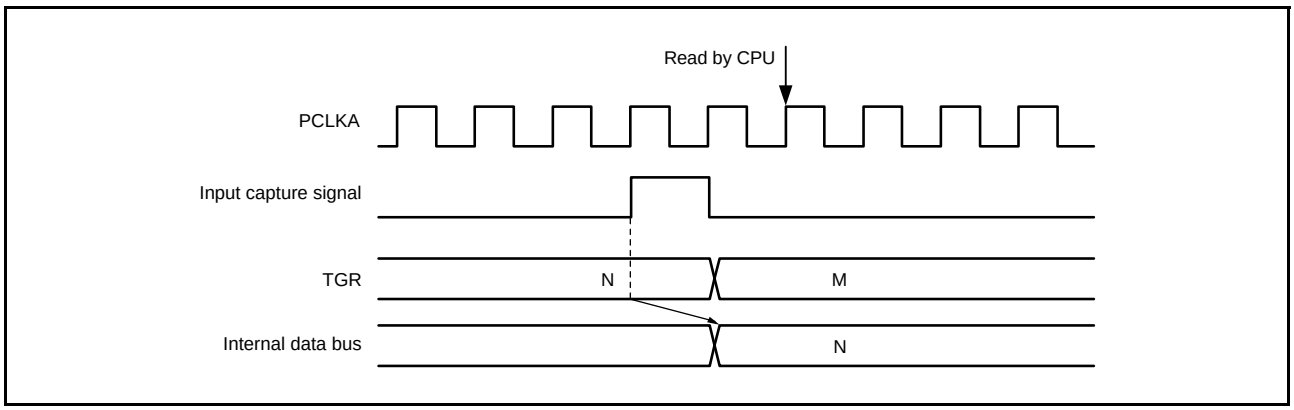


Figure 24.139 Contention between TGR Read Operation and Input Capture (MTU0 to MTU8)

24.6.10 Contention between TGR Write Operation and Input Capture

If an input capture signal is generated in the TGR write cycle, the input capture operation takes precedence and the TGR write operation is not performed in MTU0 to MTU4 and MTU6 to MTU8. In MTU5, the TGR write operation is performed and the input capture signal is generated.
Figure 24.140 and Figure 24.141 show the timing in this case.

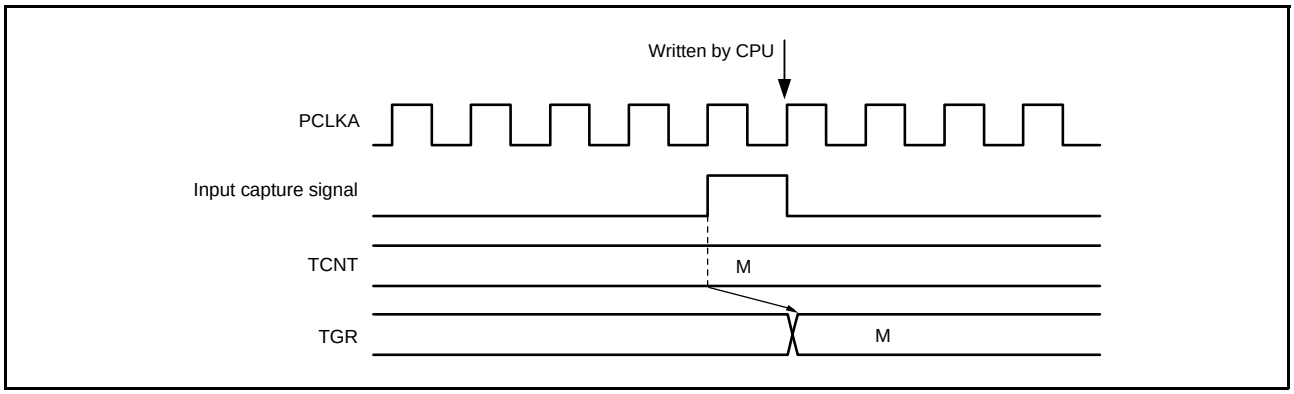


Figure 24.140 Contention between TGR Write Operation and Input Capture (MTU0 to MTU4 and MTU6 to MTU8)

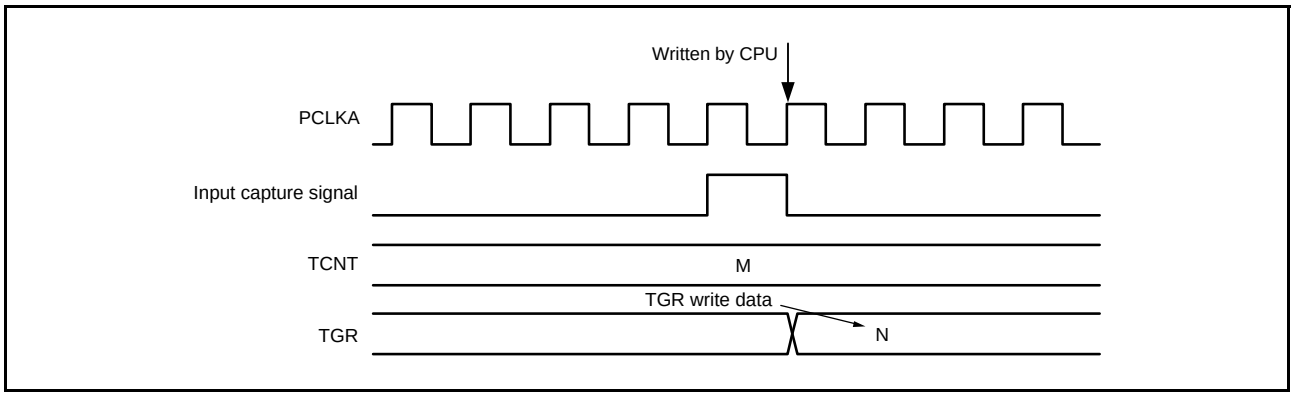


Figure 24.141 Contention between TGR Write Operation and Input Capture (MTU5)

24.6.11 Contention between Buffer Register Write Operation and Input Capture

If an input capture signal is generated in the buffer register write cycle, the buffer operation takes precedence and the buffer register write operation is not performed.
Figure 24.142 shows the timing in this case.

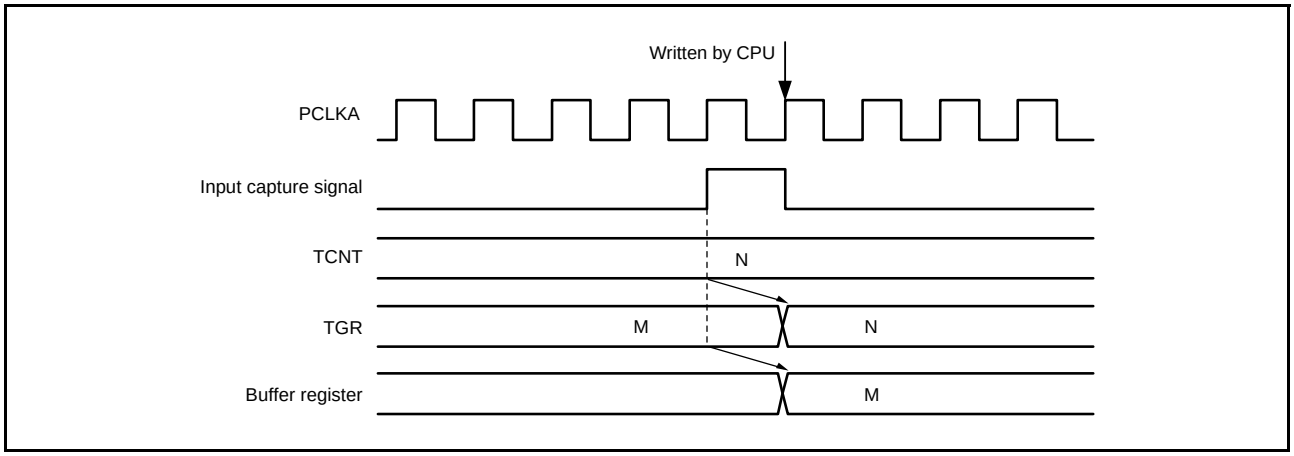


Figure 24.142 Contention between Buffer Register Write Operation and Input Capture

24.6.12 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation

With timer counters MTU1.TCNT and MTU2.TCNT in a cascade, when a contention occurs between MTU1.TCNT counting (an MTU2.TCNT overflow/underflow) and the MTU2.TCNT write operation, the MTU2.TCNT write operation is performed and the MTU1.TCNT count signal is disabled. In this case, if MTU1.TGRA works as a compare match register and there is a match between the MTU1.TGRA and MTU1.TCNT values, a compare match signal is issued.

Furthermore, when the MTU1.TCNT count clock is selected as the input capture source of MTU0, MTU0.TGRA to MTU0.TGRD work in input capture mode. In addition, when the MTU0.TGRC compare match/input capture is selected as the input capture source of MTU1.TGRB, MTU1.TGRB works in input capture mode.

Figure 24.143 shows the timing in this case.

When setting the TCNT clearing function in cascaded operation, be sure to synchronize MTU1 and MTU2.

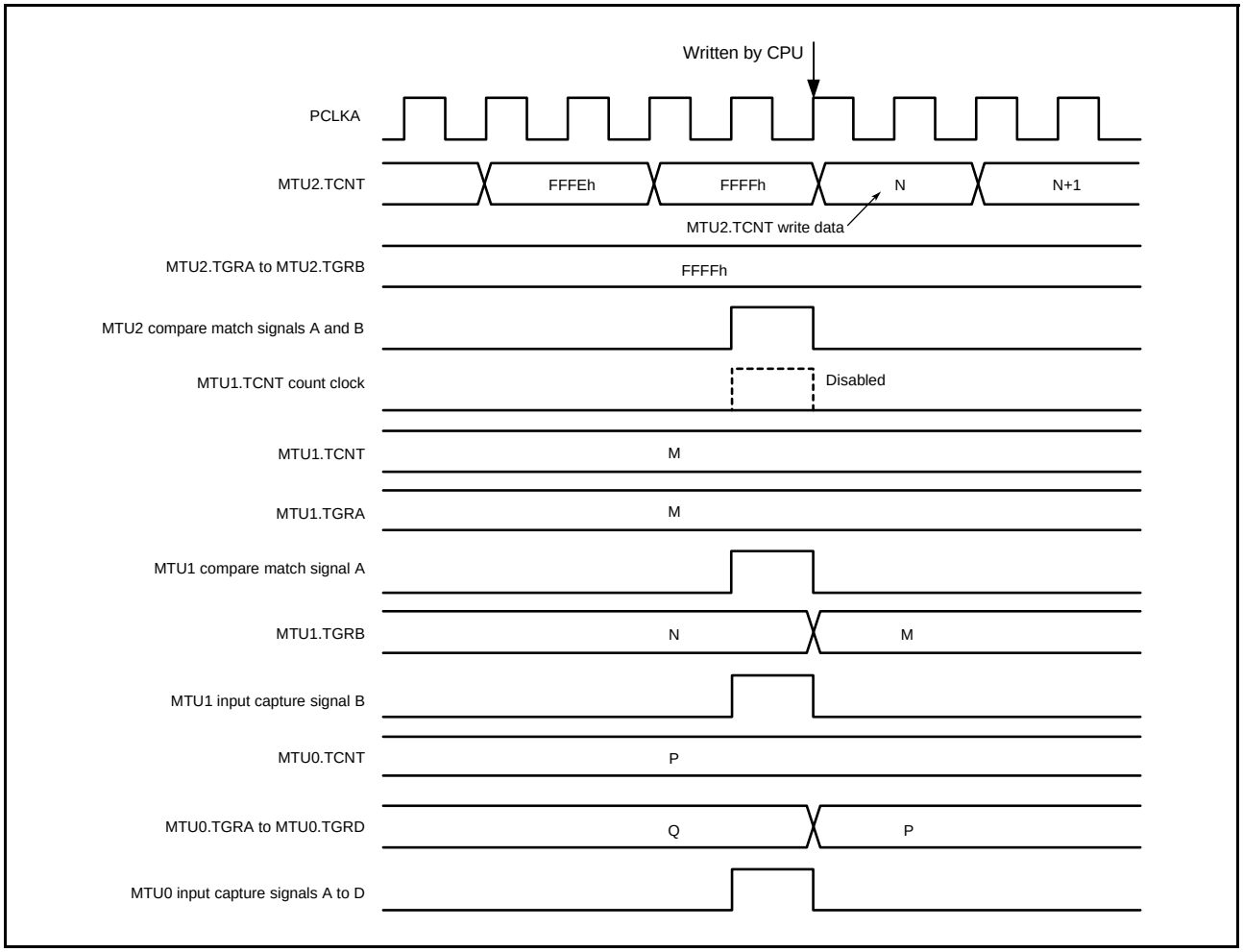


Figure 24.143 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation

24.6.13 Counter Value When Count Operation is Stopped in Complementary PWM Mode

When counting operation in MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) is stopped in complementary PWM mode, the MTU3.TCNT (MTU6.TCNT) value is set to the timer dead time register (TDDRA (TDDRB)) value and MTU4.TCNT (MTU7.TCNT) is set to 0000h.
When operation is restarted in complementary PWM mode, counting begins automatically from the initial setting state. Figure 24.144 shows this operation.
When counting begins in another operating mode, be sure to make initial settings in MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT).

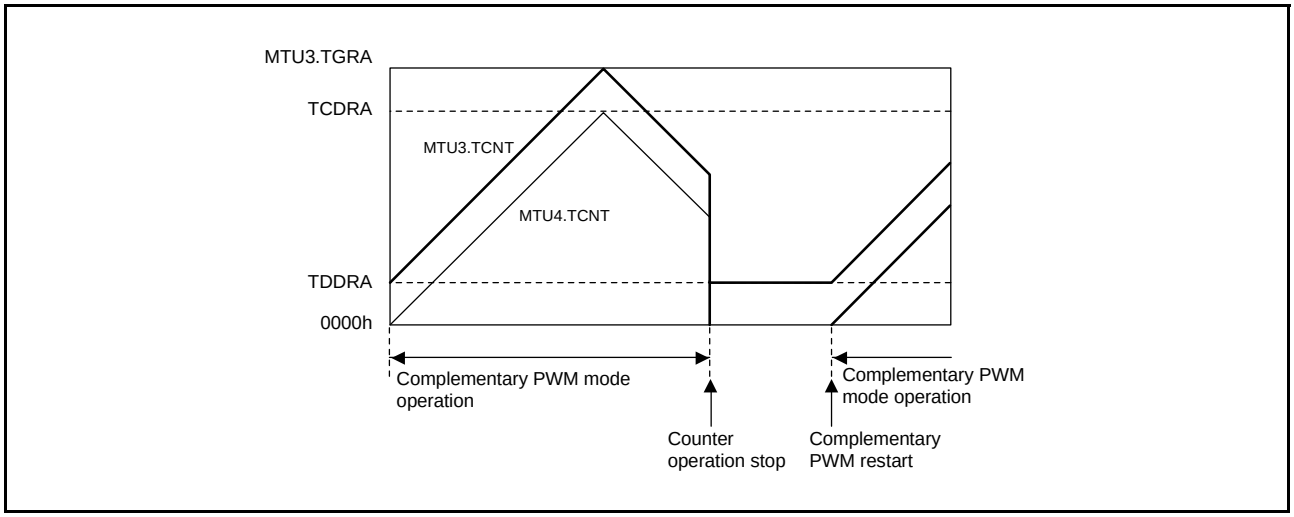


Figure 24.144 Counter Value When Stopped in Complementary PWM Mode

24.6.14 Buffer Operation Setting in Complementary PWM Mode

When modifying the PWM period set register (MTU3.TGRA or MTU6.TGRA), timer period data register (TCDRA or TCDRB), and duty set registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB (MTU6.TGRB, MTU7.TGRA, and MTU7.TGRB)) in complementary PWM mode, be sure to use buffer operation. In addition, set the BFA and BFB bits in MTU4.TMDR1 (MTU7.TMDR1) to 0. The MTIOC4C (MTIOC7C) pin cannot output waveforms if the BFA bit in MTU4.TMDR1 (MTU7.TMDR1) is set to 1. Likewise, the MTIOC4D (MTIOC7D) pin cannot output waveforms if the BFB bit in MTU4.TMDR1 (MTU7.TMDR1) is set to 1.
In complementary PWM mode, buffer operation in MTU3 and MTU4 (or MTU6 and MTU7) depends on the settings in bits BFA and BFB of MTU3.TMDR1 (MTU6.TMDR1). When the BFA bit in MTU3.TMDR1 (MTU6.TMDR1) is set to 1, MTU3.TGRC (MTU6.TGRC) functions as a buffer register for MTU3.TGRA (MTU6.TGRA). At the same time, MTU4.TGRC (MTU7.TGRC) functions as a buffer register for MTU4.TGRA (MTU7.TGRA), and TCBRA (TCBRB) functions as a buffer register for TCDRA (TCDRB).

24.6.15 Buffer Operation and Compare Match in Reset-Synchronized PWM Mode

When setting buffer operation in reset-synchronized PWM mode, set the BFA and BFB bits in MTU4.TMDR1 (MTU7.TMDR1) to 0. The MTIOC4C (MTIOC7C) pin cannot output waveforms if the BFA bit in MTU4.TMDR1 (MTU7.TMDR1) is set to 1. Likewise, the MTIOC4D (MTIOC7D) pin cannot output waveforms if the BFB bit in MTU4.TMDR1 (MTU7.TMDR1) is set to 1.

In reset-synchronized PWM mode, buffer operation in MTU3 and MTU4 (or MTU 6 and MTU7) depends on the settings in the BFA and BFB bits of MTU3.TMDR1 (MTU6.TMDR1). For example, if the BFA bit in MTU3.TMDR1 (MTU6.TMDR1) is set to 1, MTU3.TGRC (MTU6.TGRC) functions as a buffer register for MTU3.TGRA (MTU6.TGRA). At the same time, MTU4.TGRC (MTU7.TGRC) functions as a buffer register for MTU4.TGRA (MTU7.TGRA).

While the MTU3.TGRC (MTU6.TGRC) and MTU3.TGRD (MTU6.TGRD) are operating as buffer registers, a TGImn interrupt (m = C, D; n = 3, 4 or 6, 7) is not generated.

Figure 24.145 shows an example of MTU3.TGR (MTU6.TGR), MTU4.TGR (MTU7.TGR), MTIOC3 (MTIOC6), and MTIOC4 (MTIOC7) operation with the BFA and BFB bits in MTU3.TMDR1 (MTU6.TMDR1) set to 1 and the BFA and BFB bits in MTU4.TMDR1 (MTU7.TMDR1) set to 0.

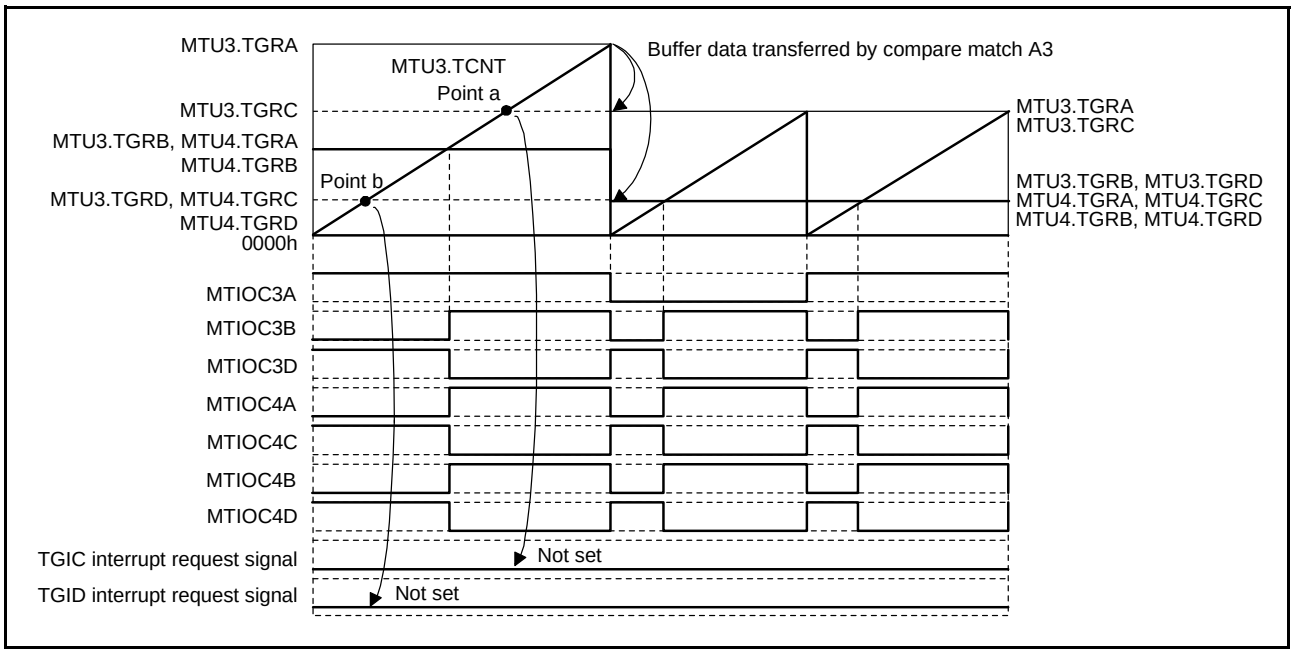


Figure 24.145 Buffer Operation and Compare Match in Reset-Synchronized PWM Mode

24.6.16 Overflow in Reset-Synchronized PWM Mode

After reset-synchronized PWM mode is selected, MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) start counting when the CST3 (CST6) bit of TSTRA (TSTRB) is set to 1. In this state, the MTU4.TCNT (MTU7.TCNT) count clock source and count edge are determined by the MTU3.TCR (MTU6.TCR) setting.

In reset-synchronized PWM mode, with period register MTU3.TGRA (MTU6.TGRA) set to FFFFh and the MTU3.TGRA (MTU6.TGRA) compare match selected as the counter clearing source, MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) count up to FFFFh, then a compare match occurs with MTU3.TGRA (MTU6.TGRA), and MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) are both cleared. In this case, a TCIVn interrupt (n = 3, 4 or 6, 7) is not generated.

Figure 24.146 shows an example of operation in reset-synchronized PWM mode with period register MTU3.TGRA (MTU6.TGRA) set to FFFFh and the MTU3.TGRA (MTU6.TGRA) compare match specified for the counter clearing source.

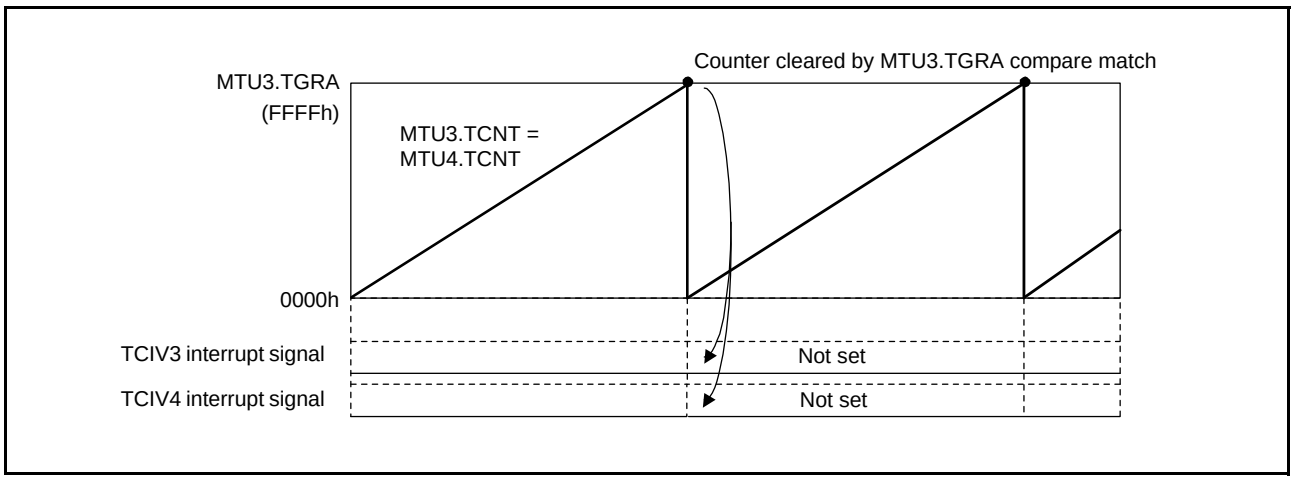


Figure 24.146 Overflow in Reset-Synchronized PWM Mode

24.6.17 Contention between Overflow/Underflow and Counter Clearing

If an overflow/underflow and counter clearing occur simultaneously, a TCIVn interrupt (n = 0 to 4, 6 to 8) nor a TCIUn interrupt (n = 1, 2) is not generated and TCNT clearing takes precedence.
Figure 24.147 shows the operation timing when a TGR compare match is specified as the clearing source and TGR is set to FFFFh.

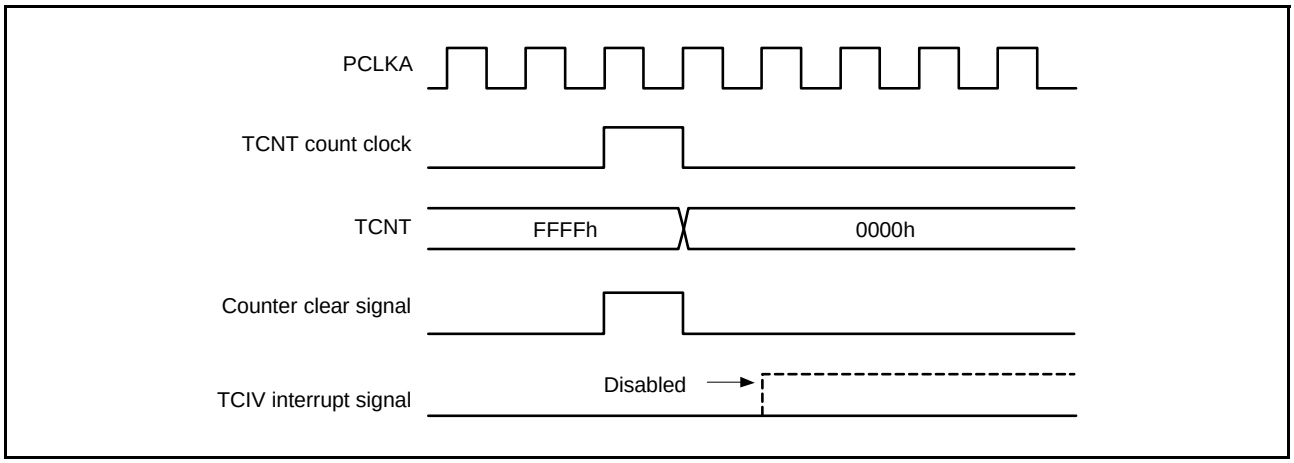


Figure 24.147 Contention between Overflow and Counter Clearing

24.6.18 Contention between TCNT Write Operation and Overflow/Underflow

If TCNT counts up or down in a TCNT write cycle and an overflow or an underflow occurs, the TCNT write operation takes precedence. A TCIVn interrupt (n = 0 to 4, 6 to 8) nor a TCIUn interrupt (n = 1, 2) is not generated.
Figure 24.148 shows the operation timing when there is contention between TCNT write operation and overflow.

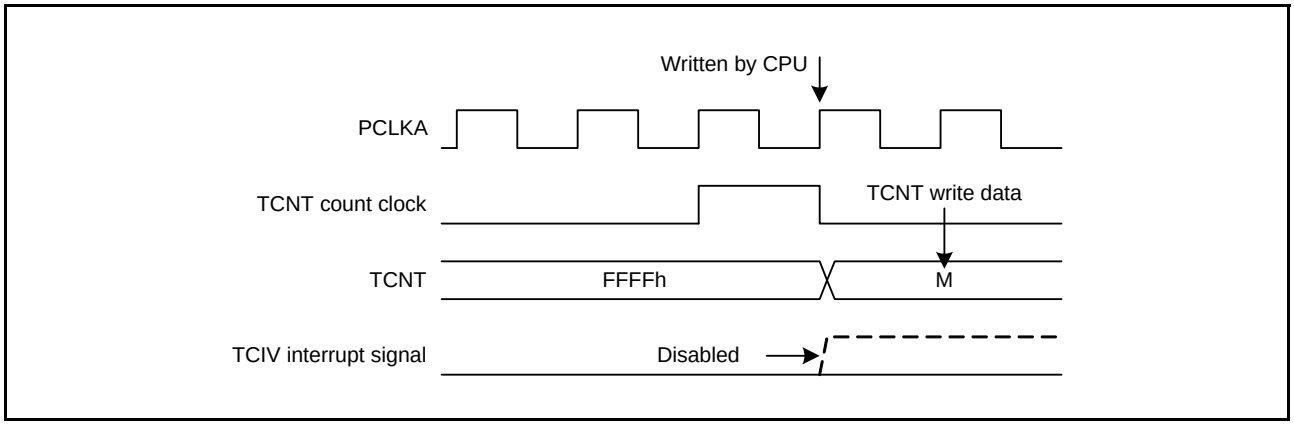


Figure 24.148 Contention between TCNT Write Operation and Overflow

24.6.19 Note on Transition from Normal Mode or PWM Mode 1 to Reset-Synchronized PWM Mode

When making a transition from normal mode or PWM mode 1 to reset-synchronized PWM mode in MTU3 and MTU4 (or MTU6 and MTU7), if the counter is stopped while the output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D, MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, and MTIOC7D) are held at a high level and then operation is started after a transition to reset-synchronized PWM mode, the initial pin output will not be correct.

When making a transition from normal mode to reset-synchronized PWM mode, write 11h to MTU3.TIORH, MTU3.TIORL, MTU4.TIORH, and MTU4.TIORL (MTU6.TIORH, MTU6.TIORL, MTU7.TIORH, and MTU7.TIORL) to initialize the output pin state to a low level, then set the registers to the initial value (00h) before making the mode transition.

When making a transition from PWM mode 1 to reset-synchronized PWM mode, switch to normal mode, initialize the output pin state to a low level, and then set the registers to the initial value (00h) before making the transition to reset-synchronized PWM mode.

24.6.20 Output Level in Complementary PWM Mode and Reset-Synchronized PWM Mode

When MTU3 and MTU4 (or MTU6 and MTU7) are in complementary PWM mode or reset-synchronized PWM mode, the PWM waveform output level is determined by the OLSP and OLSN bits in the timer output control register (TOCR1A or TOCR1B). In complementary PWM mode or reset-synchronized PWM mode, TIOR should be set to 00h. The output level in negative phase when the TDERA.TDER (TDERB.TDER) bit is set to 0 in complementary PWM mode (the dead time is not generated) does not depend on the setting of the TOCR1A.OLSN (TOCR1B.OLSN) bit. It is equivalent to the inverted level of positive phase output based on the setting of the TOCR1A.OLSP (TOCR1B.OLSP) bit.

24.6.21 Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection

When timer counters 1 and 2 (MTU1.TCNT and MTU2.TCNT) operate as a 32-bit counter in cascade connection, the cascaded counter value cannot be captured successfully in some cases even if input-capture input is simultaneously done to MTIOC1A and MTIOC2A or to MTIOC1B and MTIOC2B. This is because the input timing of MTIOC1A and MTIOC2A or to MTIOC1B and MTIOC2B may not be the same when external input-capture signals input into MTU1.TCNT and MTU2.TCNT are taken in synchronization with the internal clock.

For example, MTU1.TCNT (the counter for upper 16 bits) does not capture the count-up value by an overflow from MTU2.TCNT (the counter for lower 16 bits) but captures the count value before the up-counting. In this case, the values of MTU1.TCNT = FFF1h and MTU2.TCNT = 0000h should be transferred to MTU1.TGRA and MTU2.TGRA or to MTU1.TGRB and MTU2.TGRB, but the values of MTU1.TCNT = FFF0h and MTU2.TCNT = 0000h are erroneously transferred.

The MTU has a function that allows simultaneous capture of MTU1.TCNT and MTU2.TCNT with a single input capture input. This function can be used to read the 32-bit counter such that MTU1.TCNT and MTU2.TCNT are captured at the same time. For details, refer to section 24.2.11, Timer Input Capture Control Register (TICCR).

24.6.22 Interrupt Skipping Function 2

When interrupt skipping function 2 is in use and the difference between the values in MTU4.TADCORA and MTU4.TADCORB is small, correct counting of the number skipped may not be possible, in which case requests for A/D conversion will not be generated with the expected timing. The conditions listed below thus apply to these settings.
For MTU6 and MTU7, the same conditions apply to the settings of MTU7.TADCORA and MTU7.TADCORB.

(1) When the number skipped is zero for skipping function 2

- The difference between the values in MTU4.TADCORA and MTU4.TADCORB must be at least four.
- The interval of comparison for MTU4.TADCORA must be at least four cycles of PCLKA (the updated value of MTU4.TADCORA is set to the previous value plus or minus at least four).
- The interval of comparison for MTU4.TADCORB must be at least four cycles of PCLKA (the updated value of MTU4.TADCORB is set to the previous value plus or minus at least four).

(2) When the number skipped is one or more for skipping function 2

- The difference between the values in MTU4.TADCORA and MTU4.TADCORB must be at least two.
- The interval of comparison for MTU4.TADCORB must be at least two cycles of PCLKA (the updated value of MTU4.TADCORB is set to the previous value plus or minus at least two).

24.6.23 Notes When Complementary PWM Mode Output Protection Function is Not Used

The complementary PWM mode output protection function is initially enabled. For details, refer to section 25, Port Output Enable 3 (POE3a).

24.6.24 Notes Regarding Timer Counter (MTU5.TCNT) and Timer General Register (MTU5.TGR)

Do not set an MTU5.TGRm (m = U, V, W) bit to the value of the corresponding MTU5.TCNTm (m = U, V, W) plus one while counting by the MTU5.TCNTm (m = U, V, W) register is stopped. If an MTU5.TGRm (m = U, V, W) bit is set to the value of the corresponding MTU5.TCNTm (m = U, V, W) plus one while counting by the MTU5.TCNTm (m = U, V, W) is stopped, a compare-match will be generated even though counting is stopped.
In this case, if the compare match enable bit (MTU5.TIER.TGIE5m (m = U, V, W) bit is set to 1 (enabling interrupts), a compare-match interrupt will also be generated. If the value of the timer compare match clear register is 1 (enabled), the timer is automatically cleared to 0000h when the compare-match is generated, regardless of whether interrupts from the MTU5.TCNTm (m = U, V, W) are enabled or disabled.

24.6.25 Notes to Prevent Malfunctions in Synchronous Clearing for Complementary PWM Mode

If control of the output waveform is enabled (TWCRA.WRE bit = 1 or TWCRB.WRE bit = 1) at the time of synchronous counter clearing in complementary PWM mode, satisfaction of either condition 1 or 2 below has the following effects.

- Dead time on the PWM output pins is shortened (or disappears).
- The active level is output on the negative phase PWM output pins beyond the period for active-level output.

Condition 1: In portion (10) of the initial output inhibition period in Figure 24.149, synchronous clearing occurs within the dead-time period for PWM output.

Condition 2: In portions (10) and (11) of the initial output inhibition period in Figure 24.150, synchronous clearing occurs when any condition from among MTU3.TGRB (MTU6.TGRB) ≤ TDDRA (TDDRB), MTU4.TGRA (MTU7.TGRA) ≤ TDDRA (TDDRB), or MTU4.TGRB (MTU7.TGRB) ≤ TDDRA (TDDRB) is satisfied.

The following method avoids the above phenomena.
Ensure that synchronous clearing proceeds with the value of each comparison register (MTU3.TGRB (MTU6.TGRB), MTU4.TGRA (MTU7.TGRA), and MTU4.TGRB (MTU7.TGRB)) set to at least double the value of the TDDRA register (TDDRB register).

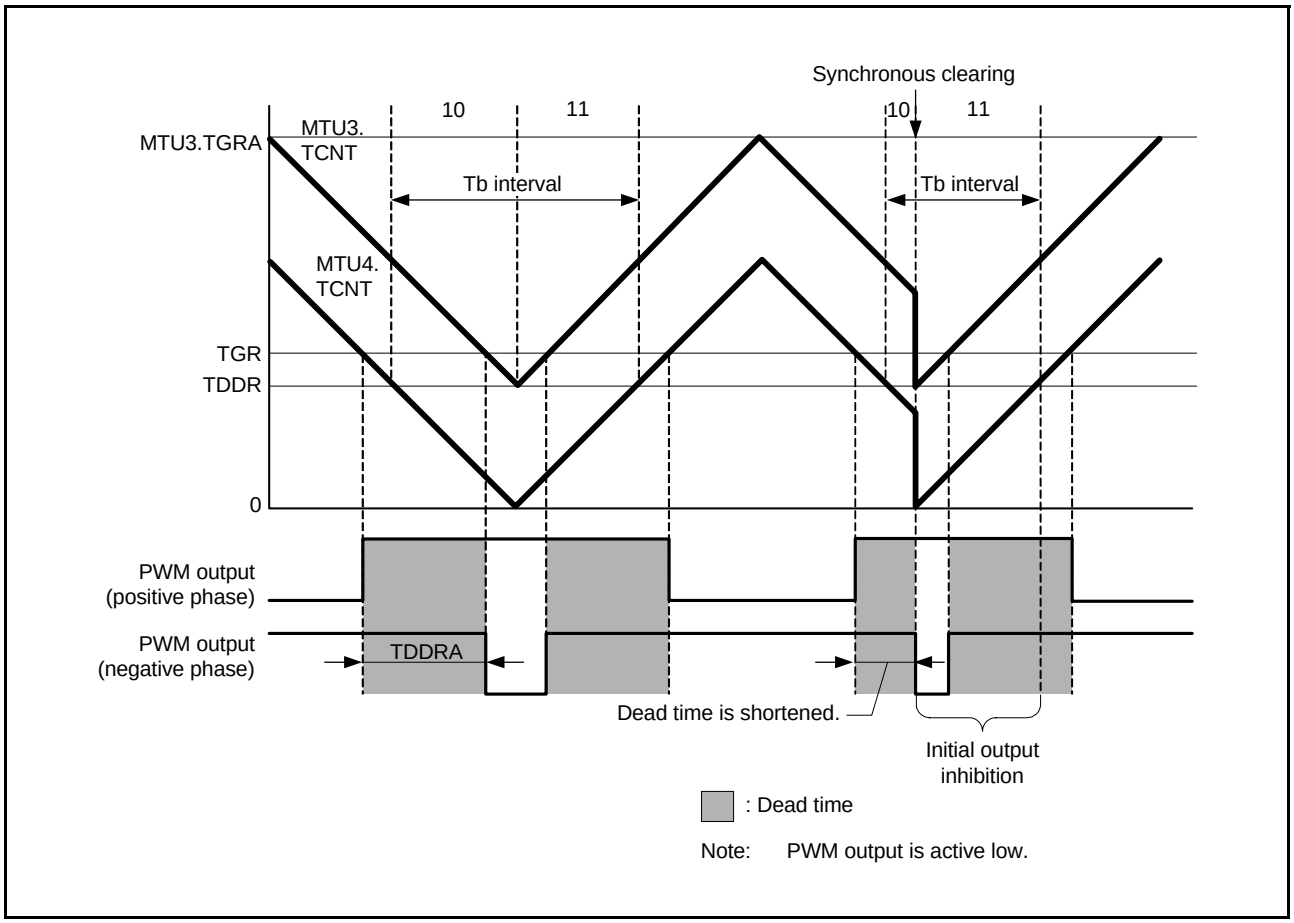


Figure 24.149 Example of Synchronous Clearing (When Condition 1 Applies)

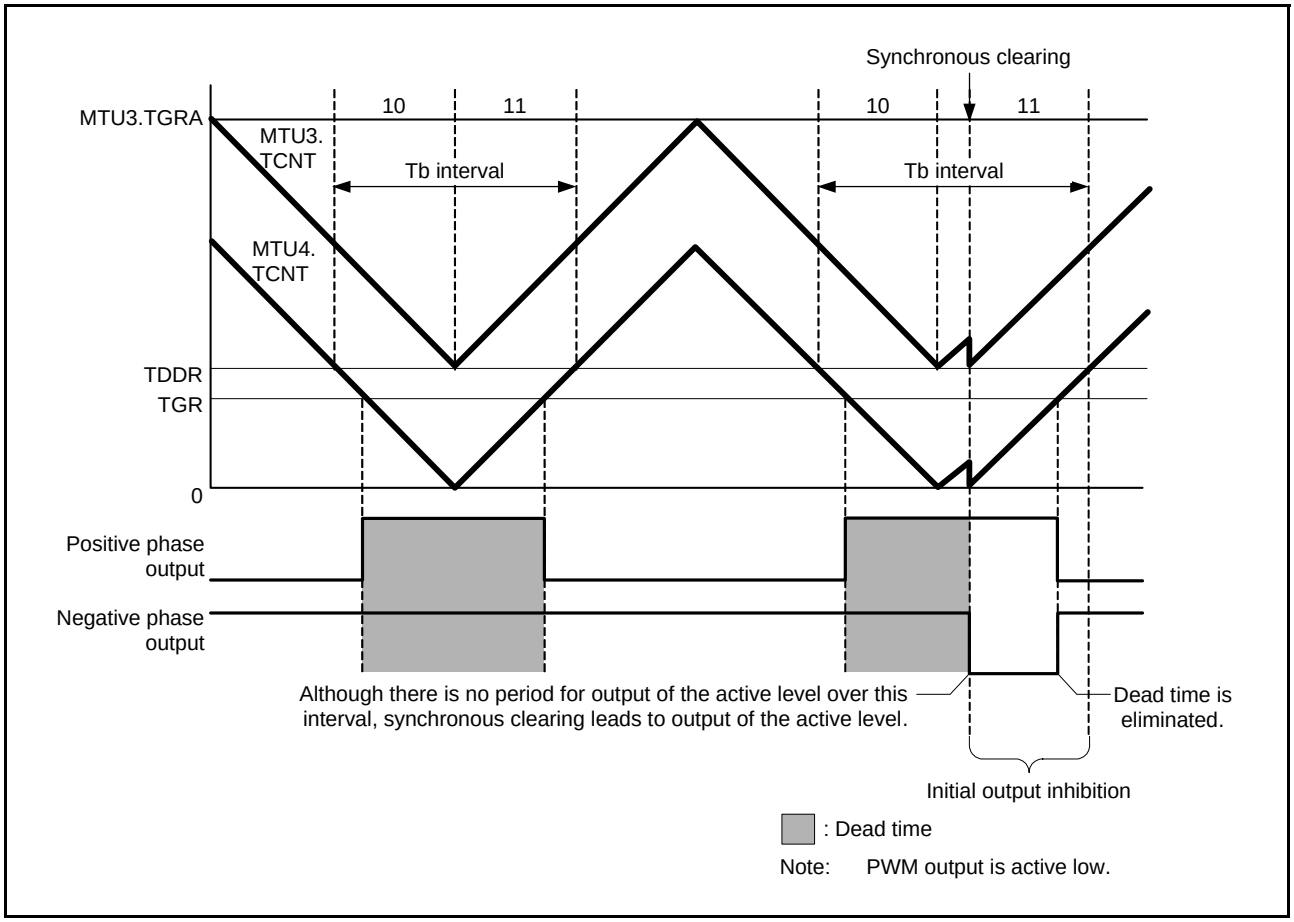


Figure 24.150 Example of Synchronous Clearing (When Condition 2 Applies)

24.6.26 Notes on Timer Mode Register Setting for ELC Event Input

When MTU is used in ELC operation, set the timer mode register (TMDR) of the relevant channel to its initial value (00h).

24.6.27 Continuous Output of Interrupt Signal in Response to a Compare Match

When the TGR register is set to 0000h, the PCLKA/1 is set as the count clock, and compare match is set as the trigger for clearing of the count clock, the value of the TCNT counter remains 0000h, and the interrupt signal will be output continuously (i.e. its level will be flat) rather than output over a single cycle. Consequently, interrupts will not be detected in response to second and subsequent compare matches.
Figure 24.151 shows the timing for continuous output of the interrupt signal in response to a compare match.

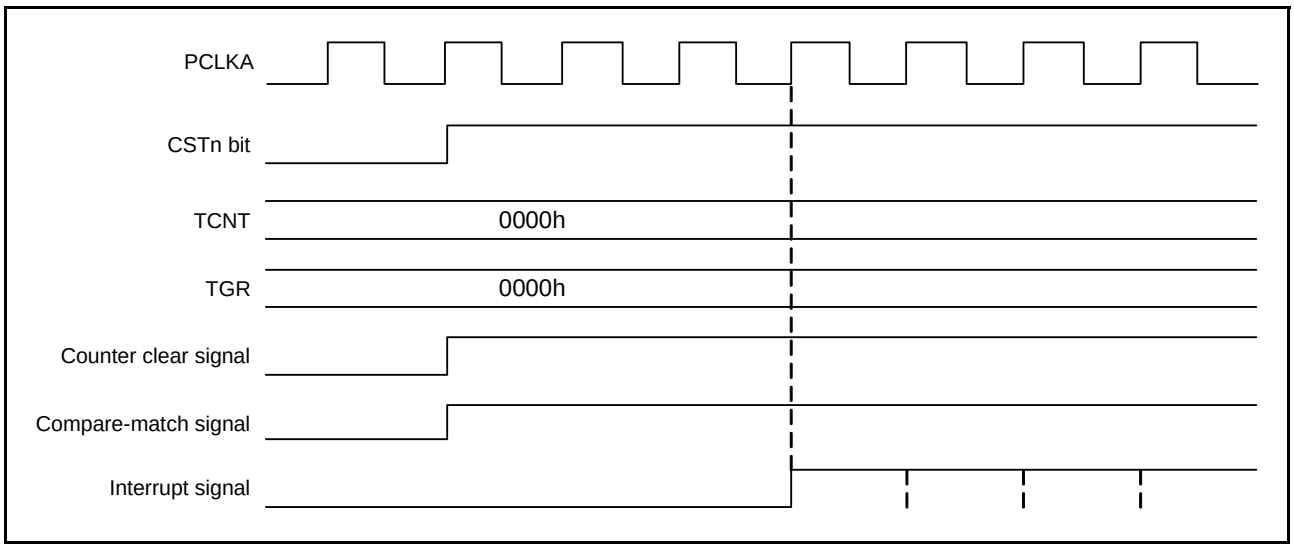


Figure 24.151 Continuous Output of Interrupt Signal in Response to a Compare Match

24.6.28 Usage Notes on A/D Conversion Start Request Delaying Function in Complementary PWM Mode

- When data is transferred from a buffer register at the trough of the MTU4.TCNT (MTU7.TCNT) counter while the MTU4.TADCOBRA and MTU4.TADCOBRB (MTU7.TADCOBRA and MTU7.TADCOBRB) registers are set to 0 and the UT4AE and UT4BE (UT7AE and UT7BE) bits in the MTU4.TADCR (MTU7.TADCR) register are set to 1, no A/D conversion start request is issued during up-counting immediately after transfer. Refer to Figure 24.152.
- When data is transferred from a buffer register at the crest of the MTU4.TCNT (MTU7.TCNT) counter while the MTU4.TADCOBRA and MTU4.TADCOBRB (MTU7.TADCOBRA and MTU7.TADCOBRB) registers are set to the same value as the TCDR and the UT4AE and UT4BE (UT7AE and UT7BE) bits in the MTU4.TADCR (MTU7.TADCR) register are set to 1, no A/D conversion start request is issued during down-counting immediately after transfer. Refer to Figure 24.153.
- To issue an A/D conversion start request linked with the interrupt skipping function, set the MTU4.TADCORA and MTU4.TADCORB (MTU7.TADCORA and MTU7.TADCORB) registers so that $2 \leq$ MTUn.TADCORA/TADCORB $\leq$ TCDR $- 2$ is satisfied (n = 4, 7).

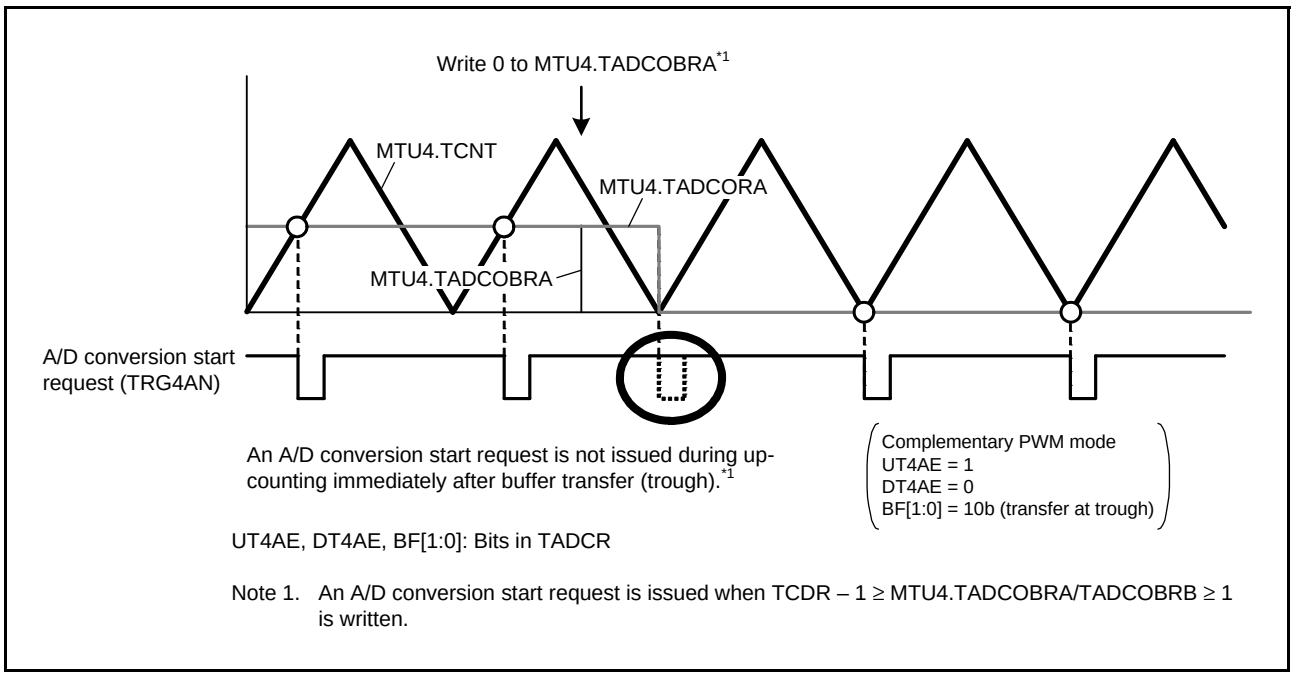


Figure 24.152 A/D Conversion Start Request When 0 is Written to MTU4.TADCOBRA

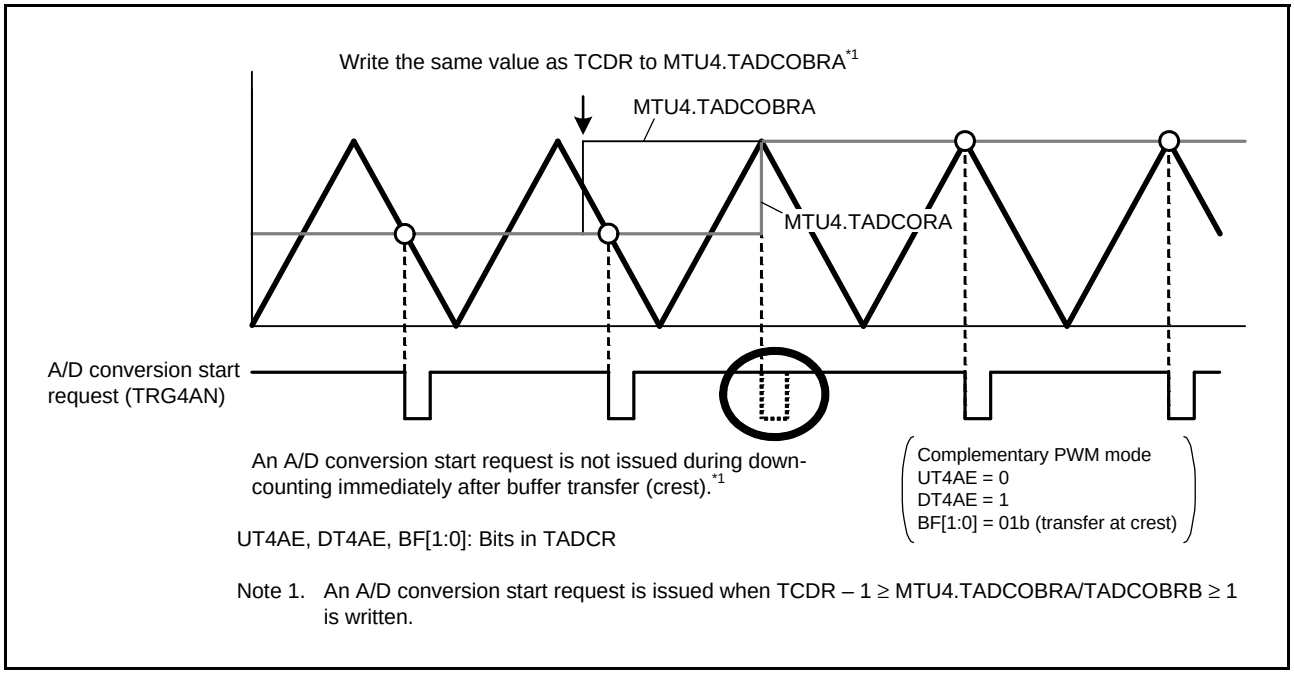


Figure 24.153 A/D Conversion Start Request When the Same Value as TCDR is Written to MTU4.TADCOBRA

24.7 MTU Output Pin Initialization

24.7.1 Operating Modes

The MTU has the following six operating modes. Waveforms can be output in any of these modes.

- Normal mode (MTU0 to MTU4 and MTU6 to MTU8)
- PWM mode 1 (MTU0 to MTU4, MTU6, and MTU7)
- PWM mode 2 (MTU0 to MTU2)
- Phase counting modes 1 to 5 (MTU1 and MTU2)
- Complementary PWM mode (MTU3, MTU4, MTU6, and MTU7)
- Reset-synchronized PWM mode (MTU3, MTU4, MTU6, and MTU7)

This section describes how to initialize the MTU output pins in each of these modes.

24.7.2 Operation in Case of Re-Setting Due to Error during Operation

If an error occurs during MTU operation, MTU output should be cut off by the system. The output can be cut off by allowing non-active level output from the pins by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports. MTU output can be disabled through TIOR settings. Complementary PWM output (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D, MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7B, MTIOC7C, and MTIOC7D) should be specified through TOERA and TOERB settings. For PWM output pins, output can also be cut by hardware, using port output enable 3 (POE3). The pin initialization procedures for re-setting due to an error during operation and the procedures for restarting in a different mode after re-setting are described below.

The MTU has six operating modes, as stated above. There are thus 36 mode transition combinations, but some transitions are not available with certain channel and mode combinations. Available mode transition combinations are shown in Table 24.79.

Table 24.79 Mode Transition Combinations

	Normal	PWM1	PWM2	PCM	CPWM	RPWM
Normal	(1)	(2)	(3)	(4)	(5)	(6)
PWM1	(7)	(8)	(9)	(10)	(11)	(12)
PWM2	(13)	(14)	(15)	(16)	Not available	Not available
PCM	(17)	(18)	(19)	(20)	Not available	Not available
CPWM	(21)	(22)	Not available	Not available	(23) (24)	(25)
RPWM	(26)	(27)	Not available	Not available	(28)	(29)

Normal: Normal mode
PWM1: PWM mode 1
PWM2: PWM mode 2
PCM: Phase counting modes 1 to 5
CPWM: Complementary PWM mode
RPWM: Reset-synchronized PWM mode

24.7.3 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation

- When making a transition to a mode (Normal, PWM1, PWM2, or PCM) in which the pin output level is selected by the timer I/O control register (TIOR) setting, initialize the pins by means of TIOR setting.
- In PWM mode 1, waveforms are not output to the MTIOCnB and MTIOCnD (n = 3, 4, 6, 7) pins. When a pin is configured for MTIOCnB or MTIOCnD, it enters high-impedance state. To output a specified level, set the pin to general output port.
- In PWM mode 2, a waveform is not output to the pin that corresponds to the TGR register used as a period register. When a pin is configured for MTIOCnm (n = 0 to 2; m = A to D), it enters high-impedance state. To output a specified level, set the pin to general output port.
- In normal mode or PWM mode 2, if TGRC and TGRD operate as buffer registers, waveforms are not output to the corresponding pins (MTIOCnC or MTIOCnD (n = 0, 3, 4, 6, 7)). When a pin is configured for MTIOCnC or MTIOCnD, it enters high-impedance state. To output a specified level, set the pin to general output port.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register, waveforms are not output to the corresponding pins (MTIOCnC or MTIOCnD (n = 0, 3, 4, 6, 7)). When a pin is configured for MTIOCnC or MTIOCnD, it enters high-impedance state. To output a specified level, set the pin to general output port.
- When making a transition to a mode (CPWM or RPWM) in which the pin output level is selected by the timer output control register (TOCR1A, TOCR2A, TOCR1B, or TOCR2B) setting, temporarily disable output in MTU3 and MTU4 (or MTU6 and MTU7) with the timer output master enable register (TOERA or TOERB). At this time, when a pin is configured for MTIOCnm (n = 3, 4, 6, 7; m = A to D), it enters high-impedance state. To output a specified level, set the pin to general output port. Switch to normal mode, perform initialization with TIOR, restore TIOR to its initial value, then operate the MTU in accordance with the mode setting procedure (TOCR1A setting, TOCR2A setting, TMDR1 setting, and TOERA setting (TOCR1B setting, TOCR2B setting, TMDR1 setting, and TOERB setting)).

Note: Channel number is substituted for “n” indicated in this section.

Pin initialization procedures are described below for the numbered combinations in Table 24.79. The active level is assumed to be low.

(1) Operation When Error Occurs in Normal Mode and Operation is Restarted in Normal Mode

Figure 24.154 shows a case in which an error occurs in normal mode and operation is restarted in normal mode after re-setting.

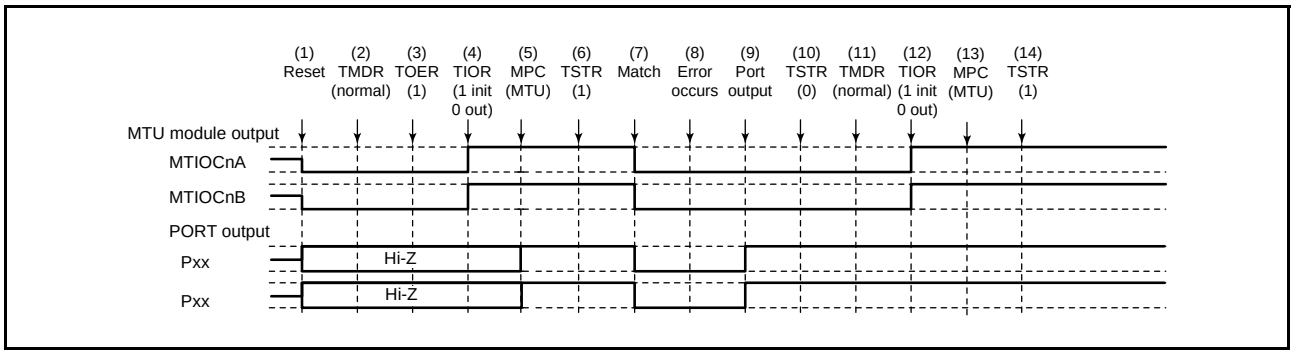


Figure 24.154 Error Occurrence in Normal Mode, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.
(2) After a reset, the TMDR1 setting is for normal mode.
(3) For MTU3 and MTU4 (MTU6 and MTU7), enable output with the TOERA (TOERB) register before initializing the pins with the TIOR register.
(4) Initialize the pins with the TIOR register. (In the example, the initial output is a high level, and a low level is output on compare match occurrence.)
(5) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(6) Start count operation by setting the TSTRA (TSTRB) register.
(7) Output goes low on compare match occurrence.
(8) An error occurs.
(9) Allow non-active level output by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(10) Stop count operation by setting the TSTRA (TSTRB) register.
(11) This step is not necessary when restarting in normal mode.
(12) Initialize the pins with the TIOR register.
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA (TSTRB) register.

(2) Operation When Error Occurs in Normal Mode and Operation is Restarted in PWM Mode 1

Figure 24.155 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.

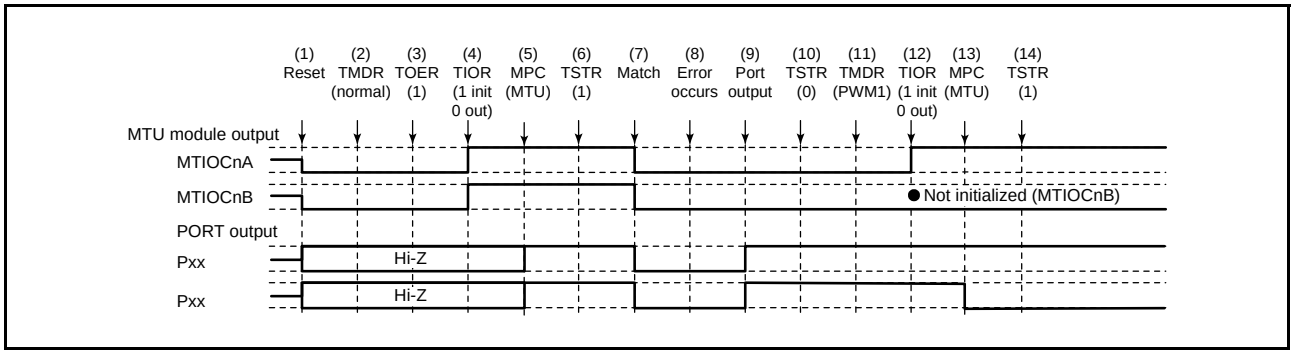


Figure 24.155 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 24.154.
(11) Set PWM mode 1.
(12) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA (TSTRB) register.

(3) Operation When Error Occurs in Normal Mode and Operation is Restarted in PWM mode 2

Figure 24.156 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 2 after re-setting.

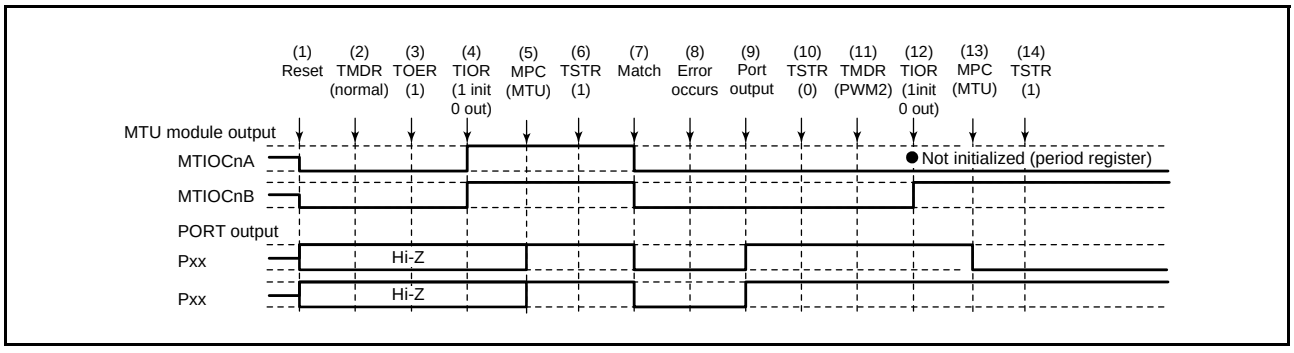


Figure 24.156 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

(1) to (10) are the same as in Figure 24.154.
(11) Set PWM mode 2.
(12) Initialize the pins with the TIOR register. (In PWM mode 2, a waveform is not output to the pin that corresponds to the TGR register used as a period register. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

Note: PWM mode 2 can only be selected for MTU0 to MTU2, and therefore the TOERA register setting is not necessary.

(4) Operation When Error Occurs in Normal Mode and Operation is Restarted in Phase Counting Mode

Figure 24.157 shows a case in which an error occurs in normal mode and operation is restarted in phase counting mode after re-setting.

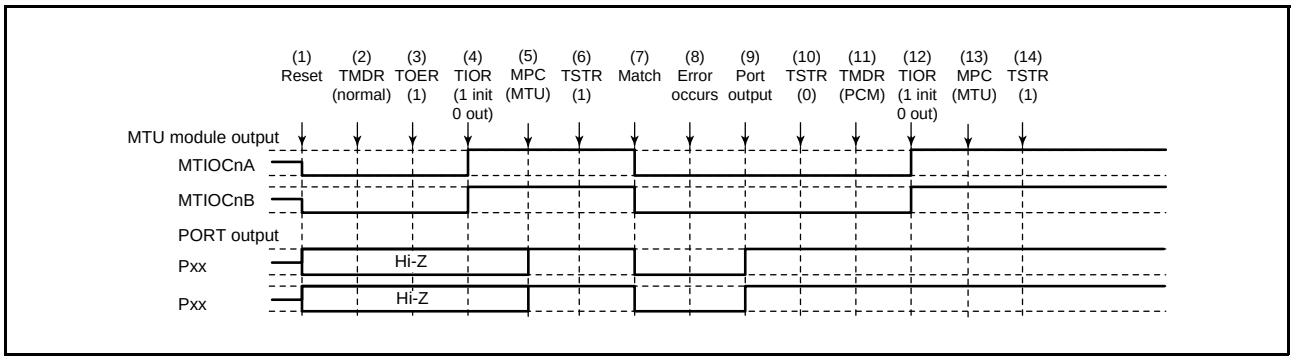


Figure 24.157 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

(1) to (10) are the same as in Figure 24.154.
(11) Set the phase counting mode.
(12) Initialize the pins with the TIOR register.
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

Note: The phase counting mode can only be selected for MTU1 and MTU2, and therefore the TOERA register setting is not necessary.

(5) Operation When Error Occurs in Normal Mode and Operation is Restarted in Complementary PWM Mode

Figure 24.158 shows a case in which an error occurs in normal mode and operation is restarted in complementary PWM mode after re-setting.

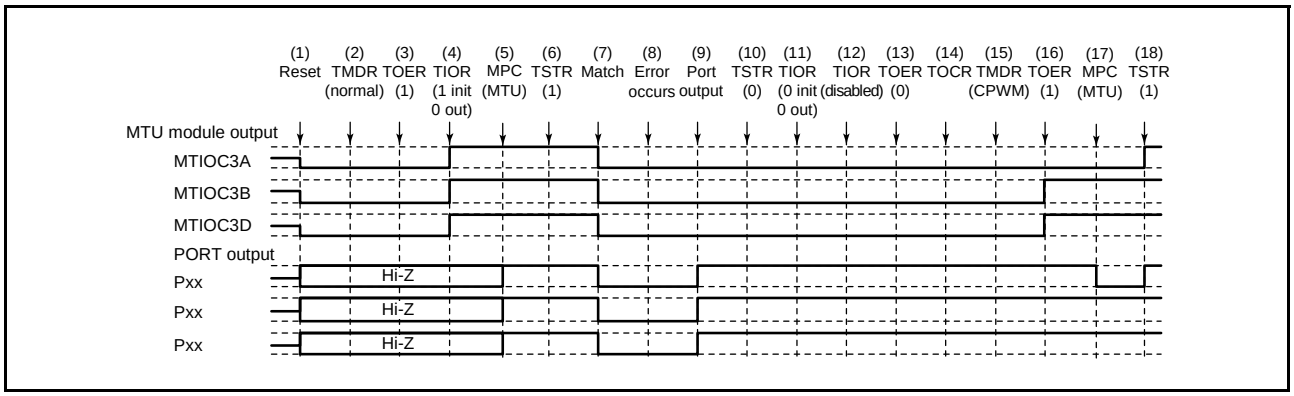


Figure 24.158 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

(1) to (10) are the same as in Figure 24.154.
(11) Initialize the normal mode waveform generation block with the TIOR register.
(12) Disable operation of the normal mode waveform generation block with the TIOR register.
(13) Disable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.
(14) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).
(15) Set complementary PWM mode.
(16) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.
(17) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(18) Restart operation by setting the TSTRA (TSTRB) register.

(6) Operation When Error Occurs in Normal Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 24.159 shows a case in which an error occurs in normal mode and operation is restarted in reset-synchronized PWM mode after re-setting.

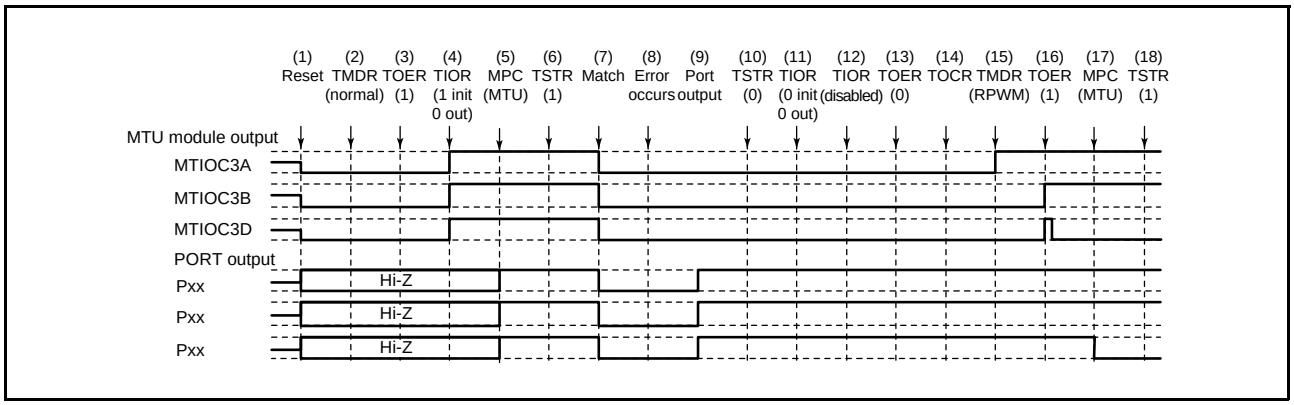


Figure 24.159 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (13) are the same as in Figure 24.158.

(14) Select the reset-synchronized PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(15) Set reset-synchronized PWM mode.

(16) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(17) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(18) Restart operation by setting the TSTRA (TSTRB) register.

(7) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Normal Mode

Figure 24.160 shows a case in which an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.

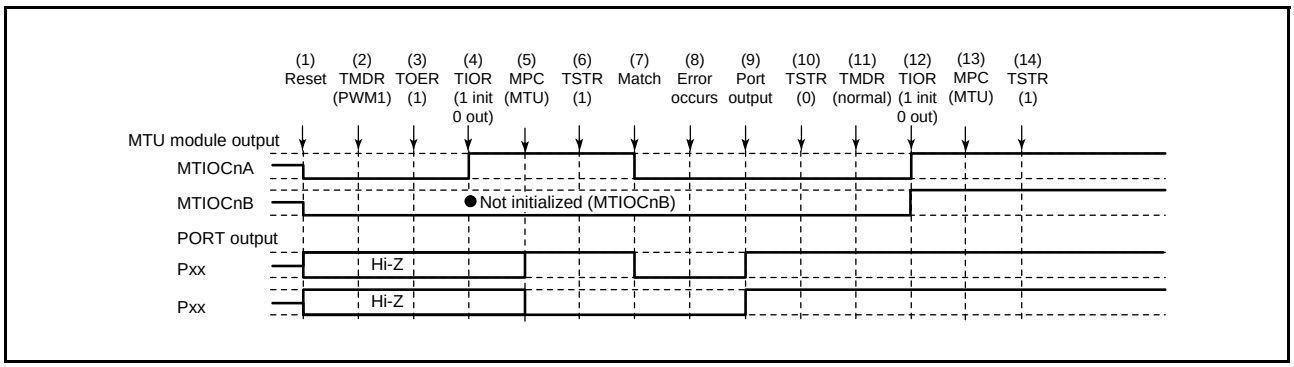


Figure 24.160 Error Occurrence in PWM Mode 1, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.
(2) Set PWM mode 1.
(3) For MTU3 and MTU4 (MTU6 and MTU7), enable output with the TOERA (TOERB) register before initializing the pins with the TIOR register.
(4) Initialize the pins with the TIOR register. (In the example, the initial output is a high level, and a low level is output on compare match occurrence. In PWM mode 1, the MTIOCnB side is not initialized.)
(5) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(6) Start count operation by setting the TSTRA (TSTRB) register.
(7) Output goes low on compare match occurrence.
(8) An error occurs.
(9) Allow non-active level output by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(10) Stop count operation by setting the TSTRA (TSTRB) register.
(11) Set normal mode.
(12) Initialize the pins with the TIOR register.
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA (TSTRB) register.

(8) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 1

Figure 24.161 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 1 after re-setting.

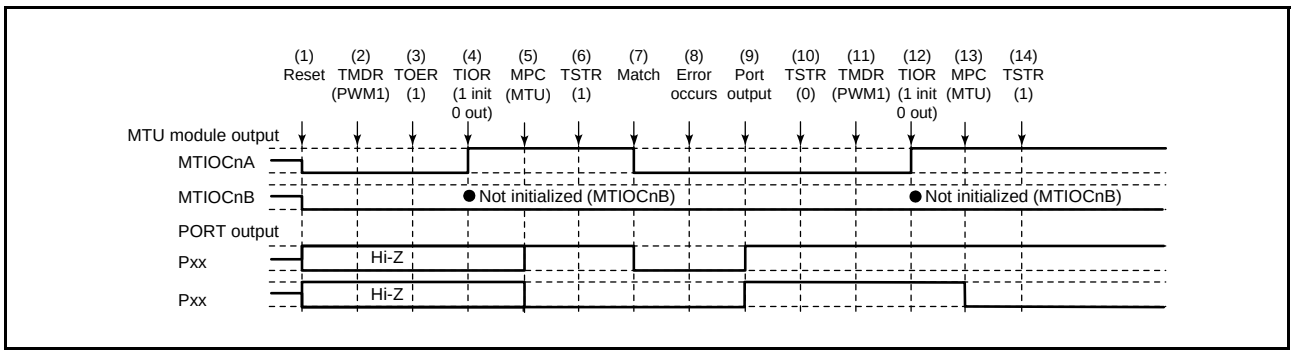


Figure 24.161 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 24.160.
(11) This step is not necessary when restarting in PWM mode 1.
(12) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA (TSTRB) register.

(9) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 2

Figure 24.162 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 2 after re-setting.

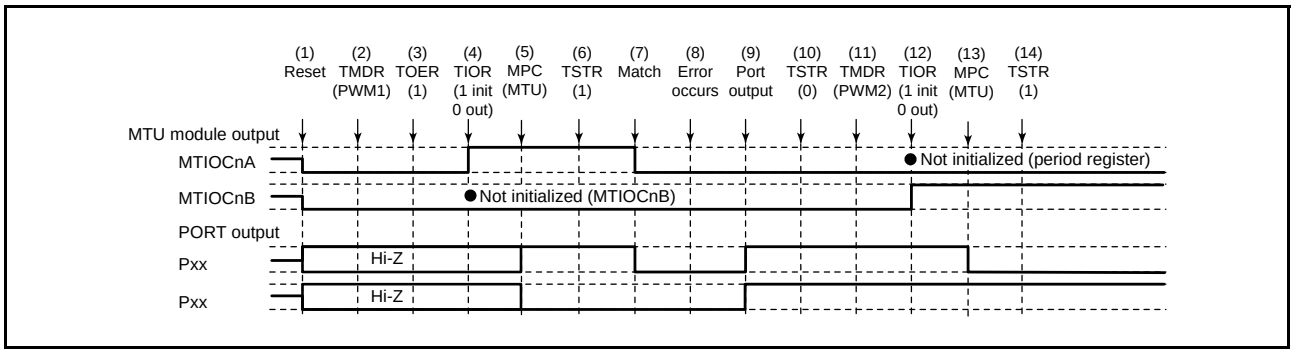


Figure 24.162 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

(1) to (10) are the same as in Figure 24.160.
(11) Set PWM mode 2.
(12) Initialize the pins with the TIOR register. (In PWM mode 2, a waveform is not output to the pin that corresponds to the TGR register used as a period register. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

Note: PWM mode 2 can only be selected for MTU0 to MTU2, and therefore the TOERA register setting is not necessary.

(10) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Phase Counting Mode

Figure 24.163 shows a case in which an error occurs in PWM mode 1 and operation is restarted in phase counting mode after re-setting.

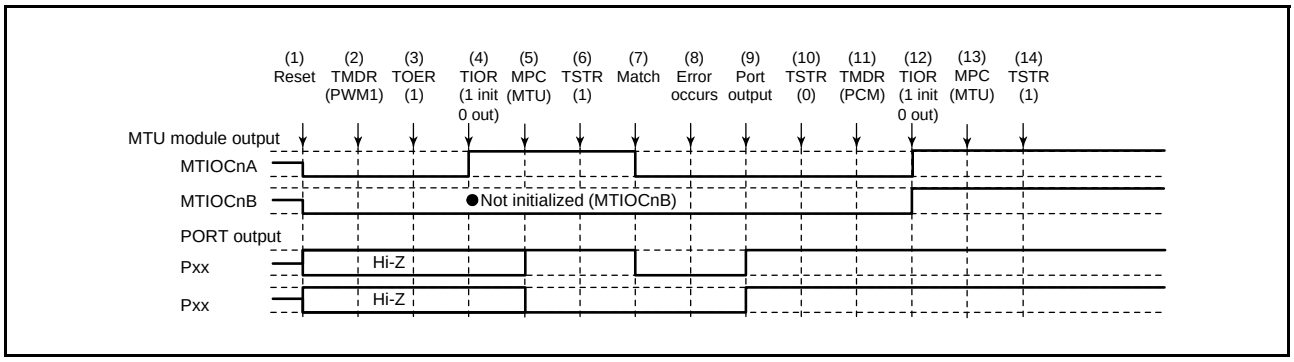


Figure 24.163 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

(1) to (10) are the same as in Figure 24.160.
(11) Set the phase counting mode.
(12) Initialize the pins with the TIOR register.
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

Note: The phase counting mode can only be selected for MTU1 and MTU2, and therefore the TOERA register setting is not necessary.

(11) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Complementary PWM Mode

Figure 24.164 shows a case in which an error occurs in PWM mode 1 and operation is restarted in complementary PWM mode after re-setting.

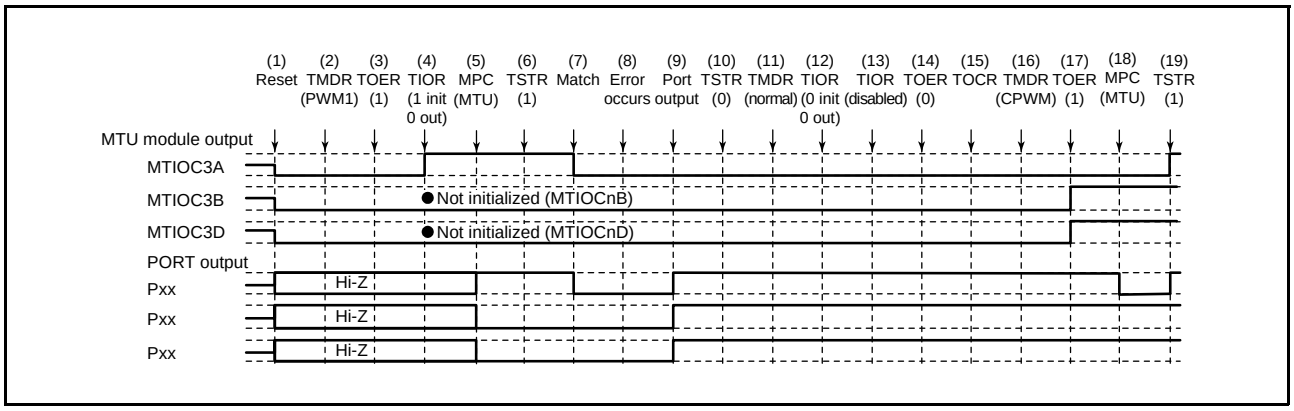


Figure 24.164 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

(1) to (10) are the same as in Figure 24.160.
(11) Set normal mode to initialize the normal mode waveform generation block.
(12) Initialize the PWM mode 1 waveform generation block with the TIOR register.
(13) Disable operation of the PWM mode 1 waveform generation block with the TIOR register.
(14) Disable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.
(15) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TSTRB).
(16) Set complementary PWM mode.
(17) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.
(18) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(19) Restart operation by setting the TSTRA (TSTRB) register.

(12) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 24.165 shows a case in which an error occurs in PWM mode 1 and operation is restarted in reset-synchronized PWM mode after re-setting.

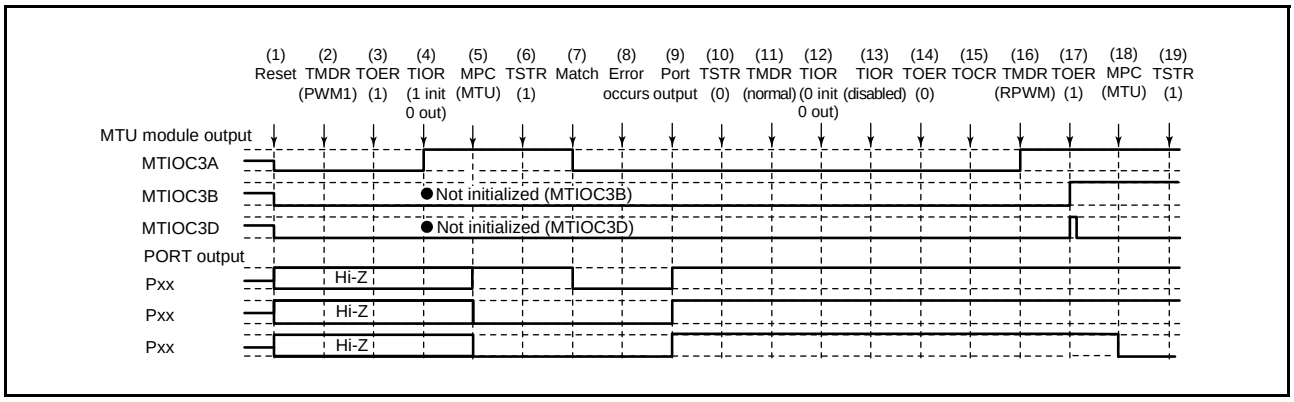


Figure 24.165 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode

(1) to (14) are the same as in Figure 24.164.

(15) Select the reset-synchronized PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(16) Set reset-synchronized PWM mode.

(17) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(18) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(19) Restart operation by setting the TSTRA (TSTRB) register.

(13) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in Normal Mode

Figure 24.166 shows a case in which an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.

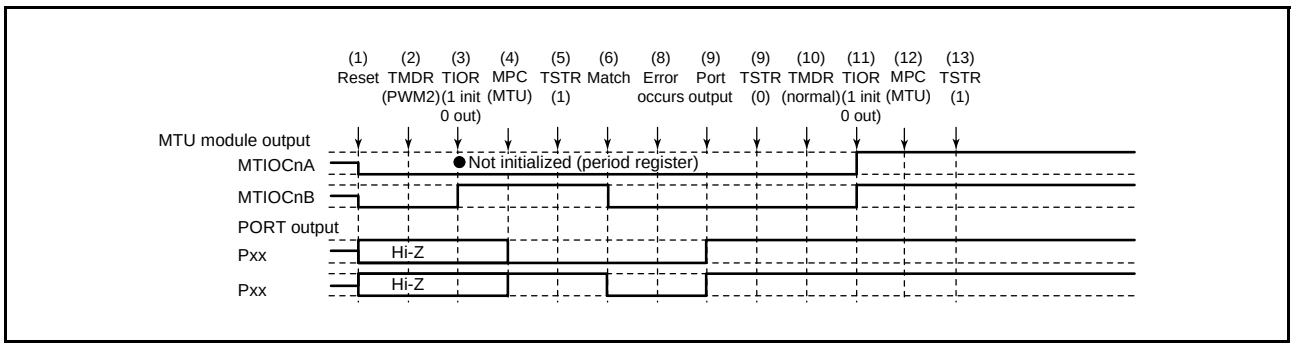


Figure 24.166 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.
(2) Set PWM mode 2.
(3) Initialize the pins with the TIOR register. (In the example, the initial output is a high level, and a low level is output on compare match occurrence. In PWM mode 2, the pin that corresponds to the TGR register used as a period register is not initialized. In the example, the MTUn.TGRA register is used as a period register.)
(4) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(5) Start count operation by setting the TSTRA register.
(6) Output goes low on compare match occurrence.
(7) An error occurs.
(8) Allow non-active level output by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(9) Stop count operation by setting the TSTRA register.
(10) Set normal mode.
(11) Initialize the pins with the TIOR register.
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(14) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 1

Figure 24.167 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 1 after re-setting.

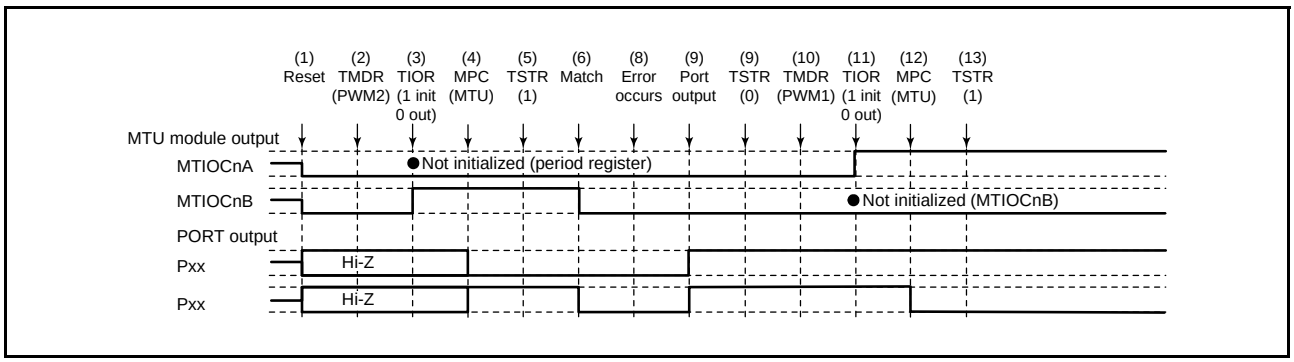


Figure 24.167 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

(1) to (9) are the same as in Figure 24.166.
(10) Set PWM mode 1.
(11) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(15) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 2

Figure 24.168 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

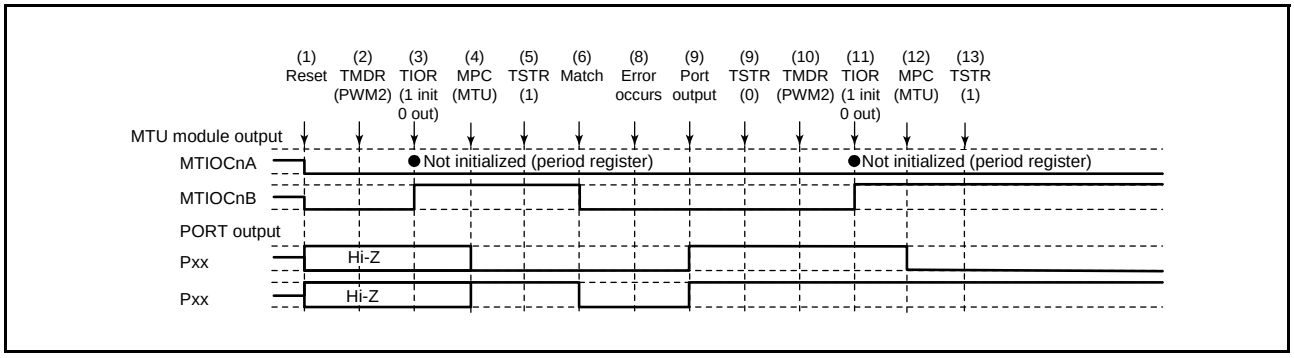


Figure 24.168 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

(1) to (9) are the same as in Figure 24.166.
(10) This step is not necessary when restarting in PWM mode 2.
(11) Initialize the pins with the TIOR register. (In PWM mode 2, a waveform is not output to the pin that corresponds to the TGR register used as a period register. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(16) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in Phase Counting Mode

Figure 24.169 shows a case in which an error occurs in PWM mode 2 and operation is restarted in phase counting mode after re-setting.

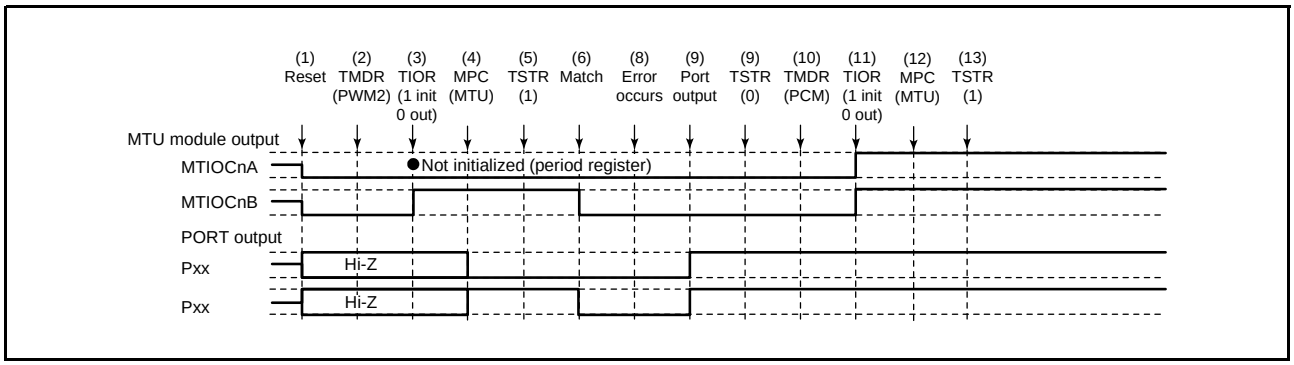


Figure 24.169 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

(1) to (9) are the same as in Figure 24.166.
(10) Set the phase counting mode.
(11) Initialize the pins with the TIOR register.
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(17) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in Normal Mode

Figure 24.170 shows a case in which an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.

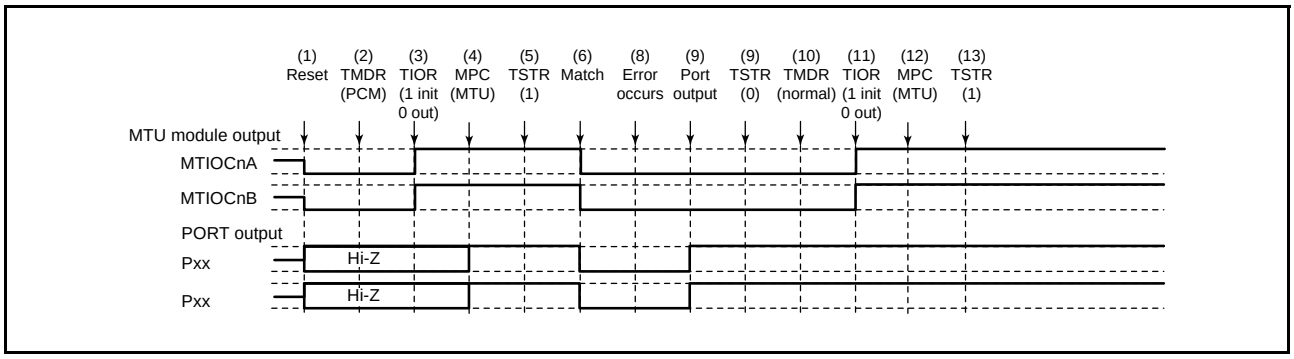


Figure 24.170 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.
(2) Set phase counting mode.
(3) Initialize the pins with the TIOR register. (In the example, the initial output is a high level, and a low level is output on compare match occurrence.)
(4) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(5) Start count operation by setting the TSTRA register.
(6) Output goes low on compare match occurrence.
(7) An error occurs.
(8) Allow non-active level output by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(9) Stop count operation by setting the TSTRA register.
(10) Set normal mode.
(11) Initialize the pins with the TIOR register.
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(18) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 1

Figure 24.171 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 1 after re-setting.

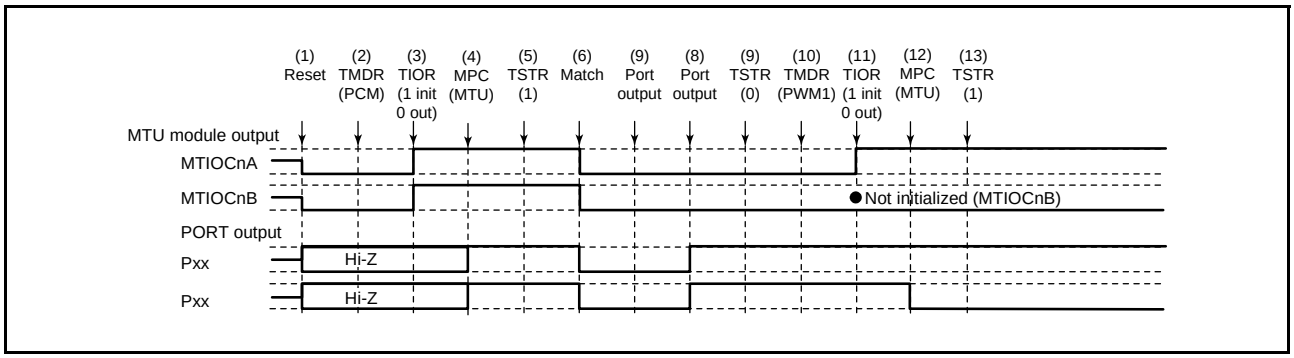


Figure 24.171 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

(1) to (9) are the same as in Figure 24.170.
(10) Set PWM mode 1.
(11) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(19) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 2

Figure 24.172 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 2 after re-setting.

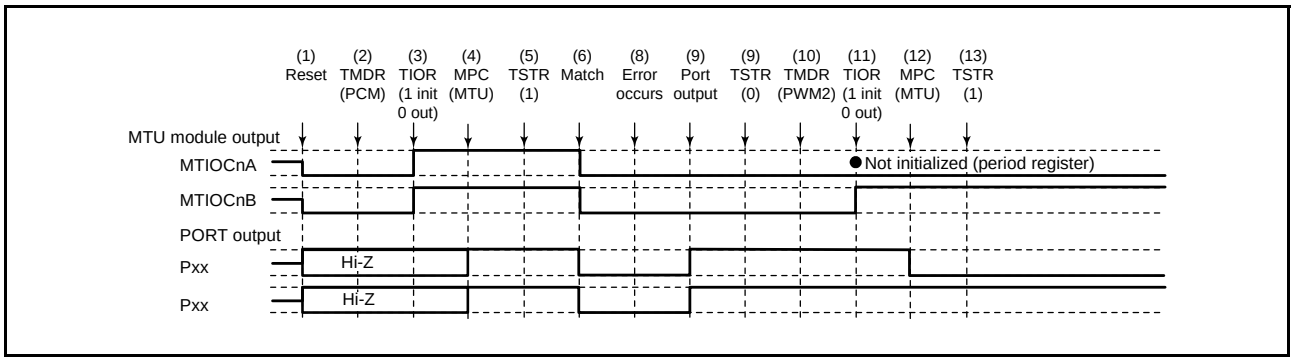


Figure 24.172 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

(1) to (9) are the same as in Figure 24.170.
(10) Set PWM mode 2.
(11) Initialize the pins with the TIOR register. (In PWM mode 2, a waveform is not output to the pin that corresponds to the TGR register used as a period register. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(20) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in Phase Counting Mode

Figure 24.173 shows a case in which an error occurs in phase counting mode and operation is restarted in phase counting mode after re-setting.

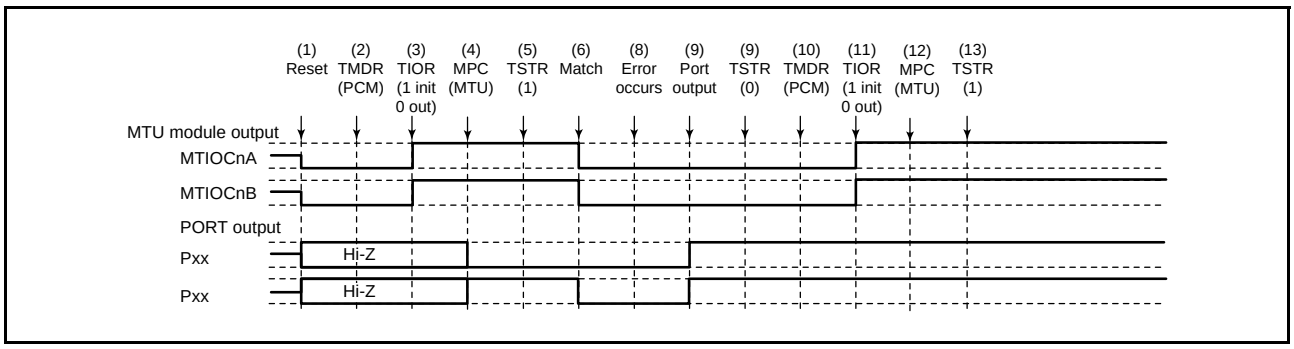


Figure 24.173 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

(1) to (9) are the same as in Figure 24.170.
(10) This step is not necessary when restarting in phase counting mode.
(11) Initialize the pins with the TIOR register.
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(21) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Normal Mode

Figure 24.174 shows a case in which an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.

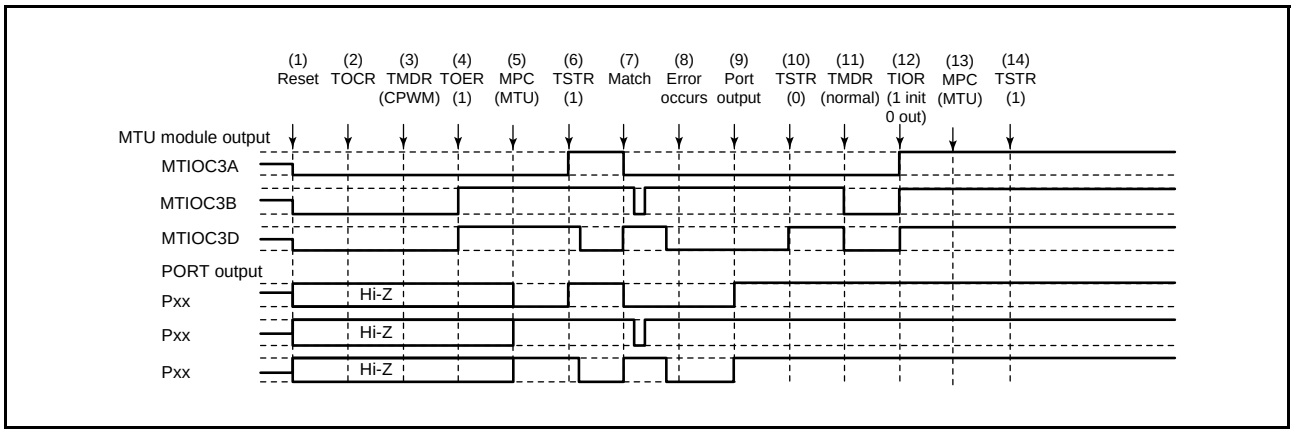


Figure 24.174 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

(1) After a reset, the MTU3 output goes low and the ports enter high-impedance state.
(2) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).
(3) Set complementary PWM mode.
(4) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.
(5) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(6) Start count operation by setting the TSTRA (TSTRB) register.
(7) The complementary PWM waveform is output on compare match occurrence.
(8) An error occurs.
(9) Allow non-active level output by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(10) Stop count operation by setting the TSTRA (TSTRB) register. (MTU output becomes the initial complementary PWM output value).
(11) Set normal mode (MTU output goes low).
(12) Initialize the pins with the TIOR register.
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA (TSTRB) register.

(22) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in PWM Mode 1

Figure 24.175 shows a case in which an error occurs in complementary PWM mode and operation is restarted in PWM mode 1 after re-setting.

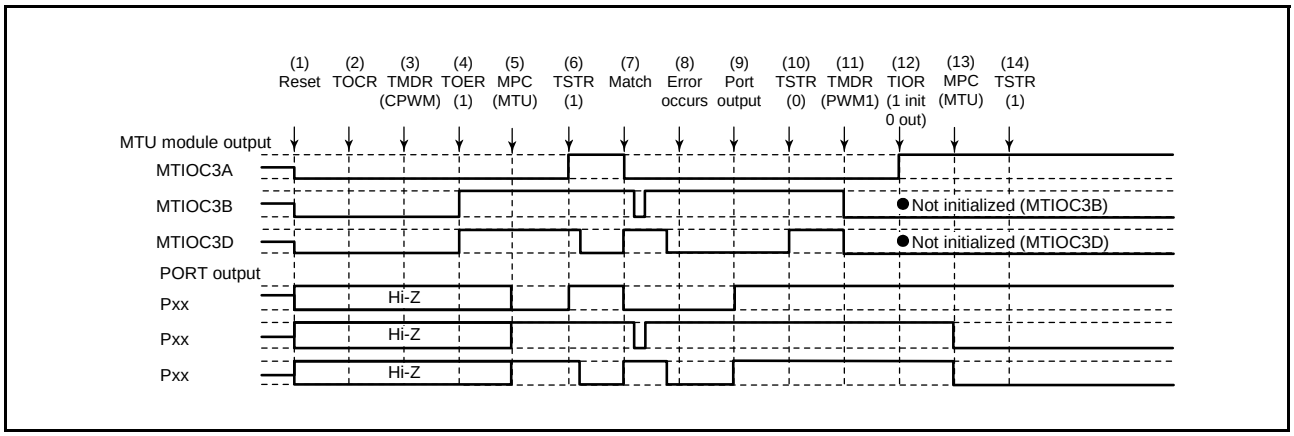


Figure 24.175 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 24.174.
(11) Set PWM mode 1 (MTU output goes low).
(12) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA (TSTRB) register.

(23) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 24.176 shows a case in which an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (when operation is restarted using the period and duty settings at the time of stopping the counter).

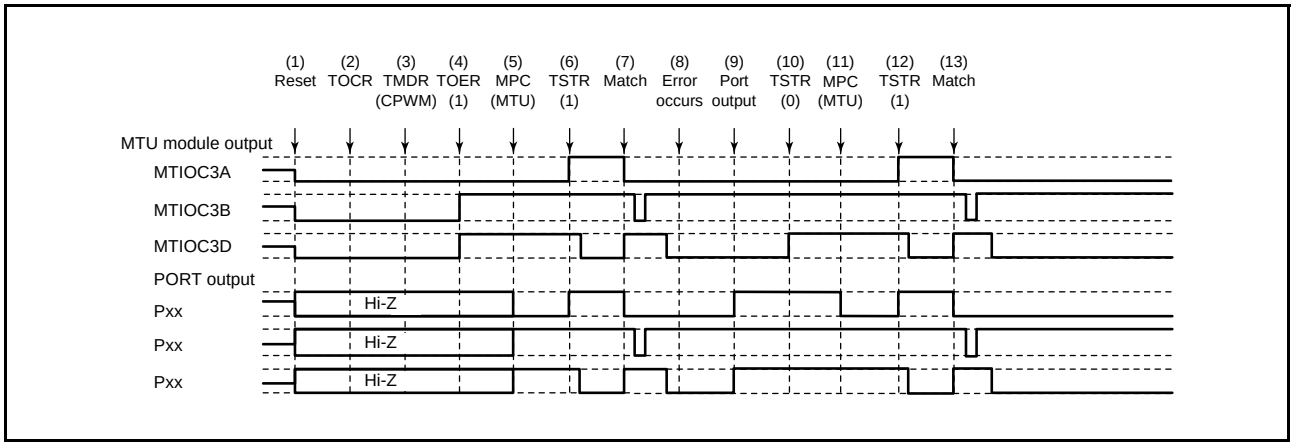


Figure 24.176 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode (1)

(1) to (10) are the same as in Figure 24.174.
(11) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(12) Restart operation by setting the TSTRA (TSTRB) register.
(13) The complementary PWM waveform is output on compare match occurrence.

(24) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Complementary PWM Mode with New Settings

Figure 24.177 shows a case in which an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (operation is restarted using new period and duty ratio settings).

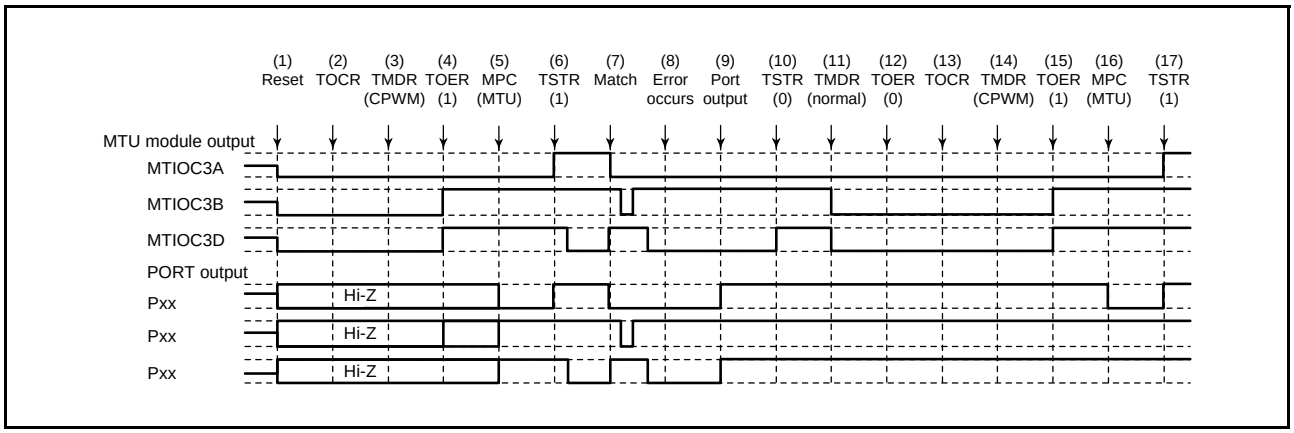


Figure 24.177 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode (2)

(1) to (10) are the same as in Figure 24.174.
(11) Set normal mode and make new settings (MTU output goes low).
(12) Disable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.
(13) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).
(14) Set complementary PWM mode.
(15) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.
(16) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(17) Restart operation by setting the TSTRA (TSTRB) register.

(25) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 24.178 shows a case in which an error occurs in complementary PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

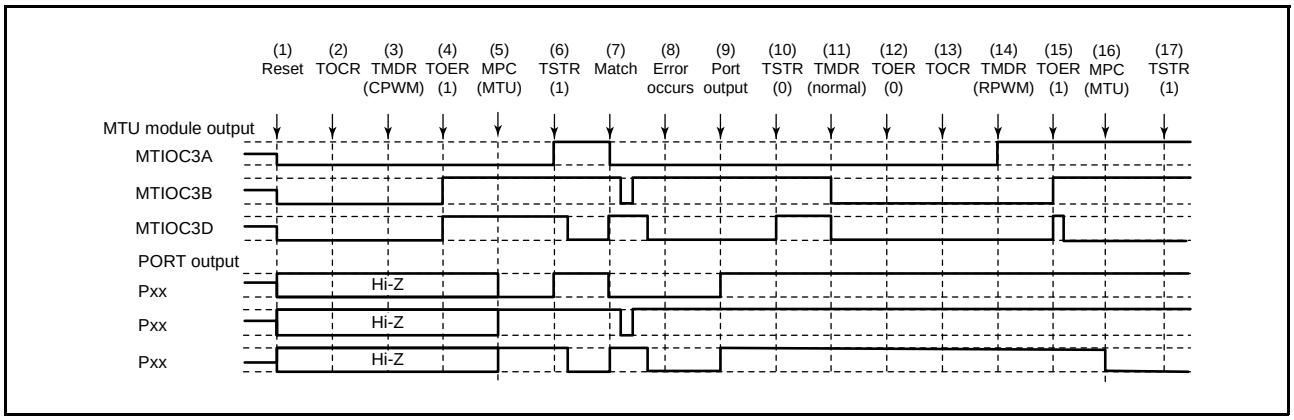


Figure 24.178 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (10) are the same as in Figure 24.174.
(11) Set normal mode (MTU output goes low).
(12) Disable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.
(13) Select the reset-synchronized PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).
(14) Set reset-synchronized PWM mode.
(15) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.
(16) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(17) Restart operation by setting the TSTRA (TSTRB) register.

(26) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Normal Mode

Figure 24.179 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in normal mode after re-setting.

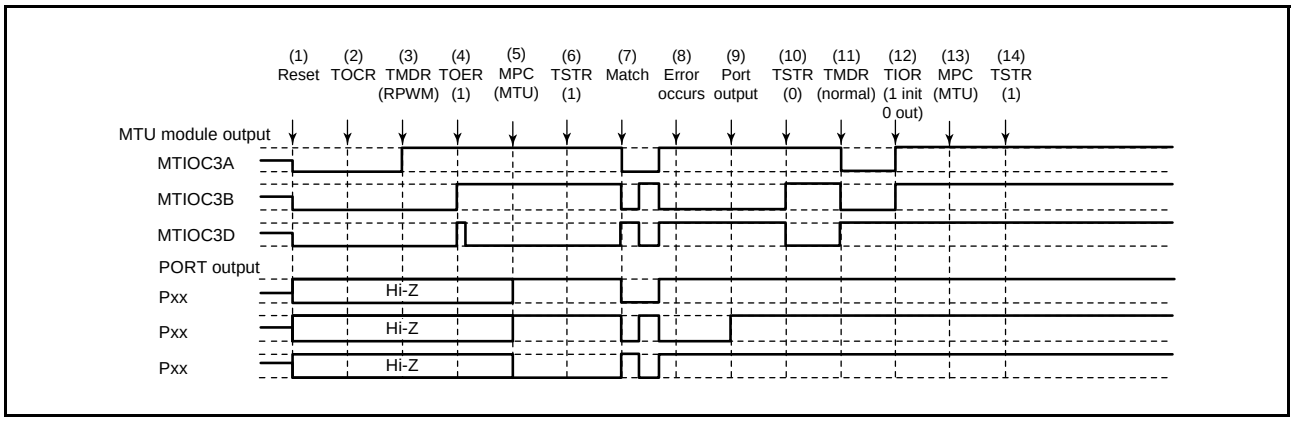


Figure 24.179 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.
(2) Select the reset-synchronized PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).
(3) Set reset-synchronized PWM mode.
(4) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.
(5) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(6) Start count operation by setting the TSTRA (TSTRB) register.
(7) The reset-synchronized PWM waveform is output on compare match occurrence.
(8) An error occurs.
(9) Allow non-active level output by setting the pins as general output ports using the port direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(10) Stop count operation by setting the TSTRA (TSTRB) register. (MTU output becomes the initial reset-synchronized PWM output value.)
(11) Set normal mode (positive-phase MTU output goes low, and negative-phase output goes high).
(12) Initialize the pins with the TIOR register.
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA (TSTRB) register.

(27) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in PWM Mode 1

Figure 24.180 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in PWM mode 1 after re-setting.

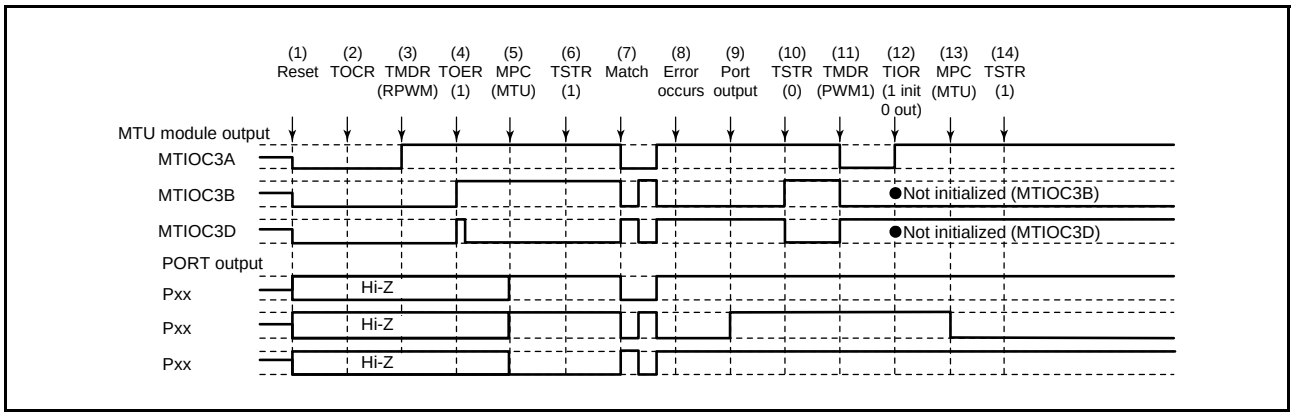


Figure 24.180 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 24.179.

(11) Set PWM mode 1 (positive-phase MTU output goes low, and negative-phase output goes high).

(12) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port direction registers (PDR) and port output data registers (PODR) of the I/O ports.)

(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA (TSTRB) register.

(28) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 24.181 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in complementary PWM mode after re-setting.

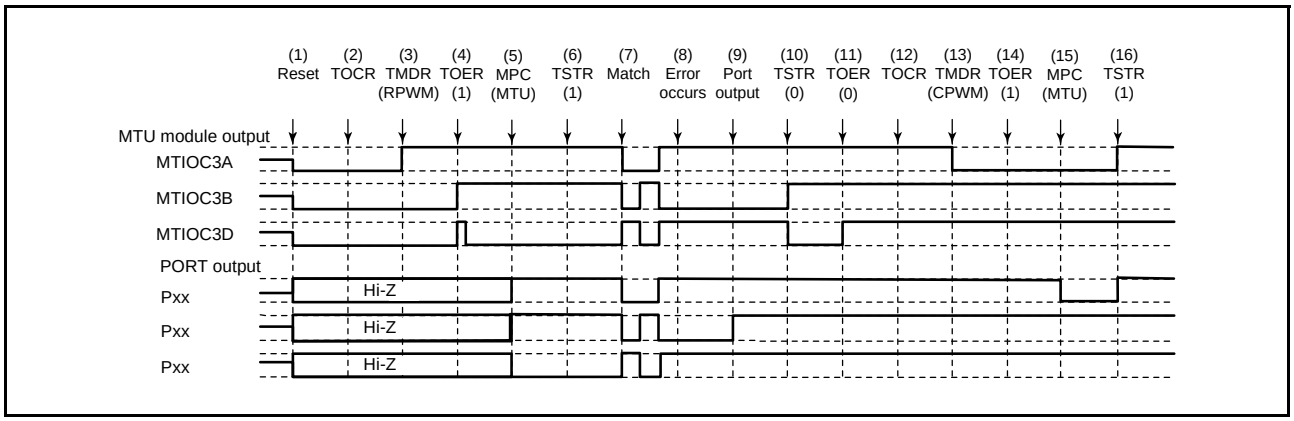


Figure 24.181 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode

(1) to (10) are the same as in Figure 24.179.
(11) Disable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.
(12) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).
(13) Set complementary PWM mode (MTU cyclic output pin goes low).
(14) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.
(15) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(16) Restart operation by setting the TSTRA (TSTRB) register.

(29) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 24.182 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

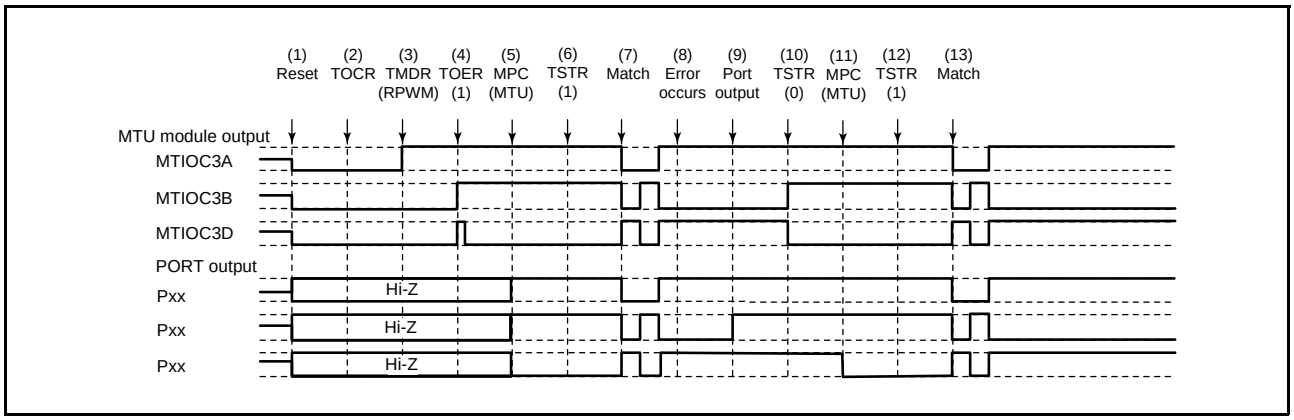


Figure 24.182 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (10) are the same as in Figure 24.179.

(11) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(12) Restart operation by setting the TSTRA (TSTRB) register.

(13) The reset-synchronized PWM waveform is output on compare match occurrence.

24.8 Operations Linked by the ELC

24.8.1 Event Signal Output to the ELC

The MTU is capable of operation linked with another module set in advance when its interrupt request signal is used as an event signal by the event link controller (ELC).
The MTU outputs the event signal regardless of the setting of the corresponding interrupt request enable bit.

24.8.2 MTU Operations in Response to Receiving Event Signals from the ELC

The MTU can perform the following operations in response to the event set in advance in the ELSRn register of the event link controller (ELC).

(1) Start Counting Operation

Select “counting is started” as the operation of the MTU by setting the ELOPA or ELOPB register in the ELC. The ELOPA register controls the operation of MTU0 and MTU3 and the ELOPB register controls the operation of MTU4. When the event specified in the ELSRn register occurs, the CSTn bit in the TSTRA register shown in Table 24.80 is set to 1, and the MTU counter starts.
However, when the specified event is generated while the CSTn bit in the TSTRA register has already been set to 1, the event has no effect. Table 24.80 lists the TSTRA register bits used for each channel.

Table 24.80 Counter Start Bit Set by the ELC

Channel No.	Counter Start Bit
MTU0	TSTRA.CST0 bit
MTU3	TSTRA.CST3 bit
MTU4	TSTRA.CST4 bit

(2) Input Capture Operation

Select “input capture” as the operation of the MTU by setting the ELOPA or ELOPB register in the ELC. The ELOPA register controls the operation of MTU0 and MTU3 and the ELOPB register controls the operation of MTU4. When the event specified in the ELSRn register occurs, the value of the TCNT register is captured in the TGR register. When using input capture in response to an event, the corresponding bit of the TIOR register in the MTU should be set for input capture and the CSTn bit of TSTRA register should be set to 1 to start counting by the counter.
In this case, the TIOCnA pin (input capture pin) input has no effect.
Table 24.81 lists the timer general register and I/O control bit used for each channel in input capture operations in response to the ELC.

Table 24.81 Timer General Register and I/O Control Bit Used in the Input Capture Operation

Channel No.	Timer General Register	I/O Control Bit
MTU0	MTU0.TGRA	MTU0.TIORH.IOA[3:0] bits
MTU3	MTU3.TGRA	MTU3.TIORH.IOA[3:0] bits
MTU4	MTU4.TGRA	MTU4.TIORH.IOA[3:0] bits

(3) Restart Counting (Clear Counter) Operation

Select “counting is restarted” as the operation of the MTU by setting the ELOPA or ELOPB register in the ELC. The ELOPA register controls the operation of MTU0 and MTU3 and the ELOPB register controls the operation of MTU4. When the event specified in the ELSRn register occurs, the TCNT register is cleared. If the corresponding CSTn bit in the TSTRA register is set to 1, counting continues. For the CSTn bits in the TSTRA register, refer to Table 24.80.

24.8.3 Usage Notes on MTU Operation by Event Signal Reception from the ELC

The following notes on usage apply when the MTU is used in event link operation.

(1) Start Counting

If the event specified in the ELSRn register occurs during a cycle of writing to a CSTn bit in the TSTRA register, writing to the CSTn bit in the TSTRA register does not proceed because setting of the bit to 1 due to the event takes priority.

(2) Restart Counting (Clear Counter)

If the event specified in the ELSRn register occurs during a cycle of writing to the TCNT counter, writing to the TCNT counter does not proceed because clearing of the counter due to the event takes priority.
In addition, for MTU3 and MTU4 in complementary PWM mode, do not use the counter restarting by the ELC.

25. Port Output Enable 3 (POE3a)

This MCU incorporates a port output enable 3 (POE3a) which can be used to, under various conditions, disable output signals for the MTU. Every output signal is put in the high-impedance state when the output is disabled.
In this section, “PCLK” is used to refer to PCLKB.

25.1 Overview

Table 25.1 lists the specifications of the POE3, and Figure 25.1 shows a block diagram of the POE3.

Table 25.1 POE3 Specifications

Item	Description
Pin status while output is disabled	• High-impedance
Target pins for switching to high-impedance state	• MTU output pins MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D) MTU3 pins (MTIOC3B, MTIOC3D) MTU4 pins (MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D) MTU6 pins (MTIOC6B, MTIOC6D) MTU7 pins (MTIOC7A, MTIOC7B, MTIOC7C, MTIOC7D)
Generating conditions of request for switching to high-impedance state	• Input signal detection: Detection of the POE0#, POE4#, POE8#, POE10#, and POE11# signal level. • Simultaneous conduction between output pins: A match (simultaneous conduction) between the output signal levels at the active level over one or more cycles on the following combination of pins • The SPOER register setting • Detection that the main clock oscillator had stopped oscillating
Function	• Each of the POE0#, POE4#, POE8#, POE10#, and POE11# input pins can be set for falling edge, PCLK/8 × 16, PCLK/16 × 16, or PCLK/128 × 16 low-level sampling. • The outputs of the target pins can be in the high-impedance state by falling-edge or low-level sampling of the POE0#, POE4#, POE8#, POE10#, or POE11# pin. • The outputs of the target pins can be in the high-impedance state when oscillation stop is detected by the oscillation stop detection function of the clock generator. • The MTU complementary PWM outputs can be in the high-impedance state when output levels of the MTU complementary PWM output pins are compared and simultaneous active-level output continues for one cycle or more. • The outputs of the target pins can be in the high-impedance state by modifying the settings of the POE3 registers. • Interrupts can be generated by input-level sampling or output-level comparison results.

	MTU Complementary PWM Output Pins
1	MTIOC3B and MTIOC3D
2	MTIOC4A and MTIOC4C
3	MTIOC4B and MTIOC4D
4	MTIOC6B and MTIOC6D
5	MTIOC7A and MTIOC7C
6	MTIOC7B and MTIOC7D

The POE3 has input-level detection circuits, pin selection circuits, output-level comparison circuits, and a high-impedance request/interrupt request generating circuit as shown in Figure 25.1.

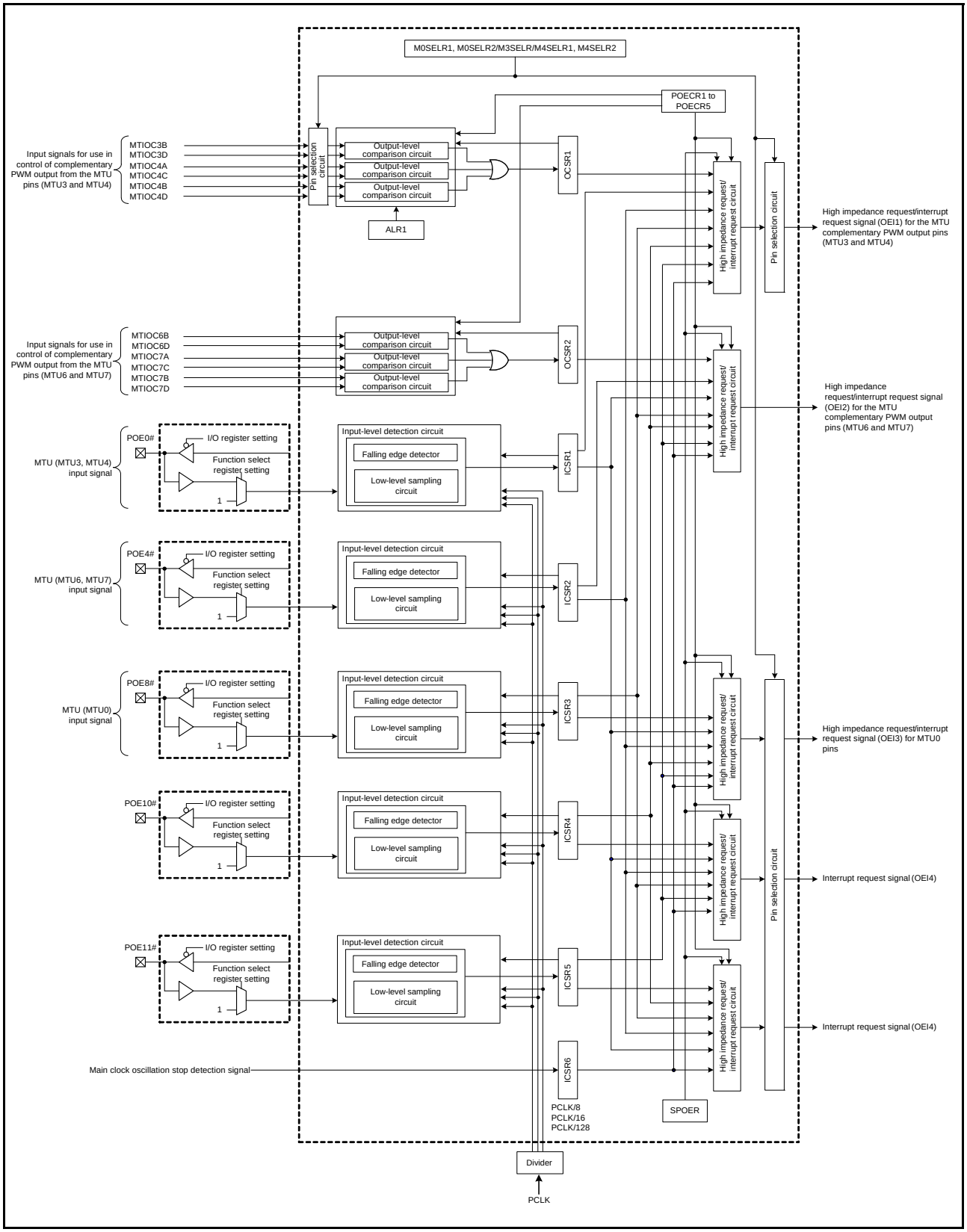


Figure 25.1 POE3 Block Diagram

Table 25.2 shows I/O pins to be used by the POE3.

Table 25.2 POE3 I/O Pins

Pin Name	I/O	Description
POE0#	Input	Request signal to put the outputs of the MTU complementary PWM output pins (MTU3, MTU4 pins) in the high-impedance state, and is also capable of controlling the other target pins by register settings.
POE4#	Input	Request signal to put the output of the MTU complementary PWM output pins (MTU6, MTU7 pins) in the high-impedance state, and is also capable of controlling the other target pins by register settings.
POE8#	Input	Request signal to put the output of the MTU0 pins in the high-impedance state, and is also capable of controlling the other target pins by register settings.
POE10#	Input	Is capable of controlling every target pins by register settings.
POE11#	Input	Is capable of controlling every target pins by register settings.

Table 25.3 shows output-level comparisons with pin combinations.

Table 25.3 Pin Combinations

Pin Combination	I/O	Description
MTIOC3B and MTIOC3D	Output	The MTU complementary PWM output pins (MTU3 and MTU4 pins) set in the M3SELR, M4SELR1, and M4SELR2 registers are in the high-impedance state when two pins of the set simultaneously output the active level (low level when the MTU.TOCR1A.OLSP bit is 0 or high level when the OLSP bit is 1 while the OLSEN bit in the ALR1 register is 0 and the MTU.TOCR1A.TOCS bit is 0, low level when the OLS1P, OLS1N, OLS2P, OLS2N, OLS3P, and OLS3N bits in the MTU.TOCR2A register are 0 or high level when these bits are 1 while the OLSEN bit in the ALR1 register is 0 and the MTU.TOCR1A.TOCS bit is 1, or low level when the OLSG0A, OLSG0B, OLSG1A, OLSG1B, OLSG2A, and OLSG2B bits in the ALR1 register are 0 and high level when these bits are 1 while the OLSEN bit in the ALR1 register is 1) for at least one cycle of the peripheral module clock (PCLK). Pin combinations for output comparison can be selected by registers of POE3.
MTIOC4A and MTIOC4C	Output	
MTIOC4B and MTIOC4D	Output	
MTIOC6B and MTIOC6D	Output	The MTU complementary PWM output pins (MTU6 and MTU7 pins) are in the high-impedance state when two pins of the set simultaneously output the active level (low level when the MTU.TOCR1B.OLSP bit is 0 or high level when the OLSP bit is 1 while the MTU.TOCR1B.TOCS bit is 0, or low level when the OLS1P, OLS1N, OLS2P, OLS2N, OLS3P, and OLS3N bits in the MTU.TOCR2B register are 0 or high level when these bits are 1 while the MTU.TOCR1B.TOCS bit is 1) for at least one cycle of the peripheral module clock (PCLK). Pin combinations for output comparison can be selected by registers of POE3.
MTIOC7A and MTIOC7C	Output	
MTIOC7B and MTIOC7D	Output	

25.2 Register Descriptions

The POE3 registers are initialized by a reset.

25.2.1 Input Level Control/Status Register 1 (ICSR1)

Address(es): POE3.ICSR1 0008 C4C0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	POE0F	—	—	—	PIE1	—	—	—	—	—	—	POE0M[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	POE0M[1:0]	POE0 Mode Select	b1 b0 0 0: Accepts a request on the falling edge of POE0# pin input. 0 1: Accepts a request when POE0# pin input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE0# pin input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE0# pin input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	PIE1	Port Interrupt Enable 1	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	POE0F	POE0 Flag	0: Indicates that a high-impedance request has not been input to the POE0# pin. 1: Indicates that a high-impedance request has been input to the POE0# pin.	R/(W)*2
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The ICSR1 register selects the POE0# pin input modes, controls the enable/disable of interrupts, and indicates status.

POE0M[1:0] Bits (POE0 Mode Select)

These bits select the input mode of the POE0# pin.

PIE1 Bit (Port Interrupt Enable 1)

This bit enables or disables interrupt requests when the POE0F flag is set to 1.

POE0F Flag (POE0 Flag)

This flag indicates that a high-impedance request has been input to the POE0# pin.

[Setting condition]

- When the input set by the POE0M[1:0] bits occurs at the POE0# pin

[Clearing condition]

- By writing 0 to the POE0F flag after reading POE0F = 1
  When low-level sampling is set by the POE0M[1:0] bits, the high level needs to be input to the POE0# pin to write 0 to this flag.
  For details, refer to section 25.3.7, Recover from High-Impedance State.

25.2.2 Input Level Control/Status Register 2 (ICSR2)

Address(es): POE3.ICSR2 0008 C4C4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	POE4F	—	—	—	PIE2	—	—	—	—	—	—	POE4M[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	POE4M[1:0]	POE4 Mode Select	b1 b0 0 0: Accepts a request on the falling edge of POE4# pin input. 0 1: Accepts a request when POE4# pin input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE4# pin input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE4# pin input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	PIE2	Port Interrupt Enable 2	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	POE4F	POE4 Flag	0: Indicates that a high-impedance request has not been input to the POE4# pin. 1: Indicates that a high-impedance request has been input to the POE4# pin.	R/(W)*2
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The ICSR2 register selects the POE4# pin input mode, controls the enable/disable of interrupts, and indicates status.

POE4M[1:0] Bits (POE4 Mode Select)

These bits select the input mode of the POE4# pin.

PIE2 Bit (Port Interrupt Enable 2)

This bit enables or disables interrupt requests when the POE4F flag is set to 1.

POE4F Flag (POE4 Flag)

This flag indicates that a high-impedance request has been input to the POE4# pin.

[Setting condition]

- When the input set by POE4M[1:0] occurs at the POE4# pin

[Clearing condition]

- By writing 0 to POE4F after reading POE4F = 1
  When low-level sampling is set by the POE4M[1:0] bits, the high level needs to be input to the POE4# pin to write 0 to this flag.
  For details, refer to section 25.3.7, Recover from High-Impedance State.

25.2.3 Input Level Control/Status Register 3 (ICSR3)

Address(es): POE3.ICSR3 0008 C4C8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	POE8F	—	—	POE8E	PIE3	—	—	—	—	—	—	POE8M[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	POE8M[1:0]	POE8 Mode Select	b1 b0 0 0: Accepts a request on the falling edge of POE8# pin input. 0 1: Accepts a request when POE8# pin input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE8# pin input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE8# pin input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	PIE3	Port Interrupt Enable 3	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	POE8E	POE8 High-Impedance Enable	0: Does not put the output in the high-impedance state by POE8# signal. 1: Put the output in the high-impedance state by POE8# signal.	R/W*1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	POE8F	POE8 Flag	0: Indicates that a high-impedance request has not been input to the POE8# pin. 1: Indicates that a high-impedance request has been input to the POE8# pin.	R/(W)*2
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The ICSR3 register selects the POE8# pin input mode, controls the enable/disable of interrupts, and indicates status.

POE8M[1:0] Bits (POE8 Mode Select)

These bits select the input mode of the POE8# pin.

PIE3 Bit (Port Interrupt Enable 3)

This bit enables or disables interrupt requests when the POE8F flag is set to 1.

POE8E Bit (POE8 High-Impedance Enable)

This bit specifies whether to put the output of the corresponding pin in the high-impedance state when the POE8F flag is set to 1.

POE8F Flag (POE8 Flag)

This flag indicates that a high-impedance request has been input to the POE8# pin.

[Setting condition]

- When the input set by the POE8M[1:0] bits occurs at the POE8# pin

[Clearing condition]

- By writing 0 to the POE8F flag after reading POE8F = 1
  When low-level sampling is set by the POE8M[1:0] bits, the high level needs to be input to the POE8# pin to write 0 to this flag.

For details, refer to section 25.3.7, Recover from High-Impedance State.

25.2.4 Input Level Control/Status Register 4 (ICSR4)

Address(es): POE3.ICSR4 0008 C4D6h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	POE10F	—	—	POE10E	PIE4	—	—	—	—	—	—	POE10M[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	POE10M[1:0]	POE10 Mode Select	b1 b0 0 0: Accepts a request on the falling edge of POE10# pin input. 0 1: Accepts a request when POE10# pin input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE10# pin input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE10# pin input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	PIE4	Port Interrupt Enable 4	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	POE10E	POE10 High-Impedance Enable	0: Does not put the output in the high-impedance state by POE10# signal. 1: Put the output in the high-impedance state by POE10# signal.	R/W*1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	POE10F	POE10 Flag	0: Indicates that a high-impedance request has not been input to the POE10# pin. 1: Indicates that a high-impedance request has been input to the POE10# pin.	R/(W)*2
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The ICSR4 register selects the POE10# pin input mode, controls the enable/disable of interrupts, and indicates status.

POE10M[1:0] Bits (POE10 Mode Select)

These bits select the input mode of the POE10# pin.

PIE4 Bit (Port Interrupt Enable 4)

This bit enables or disables interrupt requests when the POE10F flag is set to 1.

POE10E Bit (POE10 High-Impedance Enable)

This bit specifies whether to put the output of the corresponding pin in the high-impedance state when the POE10F flag is set to 1.

POE10F Flag (POE10 Flag)

This flag indicates that a high-impedance request has been input to the POE10# pin.
[Setting condition]

- When the input set by the POE10M[1:0] bits occurs at the POE10# pin

[Clearing condition]

- By writing 0 to the POE10F flag after reading POE10F = 1
  When low-level sampling is set by the POE10M[1:0] bits, the high level needs to be input to the POE10# pin to write 0 to this flag.
  For details, refer to section 25.3.7, Recover from High-Impedance State.

25.2.5 Input Level Control/Status Register 5 (ICSR5)

Address(es): POE3.ICSR5 0008 C4D8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	POE11F	—	—	POE11E	PIE5	—	—	—	—	—	—	POE11M[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	POE11M[1:0]	POE11 Mode Select	b1 b0 0 0: Accepts a request on the falling edge of POE11# pin input. 0 1: Accepts a request when POE11# pin input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE11# pin input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE11# pin input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	PIE5	Port Interrupt Enable 5	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	POE11E	POE11 High-Impedance Enable	0: Does not put the output in the high-impedance state by POE11# signal. 1: Put the output in the high-impedance state by POE11# signal.	R/W*1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	POE11F	POE11 Flag	0: Indicates that a high-impedance request has not been input to the POE11# pin. 1: Indicates that a high-impedance request has been input to the POE11# pin.	R/(W)*2
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The ICSR5 register selects the POE11# pin input mode, controls the enable/disable of interrupts, and indicates status.

POE11M[1:0] Bits (POE11 Mode Select)

These bits select the input mode of the POE11# pin.

PIE5 Bit (Port Interrupt Enable 5)

This bit enables or disables interrupt requests when the POE11F flag is set to 1.

POE11E Bit (POE11 High-Impedance Enable)

This bit specifies whether to put the output of the corresponding pin in the high-impedance state when the POE11F flag is set to 1.

POE11F Flag (POE11 Flag)

This flag indicates that a high-impedance request has been input to the POE11# pin.

[Setting condition]

- When the input set by the POE11M[1:0] bits occurs at the POE11# pin

[Clearing condition]

- By writing 0 to the POE11F flag after reading POE11F = 1
  When low-level sampling is set by the POE11M[1:0] bits, the high level needs to be input to the POE11# pin to write 0 to this flag.
  For details, refer to section 25.3.7, Recover from High-Impedance State.

25.2.6 Input Level Control/Status Register 6 (ICSR6)

Address(es): POE3.ICSR6 0008 C4DCh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	OSTSTF	—	—	OSTSTE	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9	OSTSTE	Oscillation Stop High-Impedance Enable	0: Does not put the output in the high-impedance state when the oscillation stop is detected. 1: Put the output in the high-impedance state when the oscillation stop is detected.	R/W*1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	OSTSTF	Oscillation Stop Detection Flag	0: Indicates that a high-impedance request by oscillation stop has not been generated. 1: Indicates that a high-impedance request by oscillation stop has been generated.	R/W*2
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The ICSR6 register controls the oscillation stop high-impedance and indicates status.

OSTSTE Bit (Oscillation Stop High-Impedance Enable)

This bit specifies whether to put the output of the target pin in the high-impedance state when oscillation stop is detected.

OSTSTF Flag (Oscillation Stop Detection Flag)

This flag indicates that a high-impedance request by the oscillation stop has been generated.

When the main clock oscillation stops, this flag is set to 1. To clear this flag, wait for at least 10 cycles of PCLK after this flag becomes 1 and write 0 to this flag while the OSTDSR.OSTDF flag is 0. Writing 0 to this flag while the OSTDSR.OSTDF flag is 1 cannot clear this flag. After clearing this flag, confirm that the flag has actually been modified to 0.

[Setting condition]

- When oscillation stop is detected

[Clearing condition]

- By writing 0 to the OSTSTF flag after reading OSTSTF = 1

25.2.7 Output Level Control/Status Register 1 (OCSR1)

Address(es): POE3.OCSR1 0008 C4C2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OSF1	—	—	—	—	—	OCE1	OIE1	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	OIE1	Simultaneous Conduction Interrupt Enable 1	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	OCE1	Simultaneous Conduction High-Impedance Enable 1	0: Does not put the outputs in the high-impedance state when they simultaneously go to an active level. 1: Put the outputs in the high-impedance state when they simultaneously go to an active level.	R/W*1
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	OSF1	Simultaneous Conduction Flag 1	0: Indicates that outputs have not simultaneously become an active level. 1: Indicates that outputs have simultaneously become an active level.	R/(W)*2

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The OCSR1 register controls the enable/disable of output-level comparison and interrupts, and indicates status.

OIE1 Bit (Simultaneous Conduction Interrupt Enable 1)

This bit enables or disables interrupt requests when the OSF1 flag is set to 1.

OCE1 Bit (Simultaneous Conduction High-Impedance Enable 1)

This bit specifies whether to put the output of the target pin in the high-impedance state when the OSF1 flag is set to 1.

OSF1 Flag (Simultaneous Conduction Flag 1)

This flag indicates that at least one of the three pairs of two-phase output pins for MTU complementary PWM output (MTU3 and MTU4) has simultaneously become at the active level. If the high-impedance control for the corresponding pins is not enabled, this flag does not become 1.
For setting the active level, refer to section 25.2.9, Active Level Setting Register 1 (ALR1).
[Setting condition]

- When the MTIOC3B and MTIOC3D pins simultaneously go to the active level*1 for at least one cycle of PCLK while the POECR2.MTU3BDZE bit is 1.
- When the MTIOC4A and MTIOC4C pins simultaneously go to the active level*1 for at least one cycle of PCLK while the POECR2.MTU4ACZE bit is 1.
- When the MTIOC4B and MTIOC4D pins simultaneously go to the active level*1 for at least one cycle of PCLK while the POECR2.MTU4BDZE bit is 1.

Note 1. The setting condition is judged only by the level of the pin regardless the setting of the MPC.PmnPFS register.

[Clearing condition]

- By writing 0 to the OSF1 flag after reading OSF1 = 1

To write 0 to this flag, the inactive level needs to be output from MTU complementary PWM output pins. For details,

refer to section 25.3.7, Recover from High-Impedance State.

25.2.8 Output Level Control/Status Register 2 (OCSR2)

Address(es): POE3.OCSR2 0008 C4C6h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OSF2	—	—	—	—	—	OCE2	OIE2	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	OIE2	Simultaneous Conduction Interrupt Enable 2	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	OCE2	Simultaneous Conduction High-Impedance Enable 2	0: Does not put the outputs in the high-impedance state when they simultaneously go to an active level. 1: Put the outputs in the high-impedance state when they simultaneously go to an active level.	R/W*1
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	OSF2	Simultaneous Conduction Flag 2	0: Indicates that outputs have not simultaneously become an active level. 1: Indicates that outputs have simultaneously become an active level.	R/(W) *2

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The OCSR2 register controls the enable/disable of output-level comparison and interrupts, and indicates status.

OIE2 Bit (Simultaneous Conduction Interrupt Enable 2)

This bit enables or disables interrupt requests when the OSF2 flag is set to 1.

OCE2 Bit (Simultaneous Conduction High-Impedance Enable 2)

This bit specifies whether to put the output of the target pin in the high-impedance state when the OSF2 flag is set to 1.

OSF2 Flag (Simultaneous Conduction Flag 2)

This flag indicates that at least one of the three pairs of two-phase output pins for MTU complementary PWM output (MTU6 and MTU7) has simultaneously become an active level. If the high-impedance control for the corresponding pins is not enabled, this flag does not become 1.
For setting the active level, refer to section 24, Multi-Function Timer Pulse Unit 3 (MTU3a).
[Setting condition]

- When the MTIOC6B and MTIOC6D pins simultaneously go to the active level*1 for at least one cycle of PCLK while the POECR2.MTU6BDZE bit is 1.
- When the MTIOC7A and MTIOC7C pins simultaneously go to the active level*1 for at least one cycle of PCLK while the POECR2.MTU7ACZE bit is 1.
- When the MTIOC7B and MTIOC7D pins simultaneously go to the active level*1 for at least one cycle of PCLK while the POECR2.MTU7BDZE bit is 1.

Note 1. The setting condition is judged only by the level of the pin regardless the setting of the MPC.PmnPFS register.

[Clearing condition]

- By writing 0 to the OSF2 flag after reading OSF2 = 1

To write 0 to this flag, the inactive level needs to be output from MTU complementary PWM output pins. For details, refer to section 25.3.7, Recover from High-Impedance State.

25.2.9 Active Level Setting Register 1 (ALR1)

Address(es): POE3.ALR1 0008 C4DAh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	OLSEN	—	OLSG2B	OLSG2A	OLSG1B	OLSG1A	OLSG0B	OLSG0A
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OLSG0A	MTIOC3B Pin Active Level Setting	0: Active low 1: Active high	R/W*1
b1	OLSG0B	MTIOC3D Pin Active Level Setting	0: Active low 1: Active high	R/W*1
b2	OLSG1A	MTIOC4A Pin Active Level Setting	0: Active low 1: Active high	R/W*1
b3	OLSG1B	MTIOC4C Pin Active Level Setting	0: Active low 1: Active high	R/W*1
b4	OLSG2A	MTIOC4B Pin Active Level Setting	0: Active low 1: Active high	R/W*1
b5	OLSG2B	MTIOC4D Pin Active Level Setting	0: Active low 1: Active high	R/W*1
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	OLSEN	Active Level Setting Enable	0: Disabled 1: Enabled	R/W*1
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The ALR1 register specifies the active levels of the MTU outputs for detection of simultaneous conduction of those outputs as reflected in the OCSR1 register.

OLSG0A Bit (MTIOC3B Pin Active Level Setting)

This bit sets the active level of the MTIOC3B output. Specifically, setting the OLSG0A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG0B Bit (MTIOC3D Pin Active Level Setting)

This bit sets the active level of the MTIOC3D output. Specifically, setting the OLSG0B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG1A Bit (MTIOC4A Pin Active Level Setting)

This bit sets the active level of the MTIOC4A output. Specifically, setting the OLSG1A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG1B Bit (MTIOC4C Pin Active Level Setting)

This bit sets the active level of the MTIOC4C output. Specifically, setting the OLSG1B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG2A Bit (MTIOC4B Pin Active Level Setting)

This bit sets the active level of the MTIOC4B output. Specifically, setting the OLSG2A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSG2B Bit (MTIOC4D Pin Active Level Setting)

This bit sets the active level of the MTIOC4D output. Specifically, setting the OLSG2B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of simultaneous conduction.

OLSEN Bit (Active Level Setting Enable)

This bit enables or disables of the active-level settings in the OLSGnm bits (n = 0 to 2; m = A, B). Clearing the OLSEN bit to 0 disables the OLSGnm bits, in which case the active levels of the MTU output are determined by the MTU.TOCR1j and MTU.TOCR2j registers (j = A, B). Setting the OLSEN bit to 1 enables the OLSGnm bits, in which case the active levels of the MTU output are as selected by the OLSGnm bits in this register.

25.2.10 Software Port Output Enable Register (SPOER)

Address(es): POE3.SPOER 0008 C4CAh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	MTUCH0HIZ	MTUCH67HIZ	MTUCH34HIZ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MTUCH34HIZ	MTU3 and MTU4 Pin High-Impedance Enable	0: Does not put the outputs in the high-impedance state. 1: Put the outputs in the high-impedance state.	R/W
b1	MTUCH67HIZ	MTU6 and MTU7 Pin High-Impedance Enable	0: Does not put the output in the high-impedance state. 1: Put the output in the high-impedance state.	R/W
b2	MTUCH0HIZ	MTU0 Pin High-Impedance Enable	0: Does not put the outputs in the high-impedance state. 1: Put the outputs in the high-impedance state.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SPOER register is used to put the outputs of the corresponding pins in the high-impedance state.

MTUCH34HIZ Bit (MTU3 and MTU4 Pin High-Impedance Enable)

This bit specifies whether to put the outputs of the MTU complementary PWM output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D) in the high-impedance state.

[Setting condition]

- By writing 1 to the MTUCH34HIZ bit

[Clearing conditions]

- Reset
- By writing 0 to the MTUCH34HIZ bit after reading MTUCH34HIZ = 1

MTUCH67HIZ Bit (MTU6 and MTU7 Pin High-Impedance Enable)

This bit specifies whether to put the outputs of the MTU complementary PWM output pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7B, MTIOC7C, and MTIOC7D) in the high-impedance state.

[Setting condition]

- By writing 1 to the MTUCH67HIZ bit

[Clearing conditions]

- Reset
- By writing 0 to the MTUCH67HIZ bit after reading MTUCH67HIZ = 1

MTUCH0HIZ Bit (MTU0 Pin High-Impedance Enable)

This bit specifies whether to put the outputs of the MTU0 pins in the high-impedance state.

[Setting condition]

- By writing 1 to the MTUCH0HIZ bit

[Clearing conditions]

- Reset
- By writing 0 to the MTUCH0HIZ bit after reading MTUCH0HIZ = 1

25.2.11 Port Output Enable Control Register 1 (POECR1)

Address(es): POE3.POECR1 0008 C4CBh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	MTU0DZE	MTU0CZE	MTU0BZE	MTU0AZE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MTU0AZE	MTIOC0A Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b1	MTU0BZE	MTIOC0B Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b2	MTU0CZE	MTIOC0C Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b3	MTU0DZE	MTIOC0D Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECR1 register controls high-impedance state of the MTU0 pins.

MTU0AZE Bit (MTIOC0A Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC0A output to the high-impedance state when any of the ICSR3.POE8F flag, SPOER.MTUCH0HIZ bit, and ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR5 register, the ICSRn.POEmF flag (n = 1, 2, 4, 5; m = 0, 4, 10, 11), is set to 1.

MTU0BZE Bit (MTIOC0B Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC0B output to the high-impedance state when any of the ICSR3.POE8F flag, SPOER.MTUCH0HIZ bit, and ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR5 register, the ICSRn.POEmF flag (n = 1, 2, 4, 5; m = 0, 4, 10, 11), is set to 1.

MTU0CZE Bit (MTIOC0C Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC0C output to the high-impedance state when any of the ICSR3.POE8F flag, SPOER.MTUCH0HIZ bit, and ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR5 register, the ICSRn.POEmF flag (n = 1, 2, 4, 5; m = 0, 4, 10, 11), is set to 1.

MTU0DZE Bit (MTIOC0D Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC0D output to the high-impedance state when any of the ICSR3.POE8F flag, SPOER.MTUCH0HIZ bit, and ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR5 register, the ICSRn.POEmF flag (n = 1, 2, 4, 5; m = 0, 4, 10, 11), is set to 1.

25.2.12 Port Output Enable Control Register 2 (POECR2)

Address(es): POE3.POECR2 0008 C4CCh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	MTU3BDZE	MTU4ACZE	MTU4BDZE	—	—	—	—	—	MTU6BDZE	MTU7ACZE	MTU7BDZE
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	MTU7BDZE	MTIOC7B/MTIOC7D Pin High-Impedance Enable*2	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W*1
b1	MTU7ACZE	MTIOC7A/MTIOC7C Pin High-Impedance Enable*2	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W*1
b2	MTU6BDZE	MTIOC6B/MTIOC6D Pin High-Impedance Enable*2	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W*1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	MTU4BDZE	MTIOC4B/MTIOC4D Pin High-Impedance Enable	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W*1
b9	MTU4ACZE	MTIOC4A/MTIOC4C Pin High-Impedance Enable	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W*1
b10	MTU3BDZE	MTIOC3B/MTIOC3D Pin High-Impedance Enable	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W*1
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.
Note 2. Set this bit to 0 when MTU6 or MTU7 is not used.

The POECR2 register controls high-impedance state of the MTU complementary PWM output pins (MTU3, MTU4, MTU6, and MTU7 pins).

MTU7BDZE Bit (MTIOC7B/MTIOC7D Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC7B output and MTIOC7D output to the high-impedance state when at least one of the OCSR2.OSF2 flag, ICSR2.POE4F flag, SPOER.MTUCH67HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR4 register, the ICSRn.POEmF flag (n = 1, 3 to 5; m = 0, 8, 10, 11), is set to 1.

MTU7ACZE Bit (MTIOC7A/MTIOC7C Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC7A output and MTIOC7C output to the high-impedance state when at least one of the OCSR2.OSF2 flag, ICSR2.POE4F flag, SPOER.MTUCH67HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR4 register, the ICSRn.POEmF flag (n = 1, 3 to 5; m = 0, 8, 10, 11), is set to 1.

MTU6BDZE Bit (MTIOC6B/MTIOC6D Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC6B output and MTIOC6D output to the high-impedance state when at least one of the OCSR2.OSF2 flag, ICSR2.POE4F flag, SPOER.MTUCH67HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR4 register, the ICSRn.POEmF flag (n = 1, 3 to 5; m = 0, 8, 10, 11), is set to 1.

MTU4BDZE Bit (MTIOC4B/MTIOC4D Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC4B output and MTIOC4D output to the high-impedance state when at least one of the OCSR1.OSF1 flag, ICSR1.POE0F flag, SPOER.MTUCH34HIZ bit, ICSR6.OSTSTF flag (when the

OSTSTE bit is 1), or, as additionally specified in the POECR4 register, the ICSRn.POEmF flag (n = 2 to 5; m = 4, 8, 10, 11), is set to 1.

MTU4ACZE Bit (MTIOC4A/MTIOC4C Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC4A output and MTIOC4C output to the high-impedance state when at least one of the OCSR1.OSF1 flag, ICSR1.POE0F flag, SPOER.MTUCH34HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR4 register, the ICSRn.POEmF flag (n = 2 to 5; m = 4, 8, 10, 11), is set to 1.

MTU3BDZE Bit (MTIOC3B/MTIOC3D Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC3B output and MTIOC3D output to the high-impedance state when at least one of the OCSR1.OSF1 flag, ICSR1.POE0F flag, SPOER.MTUCH34HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR4 register, the ICSRn.POEmF flag (n = 2 to 5; m = 4, 8, 10, 11), is set to 1.

25.2.13 Port Output Enable Control Register 4 (POECR4)

Address(es): POE3.POECR4 0008 C4D0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	IC5ADDMT67ZE	IC4ADDMT67ZE	IC3ADDMT67ZE	—	IC1ADDMT67ZE	—	—	—	IC5ADDMT34ZE	IC4ADDMT34ZE	IC3ADDMT34ZE	IC2ADDMT34ZE	—	—
Value after reset:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b2	IC2ADDMT34ZE	MTU3 and MTU4 High-Impedance Condition POE4F Add	0: Does not add the flag to the conditions to put the output in the high-impedance state. 1: Adds the flag to the conditions to put the output in the high-impedance state.	R/W*1
b3	IC3ADDMT34ZE	MTU3 and MTU4 High-Impedance Condition POE8F Add	0: Does not add the flag to the conditions to put the output in the high-impedance state. 1: Adds the flag to the conditions to put the output in the high-impedance state.	R/W*1
b4	IC4ADDMT34ZE	MTU3 and MTU4 High-Impedance Condition POE10F Add	0: Does not add the flag to the conditions to put the output in the high-impedance state. 1: Adds the flag to the conditions to put the output in the high-impedance state.	R/W*1
b5	IC5ADDMT34ZE	MTU3 and MTU4 High-Impedance Condition POE11F Add	0: Does not add the flag to the conditions to put the output in the high-impedance state. 1: Adds the flag to the conditions to put the output in the high-impedance state.	R/W*1
b8 to b6	—	Reserved	These bits are read as 0. The write value should be 0	R/W
b9	IC1ADDMT67ZE	MTU6 and MTU7 High-Impedance Condition POE0F Add	0: Does not add the flag to the conditions to put the output in the high-impedance state. 1: Adds the flag to the conditions to put the output in the high-impedance state.	R/W*1
b10	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b11	IC3ADDMT67ZE	MTU6 and MTU7 High-Impedance Condition POE8F Add	0: Does not add the flag to the conditions to put the output in the high-impedance state. 1: Adds the flag to the conditions to put the output in the high-impedance state.	R/W*1
b12	IC4ADDMT67ZE	MTU6 and MTU7 High-Impedance Condition POE10F Add	0: Does not add the flag to the conditions to put the output in the high-impedance state. 1: Adds the flag to the conditions to put the output in the high-impedance state.	R/W*1
b13	IC5ADDMT67ZE	MTU6 and MTU7 High-Impedance Condition POE11F Add	0: Does not add the flag to the conditions to put the output in the high-impedance state. 1: Adds the flag to the conditions to put the output in the high-impedance state.	R/W*1
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECR4 register is used to extend the control conditions to put the output of the MTU complementary PWM output pins (MTU3, MTU4, MTU6, and MTU7) in the high-impedance state.

IC2ADDMT34ZE Bit (MTU3 and MTU4 High-Impedance Condition POE4F Add)

Adds the ICSR2.POE4F flag to the high-impedance control conditions for the MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

IC3ADDMT34ZE Bit (MTU3 and MTU4 High-Impedance Condition POE8F Add)

Adds the ICSR3.POE8F flag to the high-impedance control conditions for the MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

IC4ADDMT34ZE Bit (MTU3 and MTU4 High-Impedance Condition POE10F Add)

Adds the ICSR4.POE10F flag to the high-impedance control conditions for the MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

IC5ADDMT34ZE Bit (MTU3 and MTU4 High-Impedance Condition POE11F Add)

Adds the ICSR5.POE11F flag to the high-impedance control conditions for the MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

IC1ADDMT67ZE Bit (MTU6 and MTU7 High-Impedance Condition POE0F Add)

Adds the ICSR1.POE0F flag to the high-impedance control conditions for the MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

IC3ADDMT67ZE Bit (MTU6 and MTU7 High-Impedance Condition POE8F Add)

Adds the ICSR3.POE8F flag to the high-impedance control conditions for the MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

IC4ADDMT67ZE Bit (MTU6 and MTU7 High-Impedance Condition POE10F Add)

Adds the ICSR4.POE10F flag to the high-impedance control conditions for the MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

IC5ADDMT67ZE Bit (MTU6 and MTU7 High-Impedance Condition POE11F Add)

Adds the ICSR5.POE11F flag to the high-impedance control conditions for the MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

25.2.14 Port Output Enable Control Register 5 (POECR5)

Address(es): POE3.POECR5 0008 C4D2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	IC5ADDMT0ZE	IC4ADDMT0ZE	—	IC2ADDMT0ZE	IC1ADDMT0ZE	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	IC1ADDMT0ZE	MTU0 High-Impedance Condition POE0F Add	0: Does not add the flag to the conditions to put the output in the high-impedance state. 1: Adds the flag to the conditions to put the output in the high-impedance state.	R/W*1
b2	IC2ADDMT0ZE	MTU0 High-Impedance Condition POE4F Add	0: Does not add the flag to the conditions to put the output in the high-impedance state. 1: Adds the flag to the conditions to put the output in the high-impedance state.	R/W*1
b3	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b4	IC4ADDMT0ZE	MTU0 High-Impedance Condition POE10F Add	0: Does not add the flag to the conditions to put the output in the high-impedance state. 1: Adds the flag to the conditions to put the output in the high-impedance state.	R/W*1
b5	IC5ADDMT0ZE	MTU0 High-Impedance Condition POE11F Add	0: Does not add the flag to the conditions to put the output in the high-impedance state. 1: Adds the flag to the conditions to put the output in the high-impedance state.	R/W*1
b15 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECR5 register is used to extend the control conditions to put the output of the MTU0 pins in the high-impedance state.

IC1ADDMT0ZE Bit (MTU0 High-Impedance Condition POE0F Add)

Adds the ICSR1.POE0F flag to the high-impedance control conditions for the MTU0 pin (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

IC2ADDMT0ZE Bit (MTU0 High-Impedance Condition POE4F Add)

Adds the ICSR2.POE4F flag to the high-impedance control conditions for the MTU0 pin (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

IC4ADDMT0ZE Bit (MTU0 High-Impedance Condition POE10F Add)

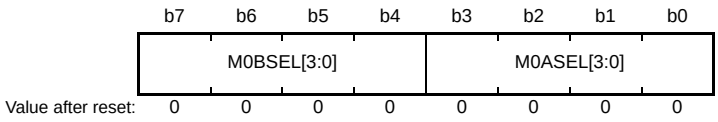
Adds the ICSR4.POE10F flag to the high-impedance control conditions for the MTU0 pin (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

IC5ADDMT0ZE Bit (MTU0 High-Impedance Condition POE11F Add)

Adds the ICSR5.POE11F flag to the high-impedance control conditions for the MTU0 pin (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

25.2.15 MTU0 Pin Select Register 1 (M0SELR1)

Address(es): POE3.M0SELR1 0008 C4E4h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	M0ASEL[3:0]	MTU0-A (MTIOC0A) Pin Select*2	b3 b0 0000: Controls the high-impedance state of P34 on the assumption that the MTIOC0A pin is assigned to P34. 0010: Controls the high-impedance state of PB3 on the assumption that the MTIOC0A pin is assigned to PB3.*3 Settings other than above are prohibited.	R/W*1
b7 to b4	M0BSEL[3:0]	MTU0-B (MTIOC0B) Pin Select	b7 b4 0000: Controls the high-impedance state of P13 on the assumption that the MTIOC0B pin is assigned to P13. 0001: Controls the high-impedance state of P15 on the assumption that the MTIOC0B pin is assigned to P15.*3 0010: Controls the high-impedance state of PA1 on the assumption that the MTIOC0B pin is assigned to PA1. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.
Note 2. These bits are only effective in products with at least 100 pins. Set these bits to 0000b in products with fewer pins.
Note 3. This selection is only usable in products with at least 100 pins.

The M0SELR1 register is an 8-bit readable/writable register that selects the MTU0-A/B pins as targets for high-impedance control.

M0ASEL[3:0] Bits (MTU0-A (MTIOC0A) Pin Select)

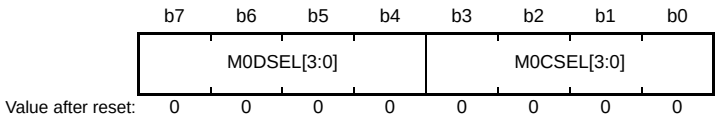
These bits select the target MTIOC0A pin for high-impedance control.

M0BSEL[3:0] Bits (MTU0-B (MTIOC0B) Pin Select)

These bits select the target MTIOC0B pin for high-impedance control.

25.2.16 MTU0 Pin Select Register 2 (M0SELR2)

Address(es): POE3.M0SELR2 0008 C4E5h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	M0CSEL[3:0]	MTU0-C (MTIOC0C) Pin Select*2	b3 b0 0000: Controls the high-impedance state of P32 on the assumption that the MTIOC0C pin is assigned to P32. 0010: Controls the high-impedance state of PB1 on the assumption that the MTIOC0C pin is assigned to PB1. Settings other than above are prohibited.	R/W*1
b7 to b4	M0DSEL[3:0]	MTU0-D (MTIOC0D) Pin Select*2	b7 b4 0000: Controls the high-impedance state of P33 on the assumption that the MTIOC0D pin is assigned to P33. 0010: Controls the high-impedance state of PA3 on the assumption that the MTIOC0D pin is assigned to PA3. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.
Note 2. These bits are only effective in products with at least 100 pins. Set these bits to 0000b in products with fewer pins.

The M0SELR2 register is an 8-bit readable/writable register that selects the MTU0-C/D pins as targets for high-impedance control.

M0CSEL[3:0] Bits (MTU0-C (MTIOC0C) Pin Select)

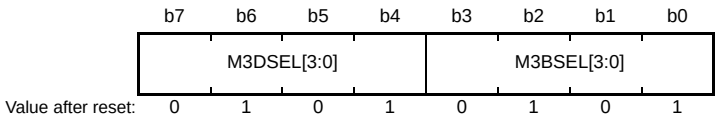
These bits select the target MTIOC0C pin for high-impedance control.

M0DSEL[3:0] Bits (MTU0-D (MTIOC0D) Pin Select)

These bits select the target MTIOC0D pin for high-impedance control.

25.2.17 MTU3 Pin Select Register (M3SELR)

Address(es): POE3.M3SELR 0008 C4E6h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	M3BSEL[3:0]	MTU3-B (MTIOC3B) Pin Select	b3 b0 0000: Controls the high-impedance state of PE1 on the assumption that the MTIOC3B pin is assigned to PE1.*2 0001: Controls the high-impedance state of P22 on the assumption that the MTIOC3B pin is assigned to P22.*3 0010: Controls the high-impedance state of P80 on the assumption that the MTIOC3B pin is assigned to P80.*4 0011: Controls the high-impedance state of PC5 on the assumption that the MTIOC3B pin is assigned to PC5. 0100: Controls the high-impedance state of PB7 on the assumption that the MTIOC3B pin is assigned to PB7. 0101: Controls the high-impedance state of P17 on the assumption that the MTIOC3B pin is assigned to P17. Settings other than above are prohibited.	R/W*1
b7 to b4	M3DSEL[3:0]	MTU3-D (MTIOC3D) Pin Select	b7 b4 0000: Controls the high-impedance state of PE0 on the assumption that the MTIOC3D pin is assigned to PE0.*2 0001: Controls the high-impedance state of P23 on the assumption that the MTIOC3D pin is assigned to P23.*3 0010: Controls the high-impedance state of PC4 on the assumption that the MTIOC3D pin is assigned to PC4. 0011: Controls the high-impedance state of P81 on the assumption that the MTIOC3D pin is assigned to P81.*4 0100: Controls the high-impedance state of PB6 on the assumption that the MTIOC3D pin is assigned to PB6. 0101: Controls the high-impedance state of P16 on the assumption that the MTIOC3D pin is assigned to P16. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.
Note 2. This selection is only usable in products with at least 64 pins.
Note 3. This selection is only usable in products with at least 100 pins.
Note 4. This selection is only usable in products with at least 144 pins.

The M3SELR register is an 8-bit readable/writable register that selects the MTU3-B/D pins as targets for high-impedance control.

M3BSEL[3:0] Bits (MTU3-B (MTIOC3B) Pin Select)

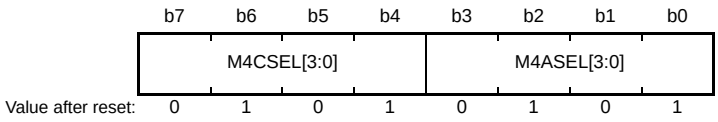
These bits select the target MTIOC3B pin for high-impedance control.

M3DSEL[3:0] Bits (MTU3-D (MTIOC3D) Pin Select)

These bits select the target MTIOC3D pin for high-impedance control.

25.2.18 MTU4 Pin Select Register 1 (M4SELR1)

Address(es): POE3.M4SELR1 0008 C4E7h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	M4ASEL[3:0]	MTU4-A (MTIOC4A) Pin Select*2	b3 b0 0000: Controls the high-impedance state of PE2 on the assumption that the MTIOC4A pin is assigned to PE2.*3 0001: Controls the high-impedance state of P21 on the assumption that the MTIOC4A pin is assigned to P21.*4 0010: Controls the high-impedance state of PB3 on the assumption that the MTIOC4A pin is assigned to PB3.*4 0011: Controls the high-impedance state of P82 on the assumption that the MTIOC4A pin is assigned to P82.*5 0100: Controls the high-impedance state of PA0 on the assumption that the MTIOC4A pin is assigned to PA0.*4 0101: Controls the high-impedance state of P24 on the assumption that the MTIOC4A pin is assigned to P24.*4 Settings other than above are prohibited.	R/W*1
b7 to b4	M4CSEL[3:0]	MTU4-C (MTIOC4C) Pin Select*6	b7 b4 0000: Controls the high-impedance state of PE5 on the assumption that the MTIOC4C pin is assigned to PE5.*4 0001: Controls the high-impedance state of P87 on the assumption that the MTIOC4C pin is assigned to P87.*5 0010: Controls the high-impedance state of PB1 on the assumption that the MTIOC4C pin is assigned to PB1.*4 0011: Controls the high-impedance state of P83 on the assumption that the MTIOC4C pin is assigned to P83.*5 0100: Controls the high-impedance state of PE1 on the assumption that the MTIOC4C pin is assigned to PE1.*3 0101: Controls the high-impedance state of P25 on the assumption that the MTIOC4C pin is assigned to P25.*4 Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.
Note 2. These bits are only effective in products with at least 100 pins. Set these bits to 0000b in products with 64 pins. Leave these bits set to 0101b in products with 48 pins.
Note 3. This selection is only usable in products with at least 64 pins.
Note 4. This selection is only usable in products with at least 100 pins.
Note 5. This selection is only usable in products with at least 144 pins.
Note 6. These bits are only effective in products with at least 100 pins. Set these bits to 0100b in products with 64 pins. Leave these bits set to 0101b in products with 48 pins.

The M4SELR1 register is an 8-bit readable/writable register that selects the MTU4-A/C pins as targets for high-impedance control.

M4ASEL[3:0] Bits (MTU4-A (MTIOC4A) Pin Select)

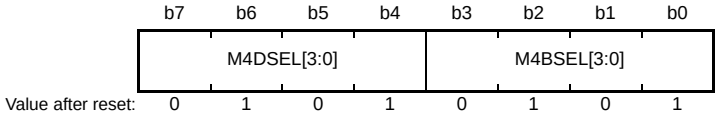
These bits select the target MTIOC4A pin for high-impedance control.

M4CSEL[3:0] Bits (MTU4-C (MTIOC4C) Pin Select)

These bits select the target MTIOC4C pin for high-impedance control.

25.2.19 MTU4 Pin Select Register 2 (M4SELR2)

Address(es): POE3.M4SELR2 0008 C4E8h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	M4BSEL[3:0]	MTU4-B (MTIOC4B) Pin Select	b3 b0 0000: Controls the high-impedance state of PE3 on the assumption that the MTIOC4B pin is assigned to PE3.*2 0001: Controls the high-impedance state of P17 on the assumption that the MTIOC4B pin is assigned to P17. 0010: Controls the high-impedance state of P54 on the assumption that the MTIOC4B pin is assigned to P54.*2 0011: Controls the high-impedance state of PC2 on the assumption that the MTIOC4B pin is assigned to PC2.*2 0100: Controls the high-impedance state of PD1 on the assumption that the MTIOC4B pin is assigned to PD1.*2 0101: Controls the high-impedance state of P30 on the assumption that the MTIOC4B pin is assigned to P30. Settings other than above are prohibited.	R/W*1
b7 to b4	M4DSEL[3:0]	MTU4-D (MTIOC4D) Pin Select	b7 b4 0000: Controls the high-impedance state of PE4 on the assumption that the MTIOC4D pin is assigned to PE4.*2 0001: Controls the high-impedance state of P86 on the assumption that the MTIOC4D pin is assigned to P86.*3 0010: Controls the high-impedance state of P55 on the assumption that the MTIOC4D pin is assigned to P55.*2 0011: Controls the high-impedance state of PC3 on the assumption that the MTIOC4D pin is assigned to PC3.*2 0100: Controls the high-impedance state of PD2 on the assumption that the MTIOC4D pin is assigned to PD2. 0101: Controls the high-impedance state of P31 on the assumption that the MTIOC4D pin is assigned to P31. Settings other than above are prohibited.	R/W*1

Note 1. Can be modified only once after a reset.
Note 2. This selection is only usable in products with at least 100 pins.
Note 3. This selection is only usable in products with at least 144 pins.

The M4SELR2 register is an 8-bit readable/writable register that selects the MTU4-B/D pins as targets for high-impedance control.

M4BSEL[3:0] Bits (MTU4-B (MTIOC4B) Pin Select)

These bits select the target MTIOC4A pin for high-impedance control.

M4DSEL[3:0] Bits (MTU4-D (MTIOC4D) Pin Select)

These bits select the target MTIOC4C pin for high-impedance control.

25.3 Operation

The following shows the target pins and conditions for high-impedance control.

(1) MTU3 pins (MTIOC3B, MTIOC3D)

When one of the following conditions is satisfied while the POECR2.MTU3BDZE bit is 1, the pins become high-impedance.

- Operation for detection of the POE0# input level
  When the ICSR1.POE0F flag becomes 1.
- Operation for comparison of the output levels on the MTIOC3B and MTIOC3D pins
  When the OCSR1.OSF1 flag becomes 1 while the OCSR1.OCE1 bit is 1.
- SPOER setting
  When the SPOER.MTUCH34HIZ bit is set to 1.
- Conditions added by POECR4
  When the ICSR2.POE4F flag becomes 1 while the POECR4.IC2ADDMT34ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR4.IC3ADDMT34ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR4.IC4ADDMT34ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR4.IC5ADDMT34ZE bit and the ICSR5.POE11E bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(2) MTU4 pins (MTIOC4A, MTIOC4C)

When one of the following conditions is satisfied while the POECR2.MTU4ACZE bit is 1, the pins become high-impedance.

- Operation for detection of the POE0# input level
  When the ICSR1.POE0F flag becomes 1.
- Operation for comparison of the output levels on the MTIOC4A and MTIOC4C pins
  When the OCSR1.OSF1 flag becomes 1 while the OCSR1.OCE1 bit is 1.
- SPOER setting
  When the SPOER.MTUCH34HIZ bit is set to 1.
- Conditions added by POECR4
  When the ICSR2.POE4F flag becomes 1 while the POECR4.IC2ADDMT34ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR4.IC3ADDMT34ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR4.IC4ADDMT34ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR4.IC5ADDMT34ZE bit and the ICSR5.POE11E bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(3) MTU4 pins (MTIOC4B, MTIOC4D)

When one of the following conditions is satisfied while the POECR2.MTU4BDZE bit is 1, the pins become high-impedance.

- Operation for detection of the POE0# input level
  When the ICSR1.POE0F flag becomes 1.
- Operation for comparison of the output levels on the MTIOC4B and MTIOC4D pins

When the OCSR1.OSF1 flag becomes 1 while the OCSR1.OCE1 bit is 1.

- SPOER setting
  When the SPOER.MTUCH34HIZ bit is set to 1.
- Conditions added by POECR4
  When the ICSR2.POE4F flag becomes 1 while the POECR4.IC2ADDMT34ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR4.IC3ADDMT34ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR4.IC4ADDMT34ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR4.IC5ADDMT34ZE bit and the ICSR5.POE11E bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(4) MTU6 pins (MTIOC6B, MTIOC6D)

When one of the following conditions is satisfied while the POECR2.MTU6BDZE bit is 1, the pins become high-impedance.

- Operation for detection of the POE4# input level
  When the ICSR2.POE4F flag becomes 1.
- Operation for comparison of the output levels on the MTIOC6B and MTIOC6D pins
  When the OCSR2.OSF2 flag becomes 1 while the OCSR2.OCE2 bit is 1.
- SPOER setting
  When the SPOER.MTUCH67HIZ bit is set to 1.
- Conditions added by POECR4
  When the ICSR1.POE0F flag becomes 1 while the POECR4.IC1ADDMT67ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR4.IC3ADDMT67ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR4.IC4ADDMT67ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR4.IC5ADDMT67ZE bit and the ICSR5.POE11E bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(5) MTU7 pins (MTIOC7A, MTIOC7C)

When one of the following conditions is satisfied while the POECR2.MTU7ACZE bit is 1, the pins become high-impedance.

- Operation for detection of the POE4# input level
  When the ICSR2.POE4F flag becomes 1.
- Operation for comparison of the output levels on the MTIOC7A and MTIOC7C pins
  When the OCSR2.OSF2 flag becomes 1 while the OCSR2.OCE2 bit is 1.
- SPOER setting
  When the SPOER.MTUCH67HIZ bit is set to 1.
- Conditions added by POECR4
  When the ICSR1.POE0F flag becomes 1 while the POECR4.IC1ADDMT67ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR4.IC3ADDMT67ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR4.IC4ADDMT67ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR4.IC5ADDMT67ZE bit and the ICSR5.POE11E bit are 1.

- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(6) MTU7 pins (MTIOC7B, MTIOC7D)

When one of the following conditions is satisfied while the POECR2.MTU7BDZE bit is 1, the pins become high-impedance.

- Operation for detection of the POE4# input level
  When the ICSR2.POE4F flag becomes 1.
- Operation for comparison of the output levels on the MTIOC7B and MTIOC7D pins
  When the OCSR2.OSF2 flag becomes 1 while the OCSR2.OCE2 bit is 1.
- SPOER setting
  When the SPOER.MTUCH67HIZ bit is set to 1.
- Conditions added by POECR4
  When the ICSR1.POE0F flag becomes 1 while the POECR4.IC1ADDMT67ZE bit is 1.
  When the ICSR3.POE8F flag becomes 1 while the POECR4.IC3ADDMT67ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR4.IC4ADDMT67ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR4.IC5ADDMT67ZE bit and the ICSR5.POE11E bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(7) MTU0 pin (MTIOC0A)

When one of the following conditions is satisfied while the POECR1.MTU0AZE bit is 1, the pin becomes high-impedance.

- Operation for detection of the POE8# input level
  When the ICSR3.POE8F flag becomes 1 while the ICSR3.POE8E bit is 1.
- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR5.IC2ADDMT0ZE bit is 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR5.IC5ADDMT0ZE bit and the ICSR5.POE11E bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(8) MTU0 pin (MTIOC0B)

When one of the following conditions is satisfied while the POECR1.MTU0BZE bit is 1, the pin becomes high-impedance.

- Operation for detection of the POE8# input level
  When the ICSR3.POE8F flag becomes 1 while the ICSR3.POE8E bit is 1.
- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.

When the ICSR2.POE4F flag becomes 1 while the POECR5.IC2ADDMT0ZE bit is 1.
When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
When the ICSR5.POE11F flag becomes 1 while the POECR5.IC5ADDMT0ZE bit and the ICSR5.POE11E bit are 1.

- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(9) MTU0 pin (MTIOC0C)

When one of the following conditions is satisfied while the POECR1.MTU0CZE bit is 1, the pin becomes high-impedance.

- Operation for detection of the POE8# input level
  When the ICSR3.POE8F flag becomes 1 while the ICSR3.POE8E bit is 1.
- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR5.IC2ADDMT0ZE bit is 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR5.IC5ADDMT0ZE bit and the ICSR5.POE11E bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(10) MTU0 pin (MTIOC0D)

When one of the following conditions is satisfied while the POECR1.MTU0DZE bit is 1, the pin becomes high-impedance.

- Operation for detection of the POE8# input level
  When the ICSR3.POE8F flag becomes 1 while the ICSR3.POE8E bit is 1.
- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR2.POE4F flag becomes 1 while the POECR5.IC2ADDMT0ZE bit is 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
  When the ICSR5.POE11F flag becomes 1 while the POECR5.IC5ADDMT0ZE bit and the ICSR5.POE11E bit are 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

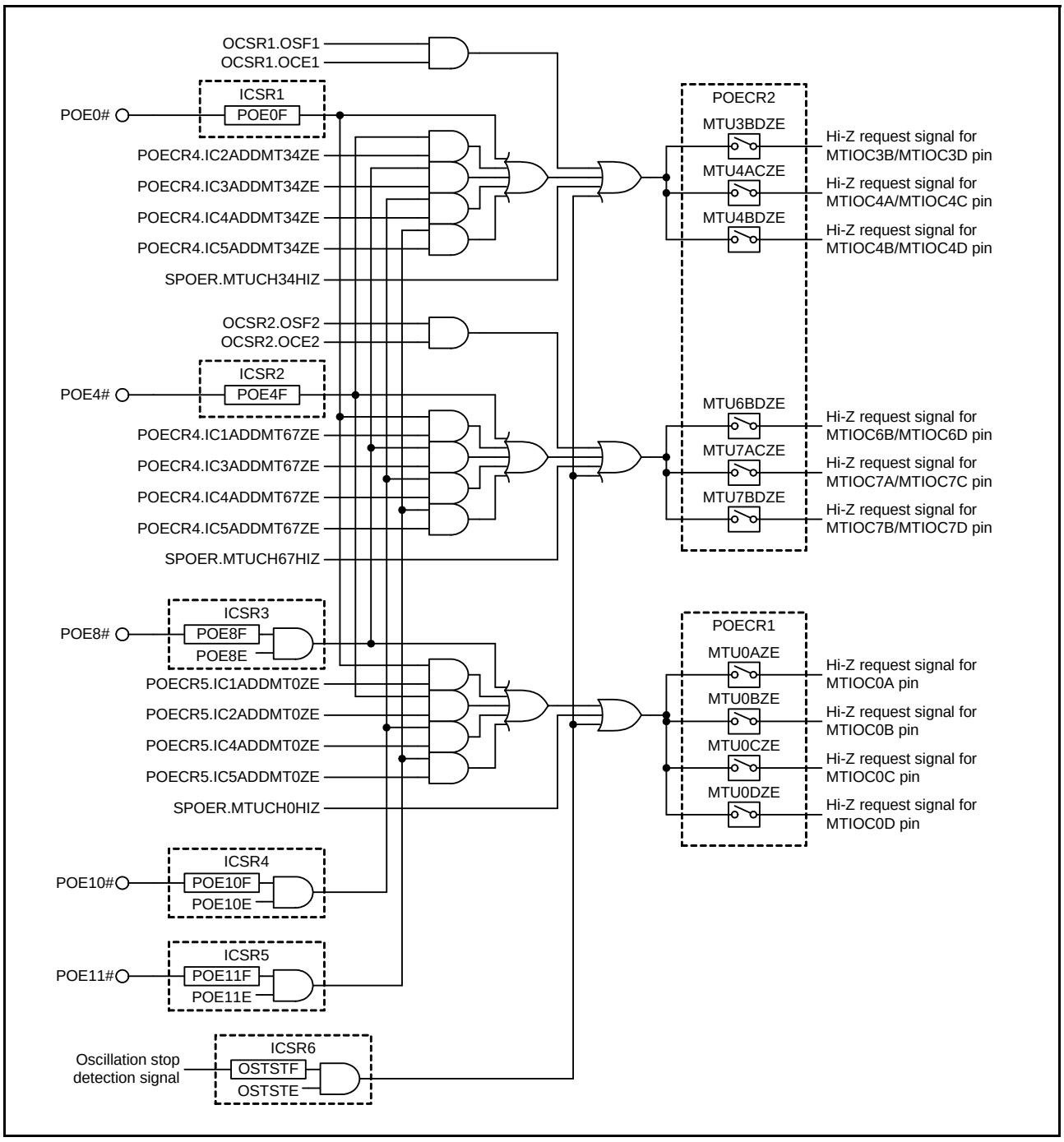


Figure 25.2 Target Pins and Conditions for High-Impedance Control

25.3.1 MTU Pin Selection

In this MCU, the pin functions for MTU are respectively multiplexed with multiple sets of port pins. The target pins for high-impedance control can be selected by the pin select register in POE3 (M0SELR1, M0SELR2, M3SELR, M4SELR1, or M4SELR2 register). Table 25.4 shows the correspondence between MTU pins and select registers. Note that settings for pins to be used as MTU must be separately made in the registers of the multi-function pin controller (MPC). Take care so that there are no differences between the pins selected in the POE3 registers and the pins selected in the MPC registers.

Table 25.4 Correspondence between MTU Pins

MTU Pin Functions	Corresponding Ports	Select Registers
MTIOC0A	P34	M0SELR1
	PB3*1	
MTIOC0B	P13	
	P15*1	
	PA1	
MTIOC0C*1	P32*1	M0SELR2
	PB1*1	
MTIOC0D*1	P33*1	
	PA3*1	
MTIOC3B	PE1*2	M3SELR
	P22*1	
	P80*3	
	PC5	
	PB7	
	P17	
MTIOC3D	PE0*2	
	P23*1	
	PC4	
	P81*3	
	PB6	
	P16	
MTIOC4A*2	PE2*2	M4SELR1
	P21*1	
	PB3*1	
	P82*3	
	PA0*1	
	P24*1	
MTIOC4C*2	PE5*1	
	P87*3	
	PB1*1	
	P83*3	
	PE1*2	
	P25*1	

MTU Pin Functions	Corresponding Ports	Select Registers
MTIOC4B	PE3*1	M4SELR2
	P17	
	P54*1	
	PC2*1	
	PD1*1	
	P30	
MTIOC4D	PE4*1	
	P86*3	
	P55*1	
	PC3*1	
	PD2	
	P31	
MTIOC6B*1	PA5*1	—
MTIOC6D*1	PA0*1	
MTIOC7A	PA2	—
MTIOC7C*3	P67*3	
MTIOC7B	PA1	
MTIOC7D*3	P66*3	

Note 1. This is only present in products with at least 100 pins.
Note 2. This is only present in products with at least 64 pins.
Note 3. This is only present in products with at least 144 pins.

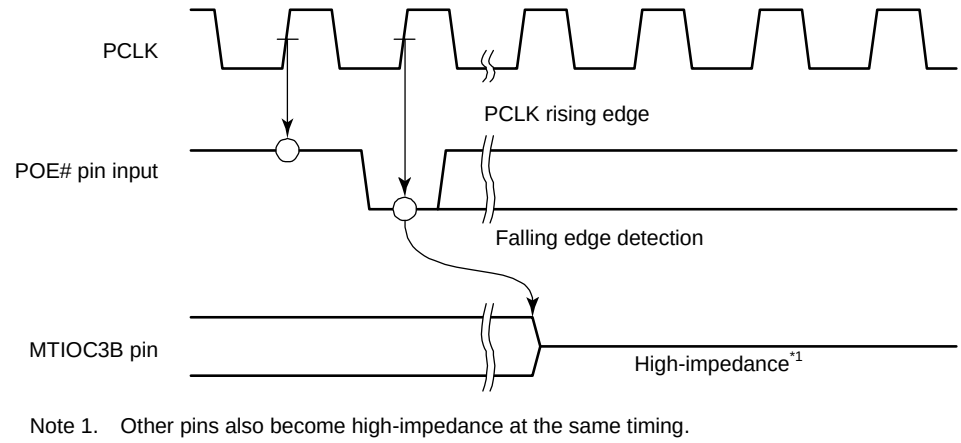
25.3.2 Input-Level Detection Operation

If the input conditions set by ICSR1 to ICSR5 occur on the POE0#, POE4#, POE8#, POE10#, and POE11# pins, the outputs of the MTU complementary PWM output pins (MTU3 and MTU4 or MTU6 and MTU7) and MTU0 pins are in the high-impedance state. Note however, that these outputs are still in the high-impedance state even when the MTU functions are not selected for the pins.

(1) Falling Edge Detection

When a change from a high to low level is input to the POE0#, POE4#, POE8#, POE10#, and POE11# pins, the outputs of the pins multiplexed with MTU complementary PWM output pins and MTU0 pins are in the high-impedance state. The falling edge is detected after the level is sampled with PCLK. Input a low level for at least one PCLK clock to the POE0#, POE4#, POE8#, POE10#, and POE11# pins.

Figure 25.3 shows a sample timing after the level changes in input to the POE0#, POE4#, POE8#, POE10#, and POE11# pins until the respective pins become high-impedance.

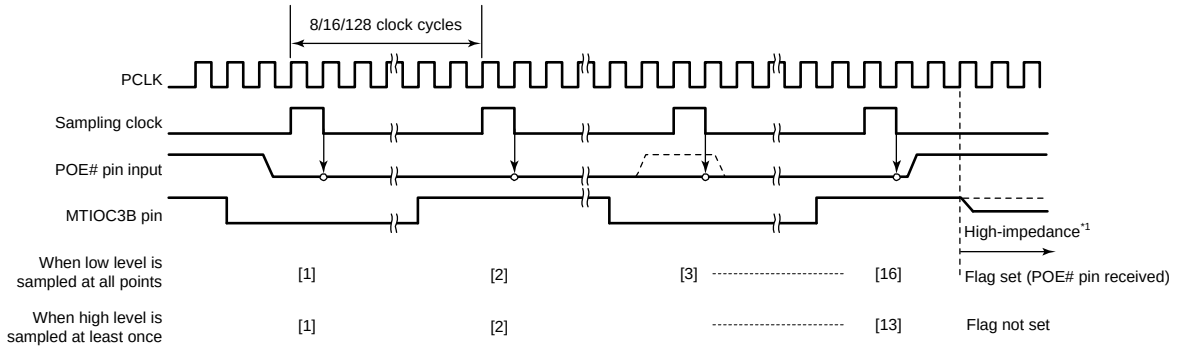


Note 1. Other pins also become high-impedance at the same timing.

Figure 25.3 Operation when A Falling Edge Detection is Selected

(2) Low-Level Detection

Figure 25.4 shows an example of operation when a pin is placed in the high-impedance state in response to low-level detection. When 16 continuous low levels are sampled with the sampling clock selected by the ICSR1 to ICSR5 registers, the low level is recognized and the outputs of the MTU complementary PWM output pins and MTU0 pins are in the high-impedance state. If even one high level is detected during this interval, the low level is not recognized.
The timing when the outputs of the MTU complementary PWM output pins and MTU0 pins are in the high-impedance state after the sampling clock is input is the same in both falling-edge detection and in low-level detection.

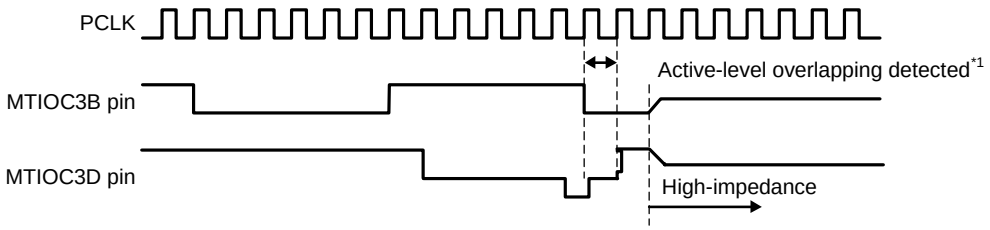


Note 1. Other pins also become high-impedance at the same timing.

Figure 25.4 Operation when A Low-Level Detection is Selected

25.3.3 Output-Level Compare Operation

Figure 25.5 shows an example of the output-level compare operation for the combination of MTIOC3B and MTIOC3D. The operation is the same for the other pin combinations.



Note 1. When the active level of both MTIOC3B and MTIOC3D pins is set to low.

Figure 25.5 Output-Level Compare Operation

25.3.4 High-Impedance Control Using Registers

The high-impedance request of the MTU pins (MTU0, MTU3, MTU4, MTU6, and MTU7) can be directly controlled by using the SPOER register.
For instance, setting the SPOER.MTUCH34HIZ bit to 1 switches the MTU3 and MTU4 pins specified by the POECR2 register to the high-impedance state.
The high-impedance request of other pins can also be controlled by setting the appropriate bits in the SPOER register.

25.3.5 High-Impedance Control through Detection of Oscillation Stop

When oscillation stop is detected by the oscillation stop detection function of the clock generator while the ICSR6.OSTSTE bit is 1, the MTU complementary PWM output pins specified by the POECR2 register and the MTU0 pins specified by the POECR1 register are switched to the high-impedance state.

25.3.6 Additional Functions for High-Impedance Control

High-impedance control conditions for the MTU complementary PWM output pins and MTU0 pins can be added by setting the POECR4 and POECR5 registers.
For instance, the settings listed below can be added as high-impedance control conditions for the MTU3 and MTU4 pins.

- Setting the POECR4.IC2ADDMT34ZE bit to 1 adds the input-level detection by the POE4# pin
- Setting the POECR4.IC3ADDMT34ZE bit to 1 and adds the input-level detection by the POE8# pin
- Setting the POECR4.IC4ADDMT34ZE bit to 1 and adds the input-level detection by the POE10# pin
- Setting the POECR4.IC5ADDMT34ZE bit to 1 and adds the input-level detection by the POE11# pin

The high-impedance control of other pins can also be controlled by setting the appropriate bits in the POECR4 and POECR5 registers.

25.3.7 Recover from High-Impedance State

The outputs which have been in the high-impedance state due to input-level detection can be recovered from the state either by returning them to their initial state with a reset, or by clearing all of the ICSR1.POE0F, ICSR2.POE4F, ICSR3.POE8F, ICSR4.POE10F, and ICSR5.POE11F flags. However, note that when low-level sampling is selected with the ICSR1.POE0M[1:0], ICSR2.POE4M[1:0], ICSR3.POE8M[1:0], ICSR4.POE10M[1:0], and ICSR5.POE11M[1:0] bits, just writing 0 to a flag is ignored (the flag is not set to 0); flags can be cleared by writing 0 to it only after a high level is input to the POE0#, POE4#, POE8#, POE10#, and POE11# pins and is detected.
The outputs which have been in the high-impedance state due to output-level detection can be recovered from the state either by returning them to their initial state with a reset, or by setting the OCSR1.OSF1 flag or the OCSR2.OSF2 flag to 0. However, note that just writing 0 to a flag is ignored (the flag is not set to 0); the flags can be cleared by writing 0 to it only after setting the inactive level to be output from the pin. In the MTU, the inactive level (initial output level) can be output by stopping the count operation.
The outputs which have been in the high-impedance state due to oscillation stop detection can be recovered from the state either by returning them to their initial state with a reset or by setting the SYSTEM.OSTDSR.OSTDF flag to 0 to set the ICSR6.OSTSTF flag to 0.

25.4 POE3 Setting Procedure

Figure 25.6 shows the procedure for setting the POE3. It illustrates an example of high-impedance control in response to comparison of the output levels on the MTU3 pins (MTIOC3B/MTIOC3D). In the figure, P22 is used as the MTIOC3B pin and P23 is used as the MTIOC3D pin.

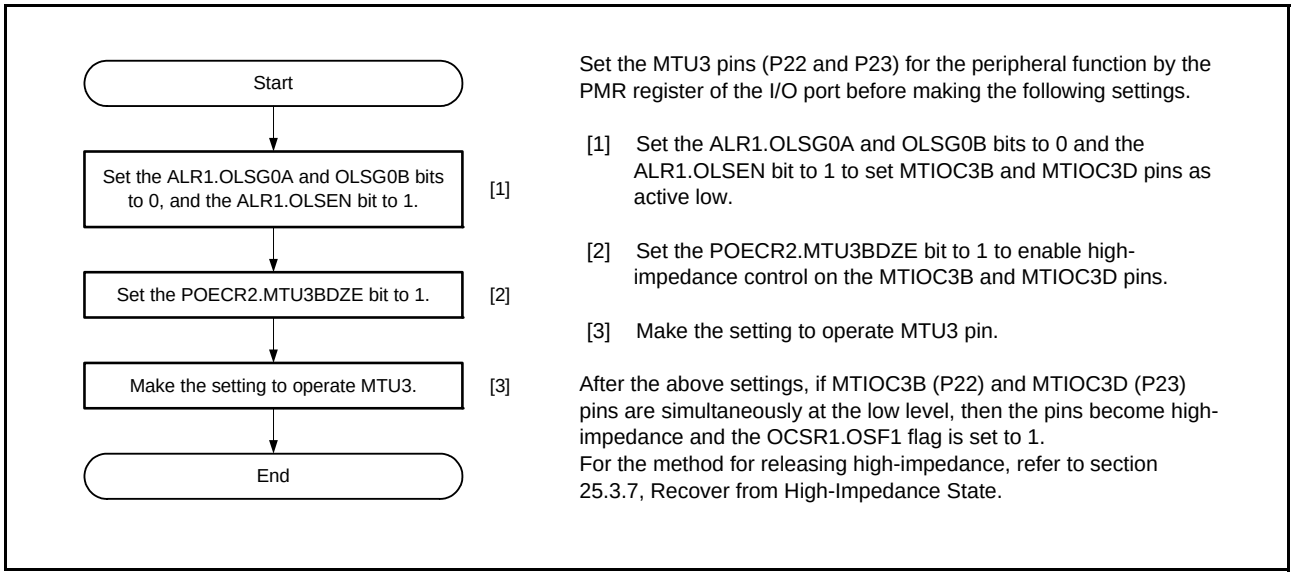


Figure 25.6 Procedure for Setting the POE3

25.5 Interrupts

The POE3 issues a request to generate an interrupt when the specified condition is satisfied during input-level detection or output-level comparison. Table 25.5 shows the interrupt sources and their conditions.

Table 25.5 Interrupt Sources and Conditions

Name	Interrupt Source	Interrupt Flag	Condition
OEI1	Output enable interrupt 1	POE0F OSF1	When the ICSR1.POE0F flag becomes 1 while the ICSR1.PIE1 bit is 1 or when the OCSR1.OSF1 flag becomes 1 while the OCSR1.OIE1 bit is 1
OEI2	Output enable interrupt 2	POE4F OSF2	When the ICSR2.POE4F flag becomes 1 while the ICSR2.PIE2 bit is 1 or when the OCSR2.OSF2 flag becomes 1 while the OCSR2.OIE2 bit is 1
OEI3	Output enable interrupt 3	POE8F	When the ICSR3.POE8F flag becomes 1 while the ICSR3.PIE3 bit is 1
OEI4	Output enable interrupt 4	POE10F POE11F	When the ICSR4.POE10F flag is set to 1 while the ICSR4.PIE4 bit is 1 or when the ICSR5.POE11F flag becomes 1 while the ICSR5.PIE5 bit is 1

25.6 Usage Notes

25.6.1 Transition to Low Power Consumption Mode

When the POE3 is used, do not make a transition to software standby mode or deep software standby mode. In these modes, the POE3 stops and thus the high-impedance control of pins cannot operate.

25.6.2 High-Impedance Control When the MTU is Not Selected

If high-impedance control for a pin having a multiplexed MTU pin function is enabled by setting the POECR1 and POECR2 registers and the high-impedance control condition is satisfied, the output is to be in the high-impedance state even if the MTU function is not selected for the pin on which it is multiplexed.
To avoid unintended high-impedance, ensure that there are no differences between the settings for MTU pin selection in the PmnPFS registers of the MPC and for MTU pin selection in the pin select register of the POE3.

25.6.3 When the POE3 is Not Used

The high-impedance control of some pins can be enabled using the POE3 after a reset. When the POE3 is not used, write 0 to the target bits in the POECR1 and POECR2 registers.

26. 16-Bit Timer Pulse Unit (TPUa)

This MCU has on-chip 16-bit timer pulse units (TPU) comprising six-channel 16-bit timers.
In this section, “PCLK” is used to refer to PCLKB.

26.1 Overview

Specifications of the TPU are listed in Table 26.1. Functions of TPU are listed in Table 26.2.
Figure 26.1 shows a block diagram of TPU.

Table 26.1 Specifications of TPU

Item	Description
Pulse input/output	Maximum 16
Count clocks	Seven or eight types are provided for each channel.
Settable operations	• Waveform output at compare match • Input capture function (noise filters can be set) • Counter clear operation • Simultaneous writing to multiple timer counters (TCNT) • Simultaneous clearing by compare match and input capture • Synchronous input/output for registers by counter synchronous operation • Maximum of 15-phase PWM output by combination with synchronous operation • Cascaded operation
TPU0 and TPU3	Buffer operation can be set.
TPU1, TPU2, TPU4, and TPU5	Phase counting mode can be set.
Interrupt sources	26 sources
Buffer operation	Automatic transfer of register data
Generation of trigger	Programmable pulse generator (PPG) output trigger can be generated.
	Conversion start trigger for the A/D converter can be generated.
Event linking (output)	Six types of event signal can be output to the ELC. • Compare match A (TPU0 to TPU3) • Compare match B (TPU0 to TPU3) • Compare match C (TPU0, TPU3) • Compare match D (TPU0, TPU3) • Overflow (TPU0 to TPU3) • Underflow (TPU1, TPU2)
Event linking (input)	Any of the three operations in response to event input is possible. • Starting counts (TPU0 to TPU3) • Restarting counts (TPU0 to TPU3) • Input capture operation (TPU0 to TPU3)
Low power consumption function	Module stop state can be set.

Table 26.2 TPU Functions (1/2)

Item		TPU0	TPU1	TPU2	TPU3	TPU4	TPU5
Count clocks		PCLK/1 PCLK/4 PCLK/16 PCLK/64 TCLKA TCLKB TCLKC TCLKD	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 TCLKA TCLKB	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/1024 TCLKA TCLKB TCLKC	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 PCLK/1024 PCLK/4096 TCLKA	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/1024 TCLKA TCLKC	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 TCLKA TCLKC TCLKD
External clocks for phase counting mode		Not possible	TCLKA TCLKB	TCLKC TCLKD	Not possible	TCLKC TCLKD	TCLKA TCLKB
Timer general registers		TGRA TGRB TGRC*1 TGRD*1	TGRA TGRB	TGRA TGRB	TGRA TGRB TGRC*1 TGRD*1	TGRA TGRB	TGRA TGRB
I/O pins		TIOCA0 TIOCB0 TIOCC0 TIOCD0	TIOCA1 TIOCB1	TIOCA2 TIOCB2	TIOCA3 TIOCB3 TIOCC3 TIOCD3	TIOCA4 TIOCB4	TIOCA5 TIOCB5
Counter clear function (y = A to D)		TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture
Compare match output	Low output	Possible	Possible	Possible	Possible	Possible	Possible
	High output	Possible	Possible	Possible	Possible	Possible	Possible
	Toggle output	Possible	Possible	Possible	Possible	Possible	Possible
Input capture function		Possible	Possible	Possible	Possible	Possible	Possible
Synchronous operation		Possible	Possible	Possible	Possible	Possible	Possible
PWM mode		Possible	Possible	Possible	Possible	Possible	Possible
Phase counting mode		Not possible	Possible	Possible	Not possible	Possible	Possible
Buffer operation		Possible	Not possible	Not possible	Possible	Not possible	Not possible
DTC activation (y = A to D)		TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture
DMAC activation		TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture
A/D conversion start trigger		TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	Not possible
PPG trigger		TGRA/TGRB compare match or input capture	TGRA/TGRB compare match or input capture	TGRA/TGRB compare match or input capture	TGRA/TGRB compare match or input capture	Not possible	Not possible
Interrupt sources		5 sources • Compare match or input capture 0A • Compare match or input capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Overflow	4 sources • Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	4 sources • Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow	5 sources • Compare match or input capture 3A • Compare match or input capture 3B • Compare match or input capture 3C • Compare match or input capture 3D • Overflow	4 sources • Compare match or input capture 4A • Compare match or input capture 4B • Overflow • Underflow	4 sources • Compare match or input capture 5A • Compare match or input capture 5B • Overflow • Underflow

Table 26.2 TPU Functions (2/2)

Item	TPU0	TPU1	TPU2	TPU3	TPU4	TPU5
Event linking (output)	5 sources • Compare match 0A • Compare match 0B • Compare match 0C • Compare match 0D • Overflow	4 sources • Compare match 1A • Compare match 1B • Overflow • Underflow	4 sources • Compare match 2A • Compare match 2B • Overflow • Underflow	5 sources • Compare match 3A • Compare match 3B • Compare match 3C • Compare match 3D • Overflow	Not possible	Not possible
Event linking (input)	• Starting counts • Restarting counts • Input capture operation (data is captured in TGRA)	• Starting counts • Restarting counts • Input capture operation (data is captured in TGRA)	• Starting counts • Restarting counts • Input capture operation (data is captured in TGRA)	• Starting counts • Restarting counts • Input capture operation (data is captured in TGRA)	Not possible	Not possible
Module stop setting*2	MSTPCRA.MSTPA13 bit					

Note 1. TGRC and TGRD can be set as a buffer register.
Note 2. For details, see section 11, Low Power Consumption.

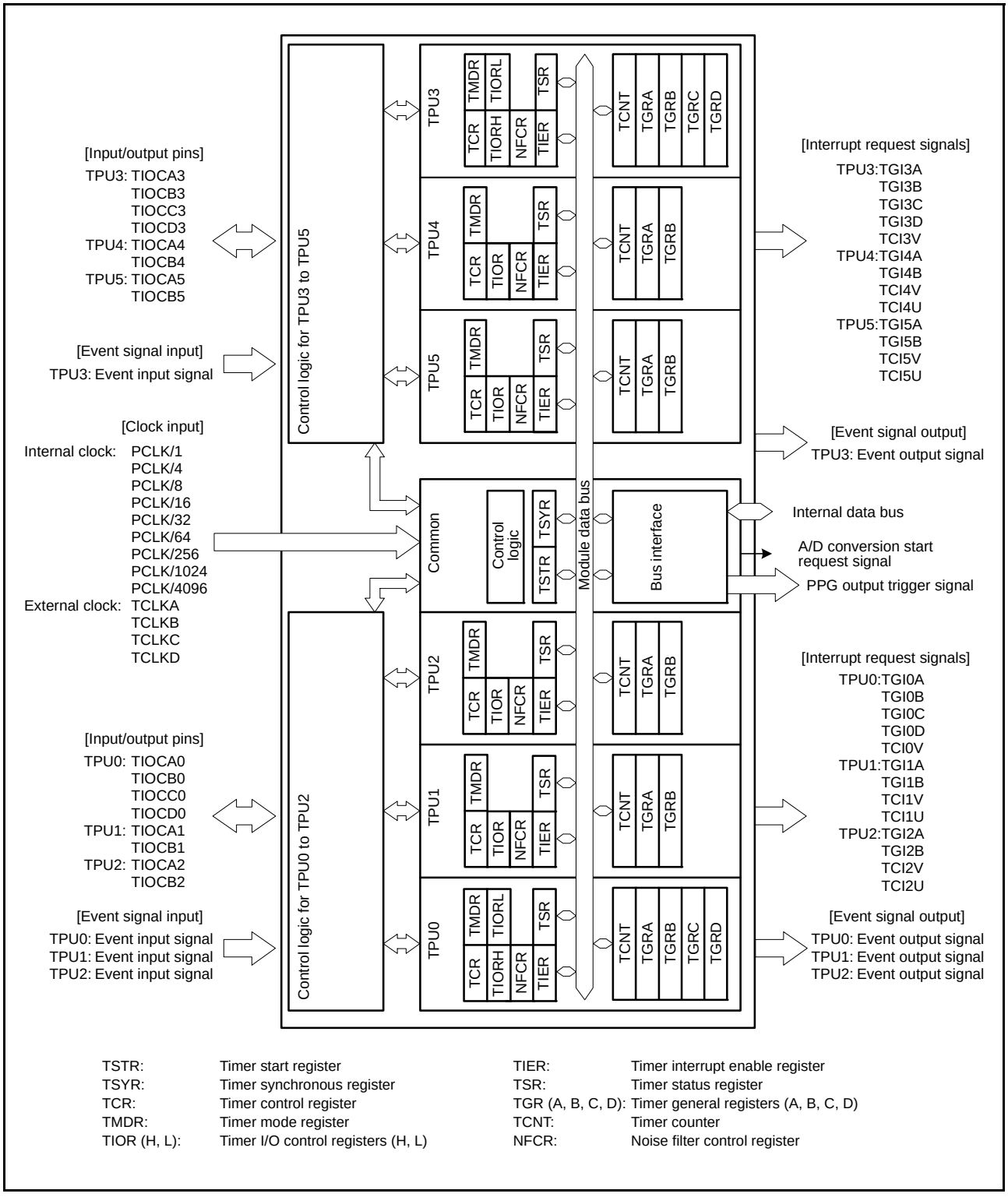


Figure 26.1 Block Diagram of TPU

Table 26.3 lists the input/output pins of the TPU.

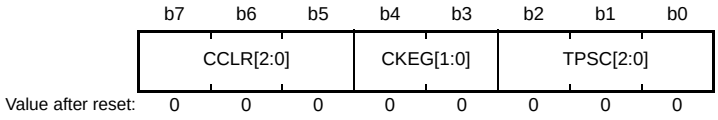
Table 26.3 Pin Configuration of TPU

Channel	Pin Name	I/O	Description
Common	TCLKA	Input	External clock A input pin (TPU1 and TPU5 phase counting mode A phase input)
	TCLKB	Input	External clock B input pin (TPU1 and TPU5 phase counting mode B phase input)
	TCLKC	Input	External clock C input pin (TPU2 and TPU4 phase counting mode A phase input)
	TCLKD	Input	External clock D input pin (TPU2 and TPU4 phase counting mode B phase input)
TPU0	TIOCA0	I/O	TPU0.TGRA input capture input/output compare output/PWM output pin
	TIOCB0	I/O	TPU0.TGRB input capture input/output compare output/PWM output pin
	TIOCC0	I/O	TPU0.TGRC input capture input/output compare output/PWM output pin
	TIOCD0	I/O	TPU0.TGRD input capture input/output compare output/PWM output pin
TPU1	TIOCA1	I/O	TPU1.TGRA input capture input/output compare output/PWM output pin
	TIOCB1	I/O	TPU1.TGRB input capture input/output compare output/PWM output pin
TPU2	TIOCA2	I/O	TPU2.TGRA input capture input/output compare output/PWM output pin
	TIOCB2	I/O	TPU2.TGRB input capture input/output compare output/PWM output pin
TPU3	TIOCA3	I/O	TPU3.TGRA input capture input/output compare output/PWM output pin
	TIOCB3	I/O	TPU3.TGRB input capture input/output compare output/PWM output pin
	TIOCC3	I/O	TPU3.TGRC input capture input/output compare output/PWM output pin
	TIOCD3	I/O	TPU3.TGRD input capture input/output compare output/PWM output pin
TPU4	TIOCA4	I/O	TPU4.TGRA input capture input/output compare output/PWM output pin
	TIOCB4	I/O	TPU4.TGRB input capture input/output compare output/PWM output pin
TPU5	TIOCA5	I/O	TPU5.TGRA input capture input/output compare output/PWM output pin
	TIOCB5	I/O	TPU5.TGRB input capture input/output compare output/PWM output pin

26.2 Register Descriptions

26.2.1 Timer Control Register (TCR)

Address(es): TPU0.TCR 0008 8110h, TPU1.TCR 0008 8120h, TPU2.TCR 0008 8130h, TPU3.TCR 0008 8140h, TPU4.TCR 0008 8150h, TPU5.TCR 0008 8160h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC[2:0]	Timer Prescaler Select	See Table 26.4 to Table 26.9.	R/W
b4, b3	CKEG[1:0]	Input Clock Edge Select	See Table 26.10.	R/W
b7 to b5	CCLR[2:0]	Counter Clear Source Select*1	See Table 26.11 and Table 26.12.	R/W

Note 1. Bit 7 is reserved in TPU1, TPU2, TPU4, and TPU5. These bits are read as 0. The write value should be 0.

TPUm.TCR settings should be made while TPUm.TCNT operation is stopped.

TPSC[2:0] Bits (Timer Prescaler Select)

These bits select the TCNT clock. The clock source can be selected independently for each channel.
To select the external clock as the clock source, set the bit in the port direction register (PDR) for the corresponding pin to 0 (input port), and set the bit in the port mode register (PMR) to 1 (uses the pin as an I/O port for peripheral functions). For details, see section 22, I/O Ports.

CKEG[1:0] Bits (Input Clock Edge Select)

These bits select the input clock edge.
When the internal clock is counted using both edges, the input clock period is halved (e.g. Both edges of PCLK/4 = PCLK/2 rising edge).
Internal clock edge selection is valid when the input clock is PCLK/4 or slower. This setting is ignored if the input clock is PCLK/1, or when overflow/underflow of another channel is selected.

Table 26.4 Bits TPSC[2:0] (TPU0)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU0	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on TCLKA pin input
	1	0	1	External clock: counts on TCLKB pin input
	1	1	0	External clock: counts on TCLKC pin input
	1	1	1	External clock: counts on TCLKD pin input

Table 26.5 Bits TPSC[2:0] (TPU1)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU1	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on TCLKA pin input
	1	0	1	External clock: counts on TCLKB pin input
	1	1	0	Internal clock: counts on PCLK/256
	1	1	1	Counts on TPU2.TCNT overflow/underflow

Note: This setting is invalid when TPU1 is in phase counting mode.

Table 26.6 Bits TPSC[2:0] (TPU2)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU2	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on TCLKA pin input
	1	0	1	External clock: counts on TCLKB pin input
	1	1	0	External clock: counts on TCLKC pin input
	1	1	1	Internal clock: counts on PCLK/1024

Note: This setting is invalid when TPU2 is in phase counting mode.

Table 26.7 Bits TPSC[2:0] (TPU3)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU3	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on TCLKA pin input
	1	0	1	Internal clock: counts on PCLK/1024
	1	1	0	Internal clock: counts on PCLK/256
	1	1	1	Internal clock: counts on PCLK/4096

Table 26.8 Bits TPSC[2:0] (TPU4)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU4	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on TCLKA pin input
	1	0	1	External clock: counts on TCLKC pin input
	1	1	0	Internal clock: counts on PCLK/1024
	1	1	1	Counts on TPU5.TCNT overflow/underflow

Note: This setting is invalid when TPU4 is in phase counting mode.

Table 26.9 Bits TPSC[2:0] (TPU5)

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU5	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on TCLKA pin input
	1	0	1	External clock: counts on TCLKC pin input
	1	1	0	Internal clock: counts on PCLK/256
	1	1	1	External clock: counts on TCLKD pin input

Note: This setting is invalid when TPU5 is in phase counting mode.

Table 26.10 Bits CKEG[1:0]

Bits CKEG[1:0]		Input Clock	
b4	b3	Internal Clock	External clock
0	0	Counted at falling edge	Counted at rising edge
0	1	Counted at rising edge	Counted at falling edge
1	0	Counted at both edges	Counted at both edges
1	1	Counted at both edges	Counted at both edges

Table 26.11 Bits CCLR[2:0] (TPU0, TPU3)

Channel	Bits CCLR[2:0]			Description
	b7	b6	b5	
TPU0, TPU3	0	0	0	TCNT clearing disabled
	0	0	1	TCNT cleared by TGRA compare match/input capture
	0	1	0	TCNT cleared by TGRB compare match/input capture
	0	1	1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*2
	1	0	0	TCNT clearing disabled
	1	0	1	TCNT cleared by TGRC compare match/input capture*1
	1	1	0	TCNT cleared by TGRD compare match/input capture*1
	1	1	1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*2

Note 1. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority, and compare match/input capture does not occur.
Note 2. Synchronous operation is selected by setting the TPUA.TSYR.SYNCj bit (j = 0, 3) to 1.

Table 26.12 Bits CCLR[2:0] (TPU1, TPU2, TPU4, TPU5)

Channel	Bits CCLR[2:0]			Description
	b7*1	b6	b5	
TPU1, TPU2, TPU4, TPU5	0	0	0	TCNT clearing disabled
	0	0	1	TCNT cleared by TGRA compare match/input capture
	0	1	0	TCNT cleared by TGRB compare match/input capture
	0	1	1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*2
	1	0	0	Setting prohibited
	1	0	1	Setting prohibited
	1	1	0	Setting prohibited
	1	1	1	Setting prohibited

Note 1. This bit is reserved in TPU1, TPU2, TPU4, and TPU5. This bit is read as 0. The write value should be 0.
Note 2. Synchronous operation is selected by setting the TPUA.TSYR.SYNCj bit (j = 1, 2, 4, 5) to 1.

26.2.2 Timer Mode Register (TMDR)

Address(es): TPU0.TMDR 0008 8111h, TPU1.TMDR 0008 8121h, TPU2.TMDR 0008 8131h, TPU3.TMDR 0008 8141h, TPU4.TMDR 0008 8151h, TPU5.TMDR 0008 8161h

	b7	b6	b5	b4	b3	b2	b1	b0
	ICSELD	ICSELB	BFB	BFA	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MD[3:0]	Mode Select	b3 b0 0 0 0 0: Normal operation 0 0 0 1: Setting prohibited 0 0 1 0: PWM mode 1 0 0 1 1: PWM mode 2 0 1 0 0: Phase counting mode 1*1 0 1 0 1: Phase counting mode 2*1 0 1 1 0: Phase counting mode 3*1 0 1 1 1: Phase counting mode 4*1 Settings other than above are prohibited.	R/W
b4	BFA	Buffer Operation A*2	0: TPUm.TGRA operates normally 1: TPUm.TGRA and TPUm.TGRC used together for buffer operation (m = 0, 3)	R/W
b5	BFB	Buffer Operation B*2	0: TPUm.TGRB operates normally 1: TPUm.TGRB and TPUm.TGRD used together for buffer operation (m = 0, 3)	R/W
b6	ICSELB	TGRB Input Capture Input Select	0: Input capture input source is TIOCBn pin 1: Input capture input source is TIOCAn pin (n = 0 to 5)	R/W
b7	ICSELD	TGRD Input Capture Input Select*2	0: Input capture input source is TIOCDn pin 1: Input capture input source is TIOCCn pin (n = 0, 3)	R/W

Note 1. Phase counting mode cannot be set for TPU0 and TPU3. A 0 should be written to bit 2 for them.
Note 2. These bits are reserved in TPU1, TPU2, TPU4, and TPU5. These bits are read as 0. The write value should be 0.

TPUm.TMDR settings should be made while TPUm.TCNT operation is stopped.

BFA Bit (Buffer Operation A)

Specifies whether TPUm.TGRA (m = 0, 3) is to normally operate, or TPUm.TGRA and TPUm.TGRC (m = 0, 3) are to be used together for buffer operation.
When TGRC is used as a buffer register, TGRC input capture/output compare is not generated.

BFB Bit (Buffer Operation B)

Specifies whether TPUm.TGRB (m = 0, 3) is to normally operate, or TPUm.TGRB and TPUm.TGRD (m = 0, 3) are to be used together for buffer operation.
When TGRD is used as a buffer register, TGRD input capture/output compare is not generated.

ICSELB Bit (TGRB Input Capture Input Select)

Selects the input capture input for TPUm.TGRB (m = 0 to 5).
This function allows measurement of high-level width and period of the input pulse on a TIOCAn input pin.

ICSELD Bit (TGRD Input Capture Input Select)

Selects the input capture input for TPUm.TGRD (m = 0, 3).
This function allows measurement of high-level width and period of the input pulse on a TIOCCn input pin.

26.2.3 Timer I/O Control Register (TIORH, TIORL, TIOR)

- TPU0.TIORH, TPU1.TIOR, TPU2.TIOR, TPU3.TIORH, TPU4.TIOR, TPU5.TIOR

Address(es): TPU0.TIORH 0008 8112h, TPU1.TIOR 0008 8122h, TPU2.TIOR 0008 8132h, TPU3.TIORH 0008 8142h, TPU4.TIOR 0008 8152h, TPU5.TIOR 0008 8162h

	b7	b6	b5	b4	b3	b2	b1	b0
	IOB[3:0]				IOA[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOA[3:0]	TGRA Control	See Table 26.13 to Table 26.18.*1	R/W
b7 to b4	IOB[3:0]	TGRB Control	See Table 26.13 to Table 26.18.*1	R/W

Note 1. If the IOn[3:0] bit (n = A, B) values are changed to output disabled (0000b or 0100b) during low/high/toggle output on compare match, the TIOCAn/TIOCBn pin (n = 0 to 5) is placed in high impedance state.

- TPU0.TIORL, TPU3.TIORL

Address(es): TPU0.TIORL 0008 8113h, TPU3.TIORL 0008 8143h

	b7	b6	b5	b4	b3	b2	b1	b0
	IOD[3:0]				IOC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOC[3:0]	TGRC Control	See Table 26.19 and Table 26.20.*1	R/W
b7 to b4	IOD[3:0]	TGRD Control	See Table 26.19 and Table 26.20.*1	R/W

Note 1. If the IOn[3:0] bit (n = C, D) values are changed to output disabled (0000b or 0100b) during low/high/toggle output on compare match, the TIOCCn/TIOCDn pin (n = 0, 3) is placed in high impedance state.

TPU has two TIORH registers, one for TPU0 and TPU3, and two TIORL registers, one for TPU0 and TPU3, and also has four TIOR registers, one for TPU1, TPU2, TPU4, and TPU5. Thus the TPU has eight timer I/O control registers in total.

TIORH, TIORL, and TIOR control registers TGRA, TGRB, TGRC, and TGRD.

Note that TIORH, TIORL, and TIOR are affected by the TMDR setting. For details, see Table 26.13 to Table 26.20.

The initial output specified by TIORH, TIORL, and TIOR is valid when the counter is stopped (the TPUA.TSTR.CSTj bit (j = 0 to 5) is cleared to 0). In PWM mode 2, the output at the time when the TCNT is cleared to 0 is specified as the initial output.

When buffer operation has been selected for register TGRC or TGRD, the settings of the IOC[3:0] or IOD[3:0] bits become ineffective, and the TGRC or TGRD register simply operates as a buffer.

To specify the input capture pin in TIORH, TIORL, or TIOR, set the bit in the port direction register (PDR) for the corresponding pin to 0 (input port), and set the bit in the port mode register (PMR) to 1 (uses the pin as an I/O port for peripheral functions). For details, see section 22, I/O Ports.

IOA[3:0] Bits (TGRA Control)

Select the function of TPUm.TGRA (m = 0 to 5).

IOB[3:0] Bits (TGRB Control)

Select the function of TPUm.TGRB (m = 0 to 5).

IOC[3:0] Bits (TGRC Control)

Select the function of TPUm.TGRC (m = 0, 3).

IOD[3:0] Bits (TGRD Control)

Select the function of TPUm.TGRD (m = 0, 3).

Table 26.13 TPU0.TIORH

Bits IOA[3:0]				Description	
b3	b2	b1	b0	TPU0.TGRA Function	TIOCA0 Pin Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA0 pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCA0 pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCA0 pin; input capture at both edges
1	1	x	x		Capture input source is TPU1 count clock; input capture at TPU1.TCNT count-up/ count-down*1

Bits IOB[3:0]				Description	
b7	b6	b5	b4	TPU0.TGRB Function	TIOCB0 Pin Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB0 or TIOCA0 pin*2; input capture at rising edge
1	0	0	1		Capture input source is TIOCB0 or TIOCA0 pin*2; input capture at falling edge
1	0	1	x		Capture input source is TIOCB0 or TIOCA0 pin*2; input capture at both edges
1	1	x	x		Capture input source is TPU1 count clock; input capture at TPU1.TCNT count-up/ count-down*1

x: Don’t care

Note 1. When the TPSC[2:0] bits in TPU1.TCR are set to 000b and PCLK/1 is used as the TPU1.TCNT count clock, this setting is invalid and input capture is not generated.

Note 2. Selected by the ICSELB bit in TPU0.TMDR.

Table 26.14 TPU1.TIOR

Bits IOA[3:0]				Description	
b3	**b2**	**b1**	**b0**	**TPU1.TGRA Function**	**TIOCA1 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA1 pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCA1 pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCA1 pin; input capture at both edges
1	1	x	x		Capture input source is TPU0.TGRA compare match/input capture; input capture at generation of TPU0.TGRA compare match/input capture

Bits IOB[3:0]				Description	
b7	**b6**	**b5**	**b4**	**TPU1.TGRB Function**	**TIOCB1 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB1 or TIOCA1 pin*1; input capture at rising edge
1	0	0	1		Capture input source is TIOCB1 or TIOCA1 pin*1; input capture at falling edge
1	0	1	x		Capture input source is TIOCB1 or TIOCA1 pin*1; input capture at both edges
1	1	x	x		Capture input source is TPU0.TGRC compare match/input capture; input capture at generation of TPU0.TGRC compare match/input capture

x: Don’t care

Note 1. Selected by the ICSELB bit in TPU1.TMDR.

Table 26.15 TPU2.TIOR

Bits IOA[3:0]				Description	
b3	**b2**	**b1**	**b0**	**TPU2.TGRA Function**	**TIOCA2 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	x	0	0	Input capture register	Capture input source is TIOCA2 pin; input capture at rising edge
1	x	0	1		Capture input source is TIOCA2 pin; input capture at falling edge
1	x	1	x		Capture input source is TIOCA2 pin; input capture at both edges

Bits IOB[3:0]				Description	
b7	**b6**	**b5**	**b4**	**TPU2.TGRB Function**	**TIOCB2 Pin (Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	x	0	0	Input capture register	Capture input source is TIOCB2 or TIOCA2 pin*1; input capture at rising edge
1	x	0	1		Capture input source is TIOCB2 or TIOCA2 pin*1; input capture at falling edge
1	x	1	x		Capture input source is TIOCB2 or TIOCA2 pin*1; input capture at both edges

x: Don’t care

Note 1. Selected by the ICSELB bit in TPU2.TMDR.

Table 26.16 TPU3.TIORH

Bits IOA[3:0]				Description	
b3	**b2**	**b1**	**b0**	**TPU3.TGRA Function**	**TIOCA3 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA3 pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCA3 pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCA3 pin; input capture at both edges
1	1	x	x		Capture input source is TPU4 count clock; input capture at TPU4.TCNT count-up/ count-down*1

Bits IOB[3:0]				Description	
b7	**b6**	**b5**	**b4**	**TPU3.TGRB Function**	**TIOCB3 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB3 or TIOCA3 pin*2; input capture at rising edge
1	0	0	1		Capture input source is TIOCB3 or TIOCA3 pin*2; input capture at falling edge
1	0	1	x		Capture input source is TIOCB3 or TIOCA3 pin*2; input capture at both edges
1	1	x	x		Capture input source is TPU4 count clock; input capture at TPU4.TCNT count-up/ count-down*1

x: Don’t care

Note 1. When the TPSC[2:0] bits in TPU4.TCR are set to 000b and PCLK/1 is used as the TPU4.TCNT count clock, this setting is invalid and input capture is not generated.

Note 2. Selected by the ICSELB bit in TPU3.TMDR.

Table 26.17 TPU4.TIOR

Bits IOA[3:0]				Description	
b3	**b2**	**b1**	**b0**	**TPU4.TGRA Function**	**TIOCA4 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA4 pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCA4 pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCA4 pin; input capture at both edges
1	1	x	x		Capture input source is TPU3.TGRA compare match/input capture; input capture at generation of TPU3.TGRA compare match/input capture

Bits IOB[3:0]				Description	
b7	**b6**	**b5**	**b4**	**TPU4.TGRB Function**	**TIOCB4 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB4 or TIOCA4 pin*1; input capture at rising edge
1	0	0	1		Capture input source is TIOCB4 or TIOCA4 pin*1; input capture at falling edge
1	0	1	x		Capture input source is TIOCB4 or TIOCA4 pin*1; input capture at both edges
1	1	x	x		Capture input source is TPU3.TGRC compare match/input capture; input capture at generation of TPU3.TGRC compare match/input capture

x: Don’t care

Note 1. Selected by the ICSELB bit in TPU4.TMDR.

Table 26.18 TPU5.TIOR

Bits IOA[3:0]				Description	
b3	**b2**	**b1**	**b0**	**TPU5.TGRA Function**	**TIOCA5 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	x	0	0	Input capture register	Capture input source is TIOCA5 pin; input capture at rising edge
1	x	0	1		Capture input source is TIOCA5 pin; input capture at falling edge
1	x	1	x		Capture input source is TIOCA5 pin; input capture at both edges

Bits IOB[3:0]				Description	
b7	**b6**	**b5**	**b4**	**TPU5.TGRB Function**	**TIOCB5 Pin Function and Related Issue**
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	x	0	0	Input capture register	Capture input source is TIOCB5 or TIOCA5 pin*1; input capture at rising edge
1	x	0	1		Capture input source is TIOCB5 or TIOCA5 pin*1; input capture at falling edge
1	x	1	x		Capture input source is TIOCB5 or TIOCA5 pin*1; input capture at both edges

x: Don’t care

Note 1. Selected by the ICSELB bit in TPU5.TMDR.

Table 26.19 TPU0.TIORL

Bits IOC[3:0]				Description	
b3	b2	b1	b0	TPU0.TGRC Function	TIOCC0 Pin Function and Related Issue
0	0	0	0	Output compare register*1	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register*1	Capture input source is TIOCC0 pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCC0 pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCC0 pin; input capture at both edges
1	1	x	x		Capture input source is TPU1 count clock; input capture at TPU1.TCNT count-up/ count-down*3

Bits IOD[3:0]				Description	
b7	b6	b5	b4	TPU0.TGRD Function	TIOCD0 Pin Function and Related Issue
0	0	0	0	Output compare register*2	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register*2	Capture input source is TIOCD0 or TIOCC0 pin*4; input capture at rising edge
1	0	0	1		Capture input source is TIOCD0 or TIOCC0 pin*4; input capture at falling edge
1	0	1	x		Capture input source is TIOCD0 or TIOCC0 pin*4; input capture at both edges
1	1	x	x		Capture input source is TPU1 count clock; input capture at TPU1.TCNT count-up/ count-down*3

x: Don’t care

Note 1. When the BFA bit in TPU0.TMDR is set to 1 (TPU0.TGRA and TPU0.TGRC are used for buffer operation) and TPU0.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note 2. When the BFB bit in TPU0.TMDR is set to 1 (TPU0.TGRB and TPU0.TGRD are used for buffer operation) and TPU0.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note 3. When the TPSC[2:0] bits in TPU1.TCR are set to 000b and PCLK/1 is used as the TPU0.TCNT count clock, this setting is invalid and input capture is not generated.

Note 4. Selected by the ICSELD bit in TPU0.TMDR.

Table 26.20 TPU3.TIORL

Bits IOC[3:0]				Description	
b3	**b2**	**b1**	**b0**	**TPU3.TGRC Function**	**TIOCC3 Pin Function and Related Issue**
0	0	0	0	Output compare register*1	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register*1	Capture input source is TIOCC3 pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCC3 pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCC3 pin; input capture at both edges
1	1	x	x		Capture input source is TPU4 count clock; input capture at TPU4.TCNT count-up/ count-down*3

Bits IOD[3:0]				Description	
b7	**b6**	**b5**	**b4**	**TPU3.TGRD Function**	**TIOCD3 Pin Function and Related Issue**
0	0	0	0	Output compare register*2	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register*2	Capture input source is TIOCD3 or TIOCC3 pin*4; input capture at rising edge
1	0	0	1		Capture input source is TIOCD3 or TIOCC3 pin*4; input capture at falling edge
1	0	1	x		Capture input source is TIOCD3 or TIOCC3 pin*4; input capture at both edges
1	1	X	x		Capture input source is TPU4 count clock; input capture at TPU4.TCNT count-up/ count-down*3

x: Don't care

Note 1. When the BFA bit in TPU3.TMDR is set to 1 (TPU3.TGRA and TPU3.TGRC are used for buffer operation) and TPU3.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note 2. When the BFB bit in TPU3.TMDR is set to 1 (TPU3.TGRB and TPU3.TGRD are used for buffer operation) and TPU3.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note 3. When the TPSC[2:0] bits in TPU4.TCR are set to 000b and PCLK/1 is used as the TPU3.TCNT count clock, this setting is invalid and input capture is not generated.

Note 4. Selected by the ICSELD bit in TPU3.TMDR.

26.2.4 Timer Interrupt Enable Register (TIER)

Address(es): TPU0.TIER 0008 8114h, TPU1.TIER 0008 8124h, TPU2.TIER 0008 8134h, TPU3.TIER 0008 8144h, TPU4.TIER 0008 8154h, TPU5.TIER 0008 8164h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE	—	TCIEU	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Value after reset:	0	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TGIEA	TGRA Interrupt Enable	0: Interrupt requests (TGImA) disabled 1: Interrupt requests (TGImA) enabled (m = 0 to 5)	R/W
b1	TGIEB	TGRB Interrupt Enable	0: Interrupt requests (TGImB) disabled 1: Interrupt requests (TGImB) enabled (m = 0 to 5)	R/W
b2	TGIEC	TGRC Interrupt Enable*1	0: Interrupt requests (TGImC) disabled 1: Interrupt requests (TGImC) enabled (m = 0, 3)	R/W
b3	TGIED	TGRD Interrupt Enable*1	0: Interrupt requests (TGImD) disabled 1: Interrupt requests (TGImD) enabled (m = 0, 3)	R/W
b4	TCIEV	Overflow Interrupt Enable	0: Interrupt requests (TCImV) disabled 1: Interrupt requests (TCImV) enabled (m = 0 to 5)	R/W
b5	TCIEU	Underflow Interrupt Enable*2	0: Interrupt requests (TCImU) disabled 1: Interrupt requests (TCImU) enabled (m = 1, 2, 4, 5)	R/W
b6	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7	TTGE	A/D Conversion Start Request Enable*3	0: A/D conversion start request generation disabled 1: A/D conversion start request generation enabled	R/W

Note 1. These bits are reserved in TPU1, TPU2, TPU4, and TPU5. These bits are read as 0. The write value should be 0.
Note 2. This bit is reserved in TPU0 and TPU3. This bit is read as 0. The write value should be 0.
Note 3. This bit is reserved in TPU5. This bit is read as 0. The write value should be 0.

TTGE Bit (A/D Conversion Start Request Enable)

Enables/disables generation of A/D conversion start requests by TPUm.TGRA (m = 0 to 4) input capture/compare match.

26.2.5 Timer Status Register (TSR)

Address(es): TPU0.TSR 0008 8115h, TPU1.TSR 0008 8125h, TPU2.TSR 0008 8135h, TPU3.TSR 0008 8145h, TPU4.TSR 0008 8155h, TPU5.TSR 0008 8165h

	b7	b6	b5	b4	b3	b2	b1	b0
	TCFD	—	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA
Value after reset:	1	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TGFA	Input Capture/Output Compare Flag A	0: Input capture to TPUm.TGRA or compare match with TPUm.TGRA has not occurred. 1: Input capture to TPUm.TGRA or compare match with TPUm.TGRA has occurred. (m = 0 to 5)	R/W*1
b1	TGFB	Input Capture/Output Compare Flag B	0: Input capture to TPUm.TGRB or compare match with TPUm.TGRB has not occurred. 1: Input capture to TPUm.TGRB or compare match with TPUm.TGRB has occurred. (m = 0 to 5)	R/W*1
b2	TGFC	Input Capture/Output Compare Flag C*2	0: Input capture to TPUm.TGRC or compare match with TPUm.TGRC has not occurred. 1: Input capture to TPUm.TGRC or compare match with TPUm.TGRC has occurred. (m = 0, 3)	R/W*1
b3	TGFD	Input Capture/Output Compare Flag D*2	0: Input capture to TPUm.TGRD or compare match with TPUm.TGRD has not occurred. 1: Input capture to TPUm.TGRD or compare match with TPUm.TGRD has occurred. (m = 0, 3)	R/W*1
b4	TCFV	Overflow Flag	0: TPUm.TCNT has not overflowed. 1: TPUm.TCNT has overflowed. (m = 0 to 5)	R/W*1
b5	TCFU	Underflow Flag*3	0: TPUm.TCNT has not underflowed. 1: TPUm.TCNT has underflowed. (m = 1, 2, 4, 5)	R/W*1
b6	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7	TCFD	Counting Direction Flag*4	0: TPUm.TCNT counts down. 1: TPUm.TCNT counts up. (m = 1, 2, 4, 5)	R

Note 1. Only writing 0 to this bit is possible; this clears the flag.
Note 2. These bits are reserved in TPU1, TPU2, TPU4, and TPU5. The bits are read as 0. The write value should be 0.
Note 3. This bit is reserved in TPU0 and TPU3. The bit is read as 0. The write value should be 0.
Note 4. This bit is reserved in TPU0 and TPU3. The bit is read as 1. The write value should be 1.

TGFA Flag (Input Capture/Output Compare Flag A)

This status flag indicates that input capture to TPUm.TGRA or compare match with TPUm.TGRA (m = 0 to 5) has occurred.

[Setting conditions]

- When TPUm.TGRA holds the value for comparison in output-compare operations, TPUm.TCNT matches TPUm.TGRA.
- When TPUm.TGRA is serving as an input-capture register, the input-capture signal has caused transfer of the value in TPUm.TCNT to TPUm.TGRA.

[Clearing conditions]

- Activation of the DTC by the TGImA interrupt and clearing of the DTC.MRB.DISEL bit.
- Writing 0 to TGFA after reading its value as 1.

TGFB Flag (Input Capture/Output Compare Flag B)

This status flag indicates that input capture to TPUm.TGRB or compare match with TPUm.TGRB (m = 0 to 5) has occurred.

[Setting conditions]

- When TPUm.TGRB holds the value for comparison in output-compare operations, TPUm.TCNT matches TPUm.TGRB.
- When TPUm.TGRB is serving as an input-capture register, the input-capture signal has caused transfer of the value in TPUm.TCNT to TPUm.TGRB.

[Clearing conditions]

- Activation of the DTC by the TGImB interrupt and clearing of the DTC.MRB.DISEL bit.
- Writing 0 to TGFB after reading its value as 1.

TGFC Flag (Input Capture/Output Compare Flag C)

This status flag indicates that input capture to TPUm.TGRC or compare match with TPUm.TGRC (m = 0, 3) has occurred.

[Setting conditions]

- When TPUm.TGRC holds the value for comparison in output-compare operations, TPUm.TCNT matches TPUm.TGRC.
- When TPUm.TGRC is serving as an input-capture register, the input-capture signal has caused transfer of the value in TPUm.TCNT to TPUm.TGRC.

[Clearing conditions]

- Activation of the DTC by the TGImC interrupt and clearing of the DTC.MRB.DISEL bit.
- Writing 0 to TGFC after reading its value as 1.

TGFD Flag (Input Capture/Output Compare Flag D)

This status flag indicates that input capture to TPUm.TGRD or compare match with TPUm.TGRD (m = 0, 3) has occurred.

[Setting conditions]

- When TPUm.TGRD holds the value for comparison in output-compare operations, TPUm.TCNT matches TPUm.TGRD.
- When TPUm.TGRD is serving as an input-capture register, the input-capture signal has caused transfer of the value in TPUm.TCNT to TPUm.TGRD.

[Clearing conditions]

- Activation of the DTC by the TGImD interrupt and clearing of the DTC.MRB.DISEL bit.
- Writing 0 to TGFD after reading its value as 1.

TCFV Flag (Overflow Flag)

This status flag indicates an overflow of TPUm.TCNT (m = 0 to 5).

[Setting condition]

- Overflow of the value in TPUm.TCNT (TCNT counted from FFFFh to 0000h).

[Clearing condition]

- Writing 0 to TCFV after reading its value as 1.

TCFU Flag (Underflow Flag)

This status flag indicates an underflow of TPUm.TCNT (m = 1, 2, 4, 5).

[Setting condition]

- Underflow of the value in TPUm.TCNT (TCNT counted from 0000h to FFFFh).

[Clearing condition]

- Writing 0 to TCFU after reading its value as 1.

26.2.6 Timer Counter (TCNT)

Address(es): TPU0.TCNT 0008 8116h, TPU1.TCNT 0008 8126h, TPU2.TCNT 0008 8136h, TPU3.TCNT 0008 8146h, TPU4.TCNT 0008 8156h, TPU5.TCNT 0008 8166h

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

TPUm.TCNT is a readable/writable counter that counts the internal clock or external events.

26.2.7 Timer General Register A (TGRA), Timer General Register B (TGRB), Timer General Register C (TGRC), Timer General Register D (TGRD)

Address(es): TPU0.TGRA 0008 8118h, TPU0.TGRB 0008 811Ah, TPU0.TGRC 0008 811Ch, TPU0.TGRD 0008 811Eh,
TPU1.TGRA 0008 8128h, TPU1.TGRB 0008 812Ah,
TPU2.TGRA 0008 8138h, TPU2.TGRB 0008 813Ah,
TPU3.TGRA 0008 8148h, TPU3.TGRB 0008 814Ah, TPU3.TGRC 0008 814Ch, TPU3.TGRD 0008 814Eh,
TPU4.TGRA 0008 8158h, TPU4.TGRB 0008 815Ah,
TPU5.TGRA 0008 8168h, TPU5.TGRB 0008 816Ah

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

TPU has 16 TGR registers in total, four each for TPU0 and TPU3, and two each for TPU1, TPU2, TPU4, and TPU5. TPUm.TGRA (m = 0 to 5), TPUm.TGRB (m = 0 to 5), TPUm.TGRC (m = 0, 3), and TPUm.TGRD (m = 0, 3) are readable/writable registers with a dual function as output compare and input capture registers.
TPUm.TGRC and TPUm.TGRD can also be specified for operation as buffer registers. Register combinations during buffer operations are TPUm.TGRA—TPUm.TGRC and TPUm.TGRB—TPUm.TGRD.

26.2.8 Timer Start Register (TSTR)

Address(es): TPUA.TSTR 0008 8100h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CST5	CST4	CST3	CST2	CST1	CST0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CST0	Counter Start 0	0: TCNT count operation is stopped 1: TCNT performs count operation	R/W
b1	CST1	Counter Start 1		R/W
b2	CST2	Counter Start 2		R/W
b3	CST3	Counter Start 3		R/W
b4	CST4	Counter Start 4		R/W
b5	CST5	Counter Start 5		R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TPUA.TSTR starts or stops TCNT operation for TPU0 to TPU5.
Before setting the operating mode in TPUm.TMDR or setting the TPUm.TCNT count clock in TPUm.TCR, stop the TPUm.TCNT operation.

CSTn Bit (Counter Start n) (n = 0 to 5)

This bit starts or stop the TCNT.
When the CSTn bit is cleared to 0 with CSTn = 1 and the corresponding TIOCyn pin (y = A to D; n = 0 to 5) specified for output, count operation stops but the output compare output level of the corresponding TIOCyn pin is retained.
If TIORH, TIORL, or TIOR is written to when the CSTn bit is 0, the pin output level will be changed to the set initial output value.

26.2.9 Timer Synchronous Register (TSYR)

Address(es): TPUA.TSYR 0008 8101h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SYNC0	Timer Synchronization 0	0: TCNT operates independently (TCNT setting/clearing is unrelated to other channels) 1: TCNT performs synchronous operation*1 (TCNT synchronous setting/synchronous clearing is possible)	R/W
b1	SYNC1	Timer Synchronization 1		R/W
b2	SYNC2	Timer Synchronization 2		R/W
b3	SYNC3	Timer Synchronization 3		R/W
b4	SYNC4	Timer Synchronization 4		R/W
b5	SYNC5	Timer Synchronization 5		R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To set synchronous operation, the SYNCn bit (n = 0 to 5) for at least two channels must be set to 1. To set synchronous clearing, the TCNT clearing source must also be set by the TCR.CCLR[2:0] bits in addition to the SYNCn bit.

TPUA.TSYR selects independent operation or synchronous operation for TCNT of TPU0 to TPU5.

SYNCn Bit (Timer Synchronization n) (n = 0 to 5)

This bit selects whether the TCNT operation is independent of or synchronized with TCNT of other channels. When synchronous operation is selected, synchronous setting of multiple TCNT and synchronous clearing through counter clearing on another channel are possible.

26.2.10 Noise Filter Control Register (NFCR)

Address(es): TPU0.NFCR 0008 8108h, TPU1.NFCR 0008 8109h, TPU2.NFCR 0008 810Ah, TPU3.NFCR 0008 810Bh, TPU4.NFCR 0008 810Ch, TPU5.NFCR 0008 810Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	NFCS[1:0]		NFDEN	NFCEN	NFBEN	NFAEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NFAEN	Noise Filter Enable A	0: The noise filter for TIOCAm is disabled. 1: The noise filter for TIOCAm is enabled. (m = 0 to 5)	R/W
b1	NFBEN	Noise Filter Enable B	0: The noise filter for TIOCBm is disabled. 1: The noise filter for TIOCBm is enabled. (m = 0 to 5)	R/W
b2	NFCEN	Noise Filter Enable C*1	0: The noise filter for TIOCCm is disabled. 1: The noise filter for TIOCCm is enabled. (m = 0, 3)	R/W
b3	NFDEN	Noise Filter Enable D*1	0: The noise filter for TIOCDm is disabled. 1: The noise filter for TIOCDm is enabled. (m = 0, 3)	R/W
b5, b4	NFCS[1:0]	Noise Filter Clock Select	b5 b4 0 0: PCLK/1 0 1: PCLK/8 1 0: PCLK/32 1 1: 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. These bits are reserved in TPU1, TPU2, TPU4, and TPU5. The bits are read as 0. Writing to these bits is not possible.

Only set the TPUm.NFCR registers while the TPUm.TCNT is stopped.

NFAEN Bit (Noise Filter Enable A)

This bit disables or enables the noise filter for the TIOCAm pin (m = 0 to 5).

Since unexpected edges may be internally generated when the value of NFAEN is changed, select the output compare function in the timer I/O control register before changing the NFAEN value.

NFBEN Bit (Noise Filter Enable B)

This bit disables or enables the noise filter for the TIOCBm pin (m = 0 to 5).

Since unexpected edges may be internally generated when the value of NFBEN is changed, select the output compare function in the timer I/O control register before changing the NFBEN value.

NFCEN Bit (Noise Filter Enable C)

This bit disables or enables the noise filter for the TIOCCm pin (m = 0, 3).

Since unexpected edges may be internally generated when the value of NFCEN is changed, select the output compare function in the timer I/O control register before changing the NFCEN value.

NFDEN Bit (Noise Filter Enable D)

This bit disables or enables the noise filter for the TIOCDm pin (m = 0, 3).

Since unexpected edges may be internally generated when the value of NFDEN is changed, select the output compare function in the timer I/O control register before changing the NFDEN value.

NFCS[1:0] Bits (Noise Filter Clock Select)

These bits select the sampling clock for the noise filter.

When the count source is selected with NFCS[1:0] bits set to 11b, the clock that can be used as sampling clock are the internal clocks other than PCLK/1 specified with the TPSC[2:0] bits and the external clock. To select the PCLK/1 as both the count clock and the sampling clock, set the NFCS[1:0] bits to 00b.

The input-capture signal is sampled on rising edges of the selected clock signal. If the sampled levels match three times in a row, the given level is passed through as the input-capture signal. If the levels do not match, the existing value is retained.

After setting the NFCS[1:0] bits, wait for two selected sampling periods before setting the input capture function.

26.3 Operation

26.3.1 Basic Functions

Each channel has a TPUm.TCNT and a TPUm.TGRy register (y = A to D).
TCNT is a 16-bit up-counter, which can function as a free-running counter, periodic counter, or event counter.
TGRy can be used as an input capture register or output compare register.

(1) Counter Operation

When the CSTj bit (j = 0 to 5) in TPUA.TSTR is set to 1, the TCNT for the corresponding channel starts counting.

(a) Example of count operation setting procedure

Figure 26.2 shows an example of the count operation setting procedure.

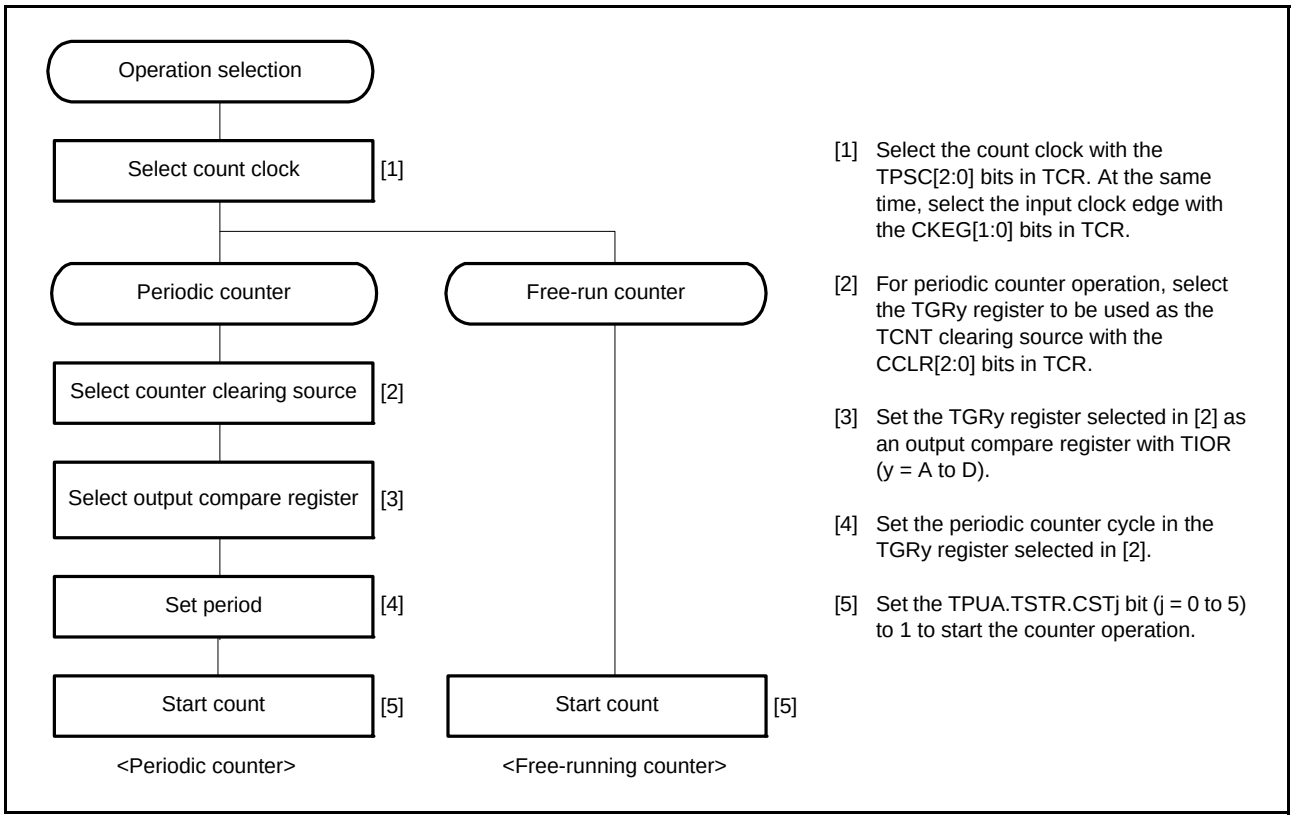


Figure 26.2 Example of Counter Operation Setting Procedure

(b) Free-running count operation and periodic count operation

Immediately after a reset, TPUm.TCNT are all set as free-running counters. When the relevant bit in TPUA.TSTR is set to 1, the corresponding TCNT starts up-count operation as a free-running counter. When TCNT overflows (changes from FFFFh to 0000h), the TPU requests an interrupt. After an overflow, TCNT restarts counting up from 0000h.
Figure 26.3 shows free-running counter operation.

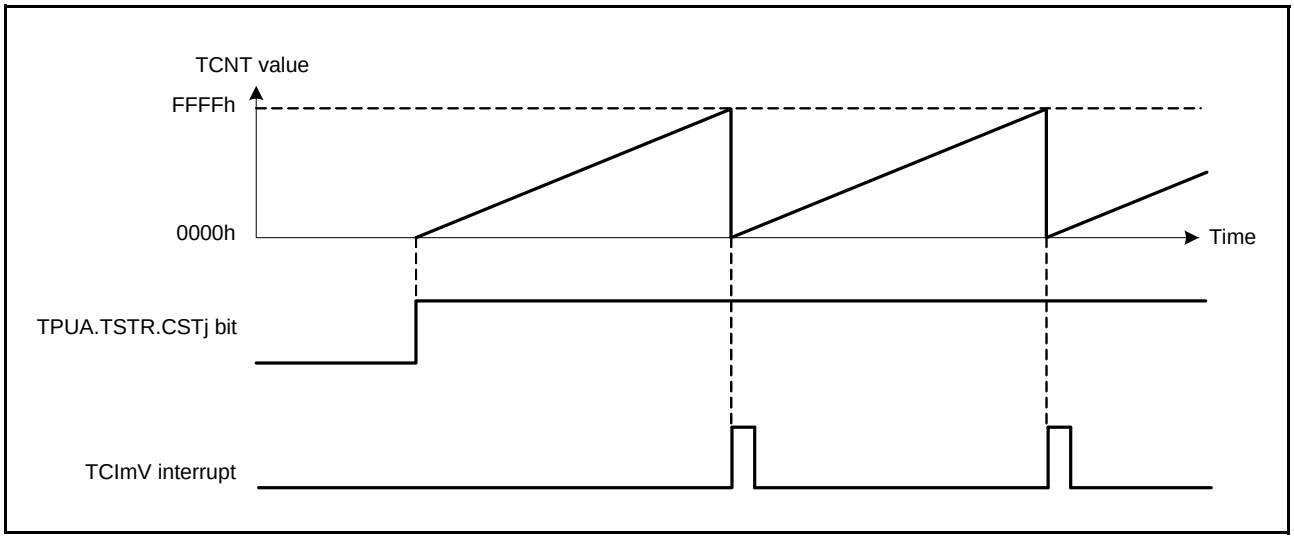


Figure 26.3 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT for the relevant channel performs periodic count operation. The TPUm.TGRy for setting the period is set as an output compare register, and counter clearing by compare match is selected by the TPUm.TCR.CCLR[2:0] bits. After the settings have been made, TCNT starts count-up operation as a periodic counter when the corresponding bit in TPUA.TSTR is set to 1. When the count value matches the TGRy value, TCNT is cleared to 0000h.
At this time, the TPU requests an interrupt. After a compare match, TCNT restarts counting up from 0000h.
Figure 26.4 shows periodic counter operation.

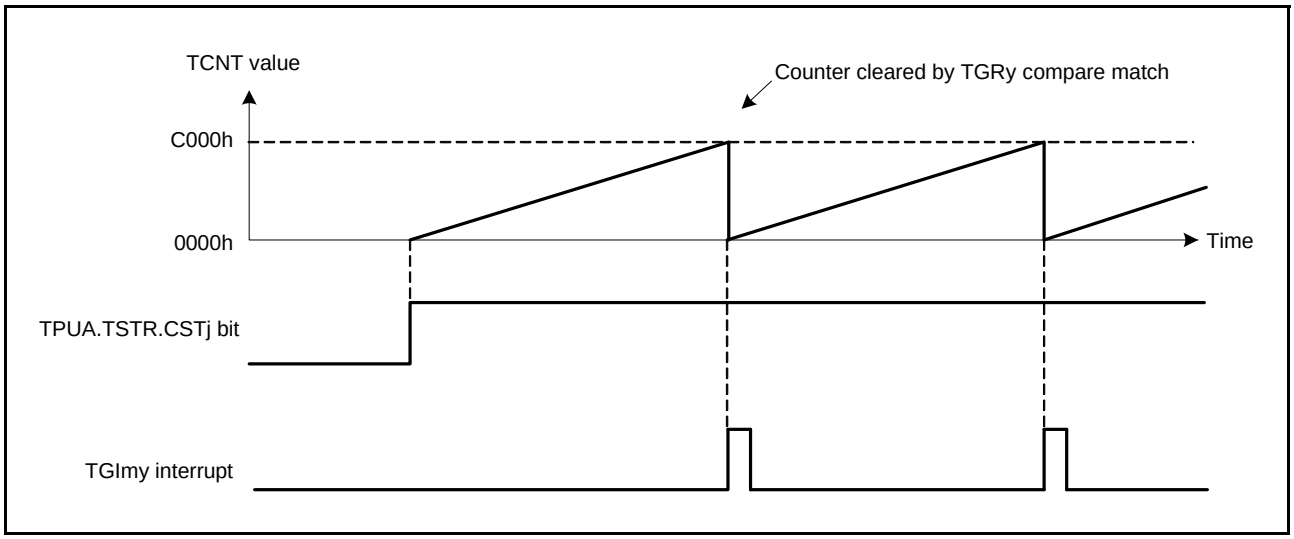


Figure 26.4 Periodic Counter Operation

(2) Waveform Output by Compare Match

The TPU can perform low, high, or toggle output from the corresponding output pin using a compare match.

(a) Example of setting procedure for waveform output by compare match

Figure 26.5 shows an example of the setting procedure for waveform output by a compare match.

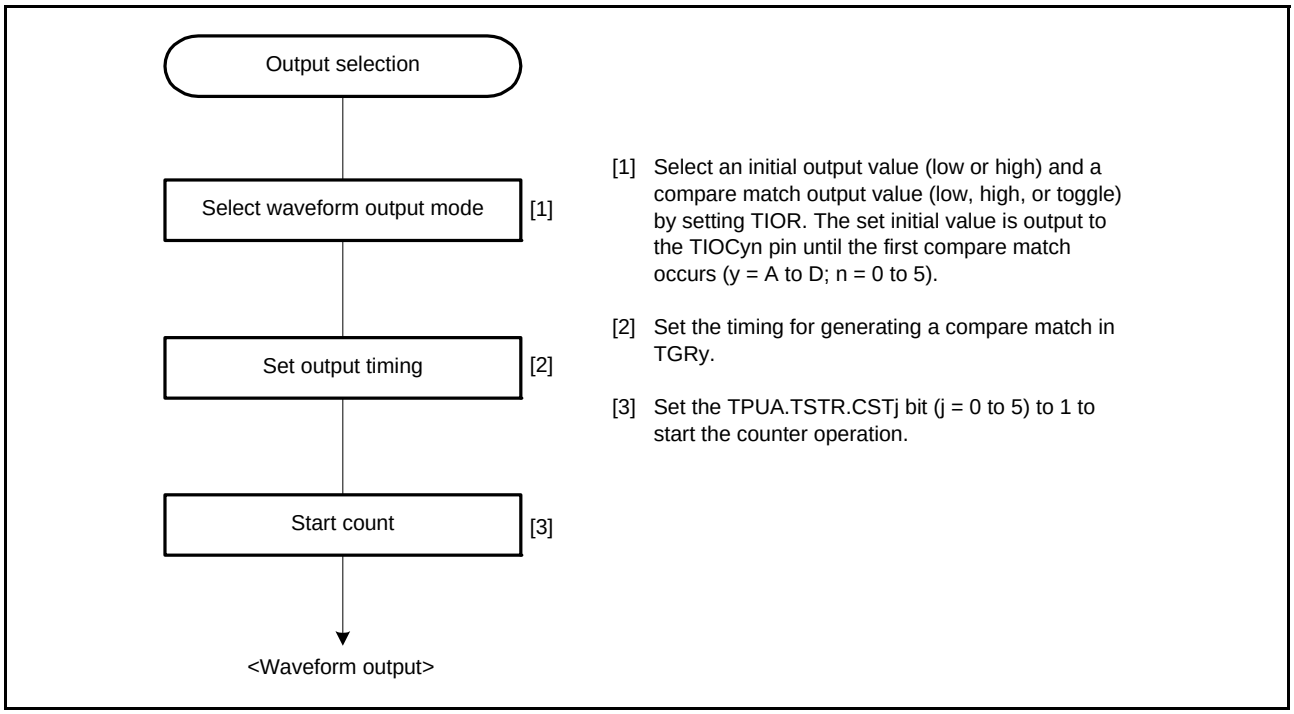


Figure 26.5 Example of Setting Procedure for Waveform Output by Compare Match

(b) Examples of waveform output operation

Figure 26.6 shows an example of low output/high output.

In this example, TPUm.TCNT has been set as a free-running counter, and settings have been made so that high is output by compare match A and low is output by compare match B. When the set level and the pin level match, the pin level does not change.

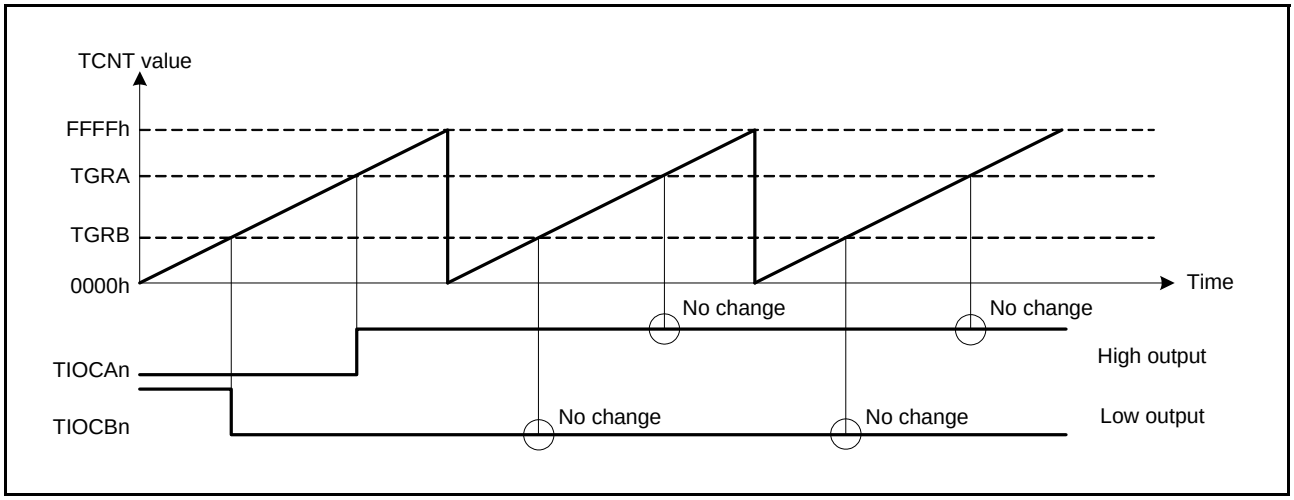


Figure 26.6 Example of Low-Output/High-Output Operation (n = 0 to 5)

Figure 26.7 shows an example of toggle output.
In this example, TPUm.TCNT has been set as a periodic counter (with counter clearing performed by compare match B), and settings have been made so that output is toggled by both compare match A and compare match B.

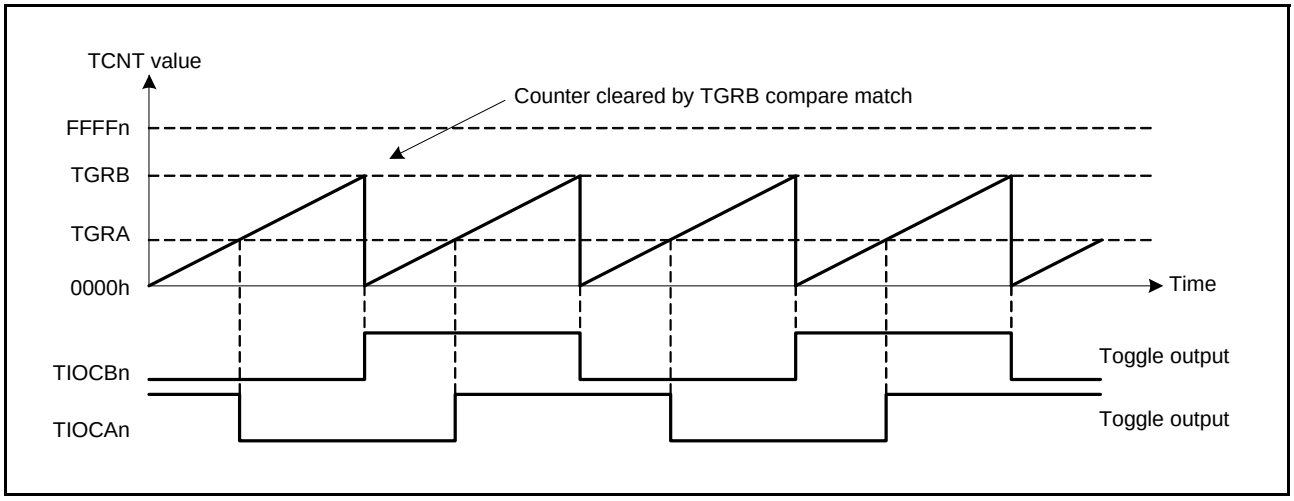


Figure 26.7 Example of Toggle Output Operation (n = 0 to 5)

(3) Input Capture Function

The TPUm.TCNT value can be transferred to TPUm.TGRy on detection of the TIOCyn pin (y = A to D; n = 0 to 5) input edge.
The rising edge, the falling edge, or both edges can be selected as the detection edge. It is also possible to specify the count clock or compare match signal of TPU0, TPU1, TPU3, and TPU4 as the input capture source. Noise filtering can be applied to the input capture input.

Note: Even if the counter is halted, an input capture is generated, and flag and interrupt signals are generated.
Note: When another channel's count clock is used as the input capture input for TPU0 and TPU3, PCLK/1 should not be selected as the count clock used for input capture input. Input capture will not be generated if PCLK/1 is selected.

(a) Example of setting procedure for input capture operation

Figure 26.8 shows an example of the setting procedure for input capture operation.

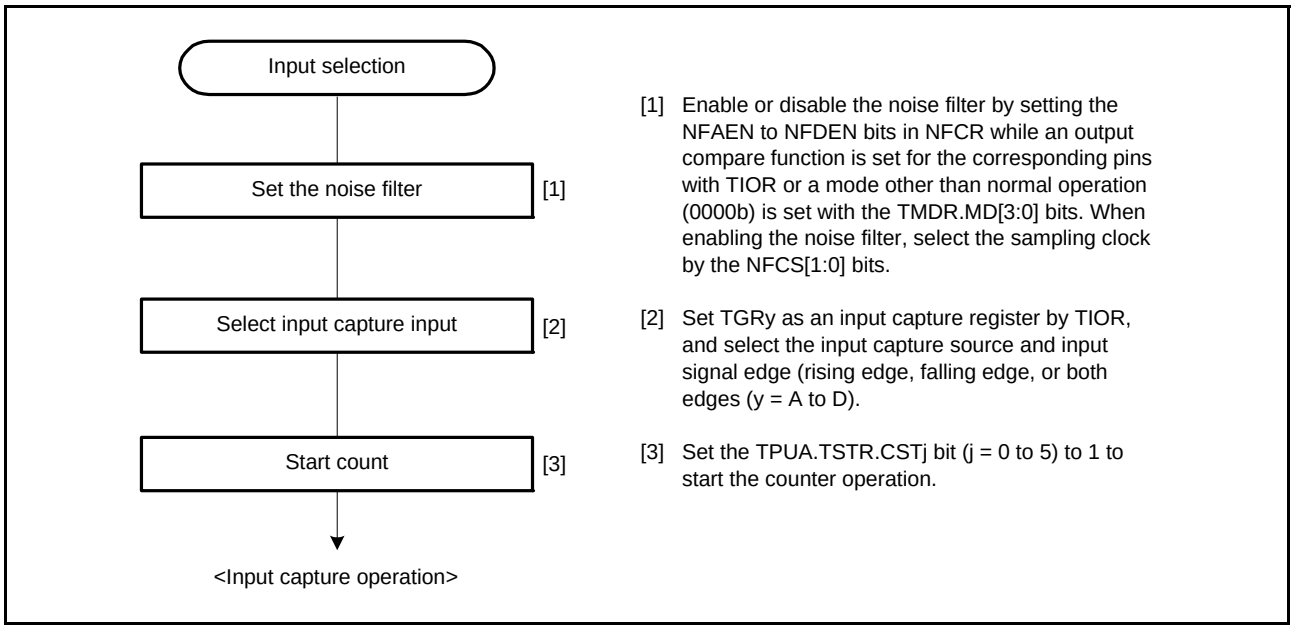


Figure 26.8 Example of Setting Procedure for Input Capture Operation

(b) Example of input capture operation

Figure 26.9 shows an example of input capture operation when the noise filter is stopped.
In this example, both rising and falling edges have been selected as the TIOCAn pin input capture input edge, the falling edge has been selected as the TIOCBn pin input capture input edge, and counter clearing by TPUm.TGRB input capture has been set for TPUm.TCNT.

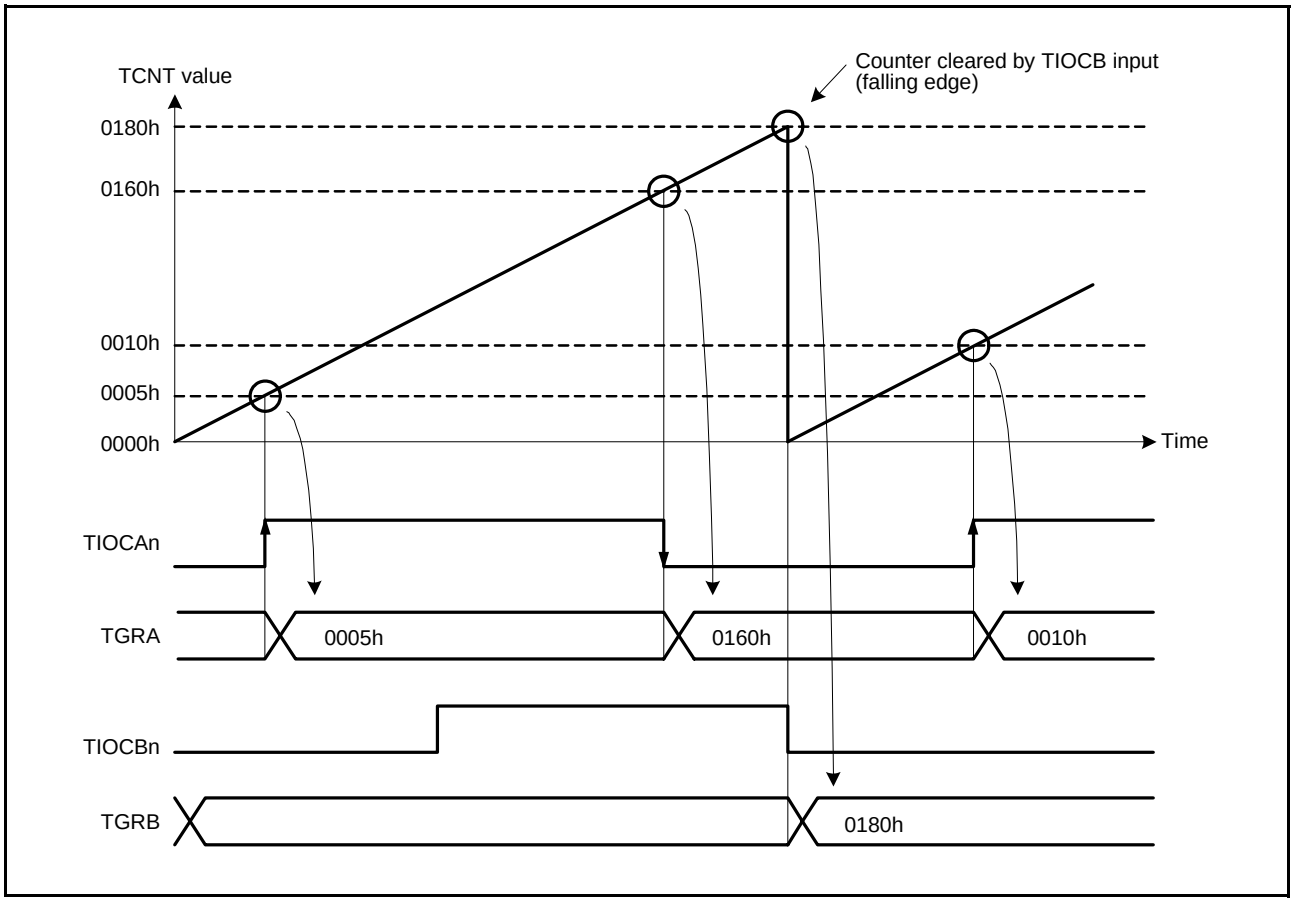


Figure 26.9 Example of Input Capture Operation (with Noise Filter Stopped) (n = 0 to 5)

When noise filtering is enabled, see Figure 26.30.

26.3.2 Synchronous Operation

In synchronous operation, the values in multiple TPUm.TCNT can be rewritten simultaneously (synchronous setting). Also, multiple TCNT can be cleared simultaneously (synchronous clearing) by making the appropriate setting in TPUm.TCR.
Synchronous operation enables TPUm.TGRy to be incremented with respect to a single time base.
TPU0 to TPU5 can all be set for synchronous operation.

(1) Example of Synchronous Operation Setting Procedure

Figure 26.10 shows an example of the synchronous operation setting procedure.

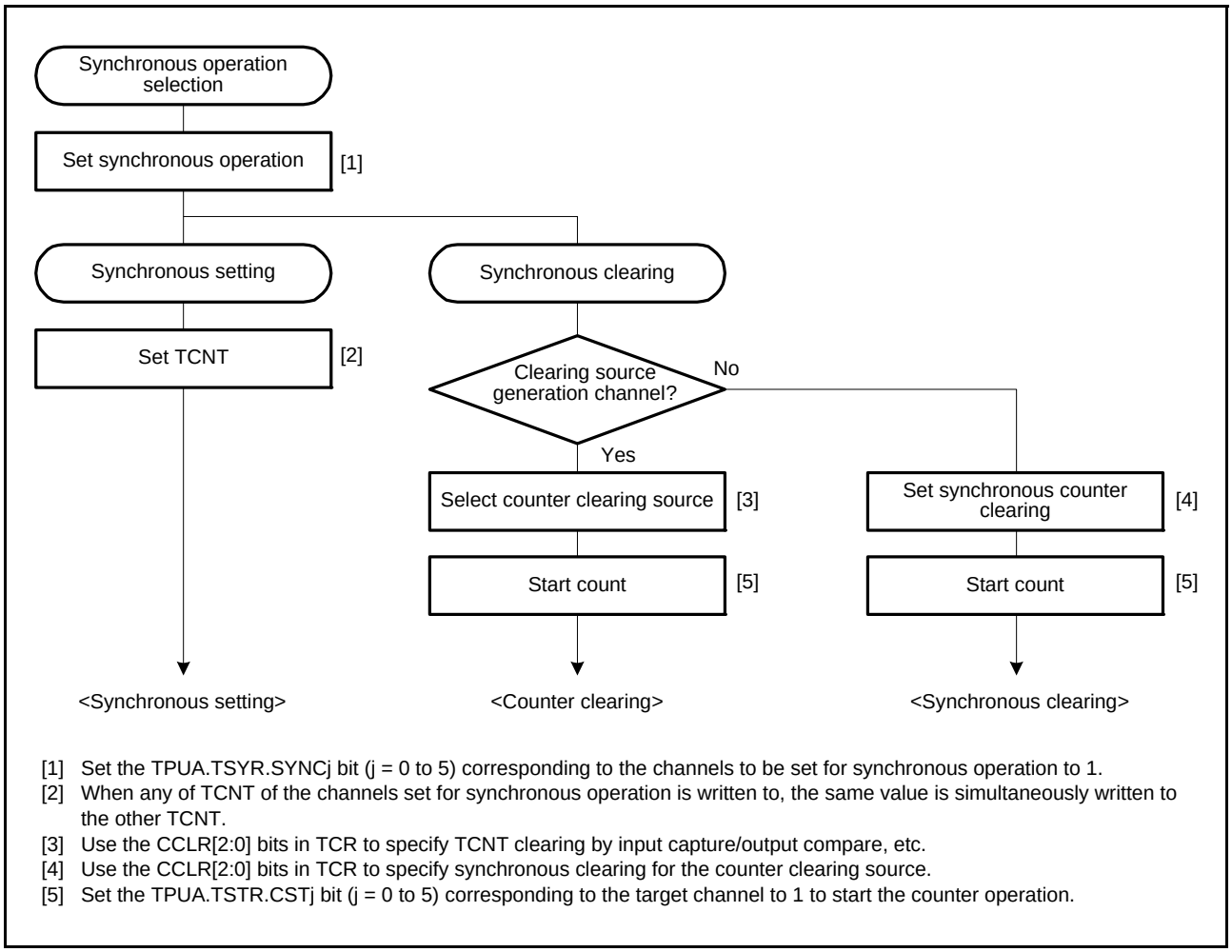


Figure 26.10 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 26.11 shows an example of synchronous operation.
In this example, synchronous operation and PWM mode 1 have been set for TPU0 to TPU2, TPU0.TGRB compare match has been set as the TPU0 counter clearing source, and synchronous clearing has been set for the TPU1 and TPU2 counter clearing source.
Three-phase PWM waveforms are output from pins TIOCA0, TIOCA1, and TIOCA2. At this time, synchronous setting and synchronous clearing by TPU0.TGRB compare match are performed for TPUm.TCNT of TPU0 to TPU2, and the data set in TPU0.TGRB is used as the PWM cycle.
For details on PWM modes, see section 26.3.5, PWM Modes.

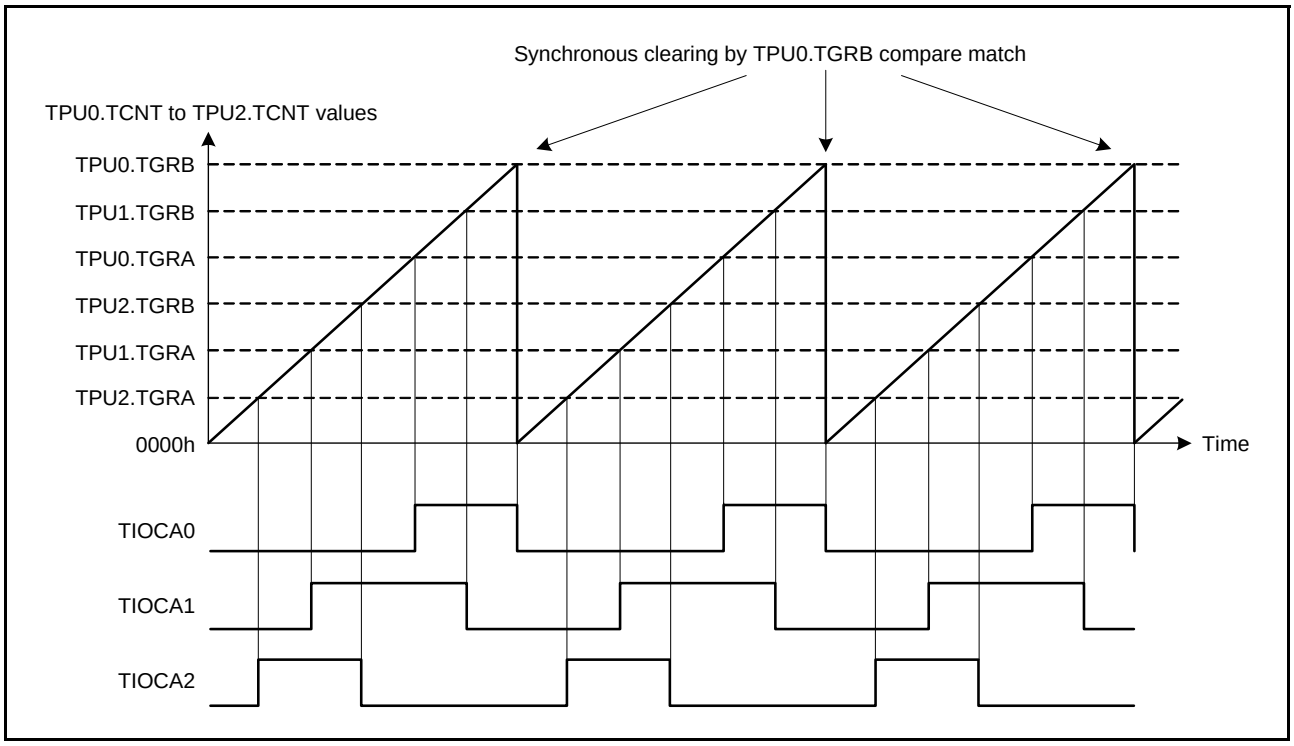


Figure 26.11 Example of Synchronous Operation

26.3.3 Buffer Operation

Buffer operation, provided for TPU0 and TPU3, enables TPUm.TGRC and TPUm.TGRD to be used as buffer registers. Buffer operation differs depending on whether TPUm.TGRy has been set as an input capture register or a compare match register.

Table 26.21 lists the register combinations used in buffer operation.

Table 26.21 Register Combinations

Channel	Timer General Register	Buffer Register
TPU0	TPU0.TGRA	TPU0.TGRC
	TPU0.TGRB	TPU0.TGRD
TPU3	TPU3.TGRA	TPU3.TGRC
	TPU3.TGRB	TPU3.TGRD

- When TPUm.TGRy is an output compare register

When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register.

This operation is shown in Figure 26.12.

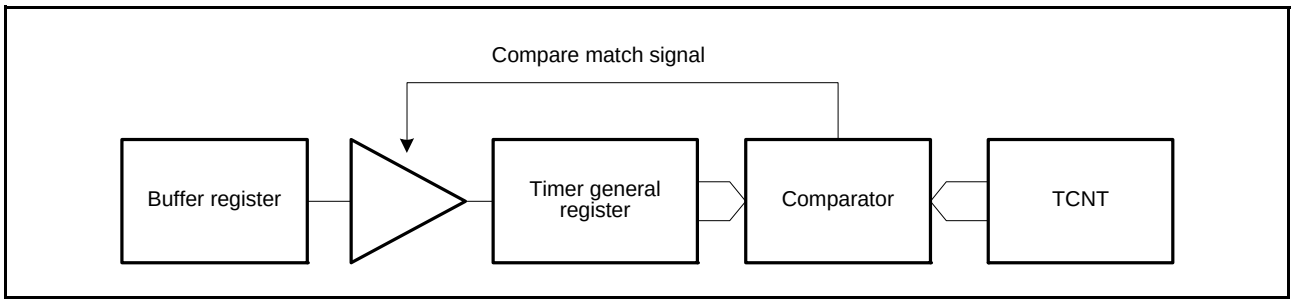


Figure 26.12 Compare Match Buffer Operation

- When TPUm.TGRy is an input capture register

When input capture occurs, the value in TPUm.TCNT is transferred to TGRy and the value previously held in TGRy is simultaneously transferred to the buffer register.

This operation is shown in Figure 26.13.

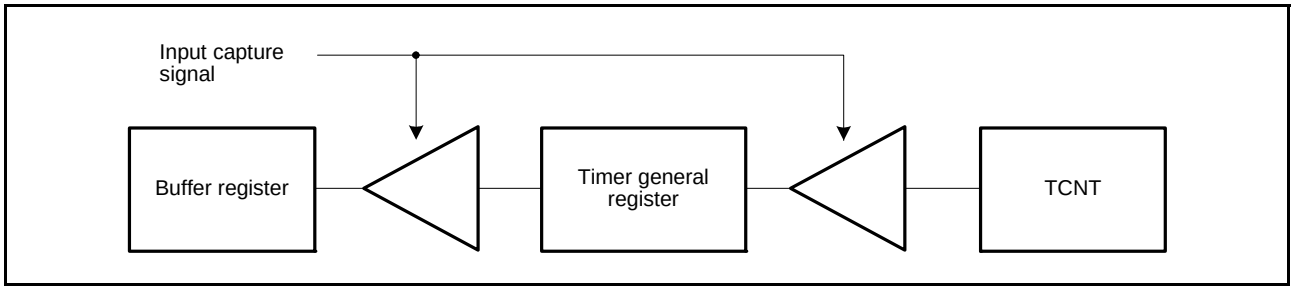


Figure 26.13 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 26.14 shows an example of the buffer operation setting procedure.

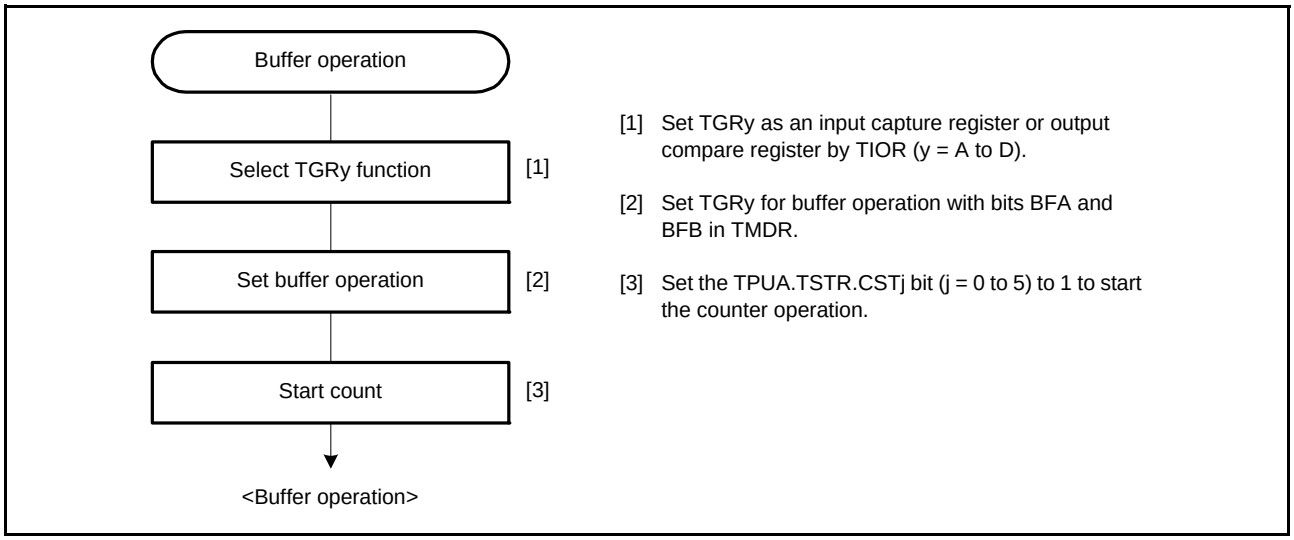


Figure 26.14 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TPUm.TGRy is an output compare register

Figure 26.15 shows an operation example in which PWM mode 1 has been set for TPU0, and buffer operation has been set for TPU0.TGRA and TPU0.TGRC. The settings used in this example are TPU0.TCNT clearing by compare match B, high output at compare match A, and low output at compare match B.

As buffer operation has been set, when compare match A occurs, the output changes and the TPU0.TGRC value is simultaneously transferred to TPU0.TGRA. This operation is repeated each time compare match A occurs.

For details on PWM modes, see section 26.3.5, PWM Modes.

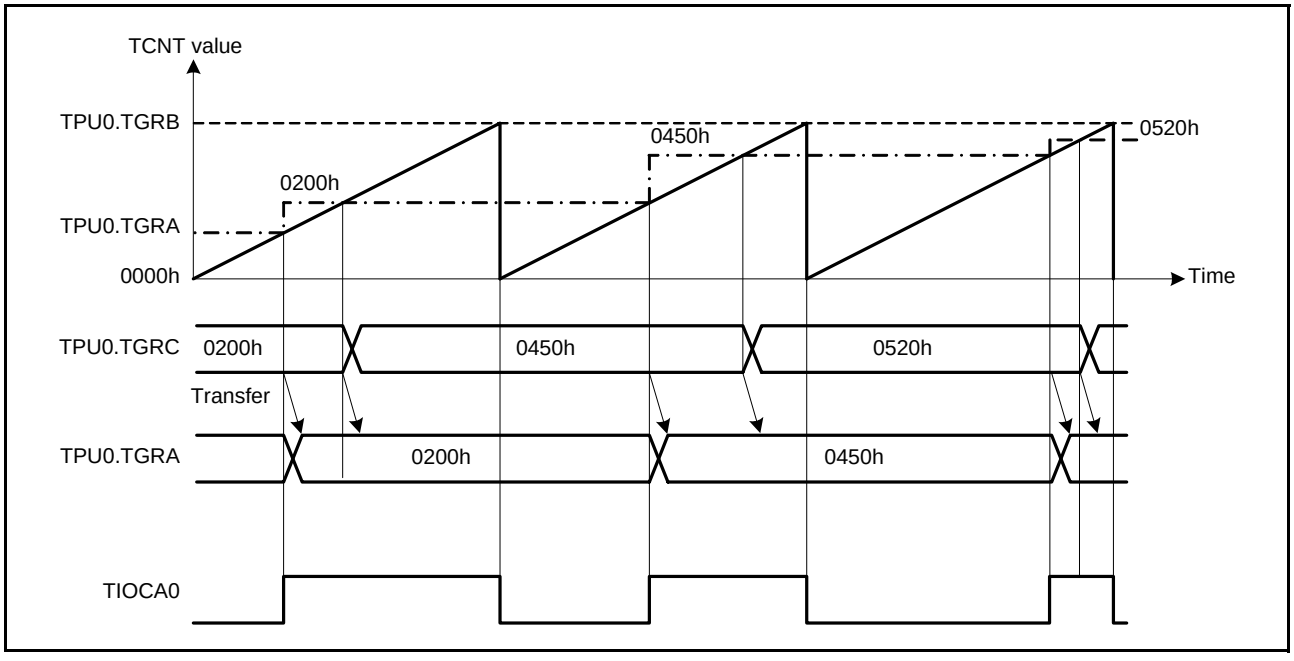


Figure 26.15 Example of Buffer Operation (1)

(b) When TPUm.TGRy is an input capture register

Figure 26.16 shows an operation example in which TPUm.TGRA has been set as an input capture register, and buffer operation has been set for the TGRA register and TPUm.TGRC.
Counter clearing by TGRA input capture has been set for TPUm.TCNT, and both rising and falling edges have been selected as the TIOCAn pin input capture input edge.
As buffer operation has been set, when the TCNT value is stored in TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

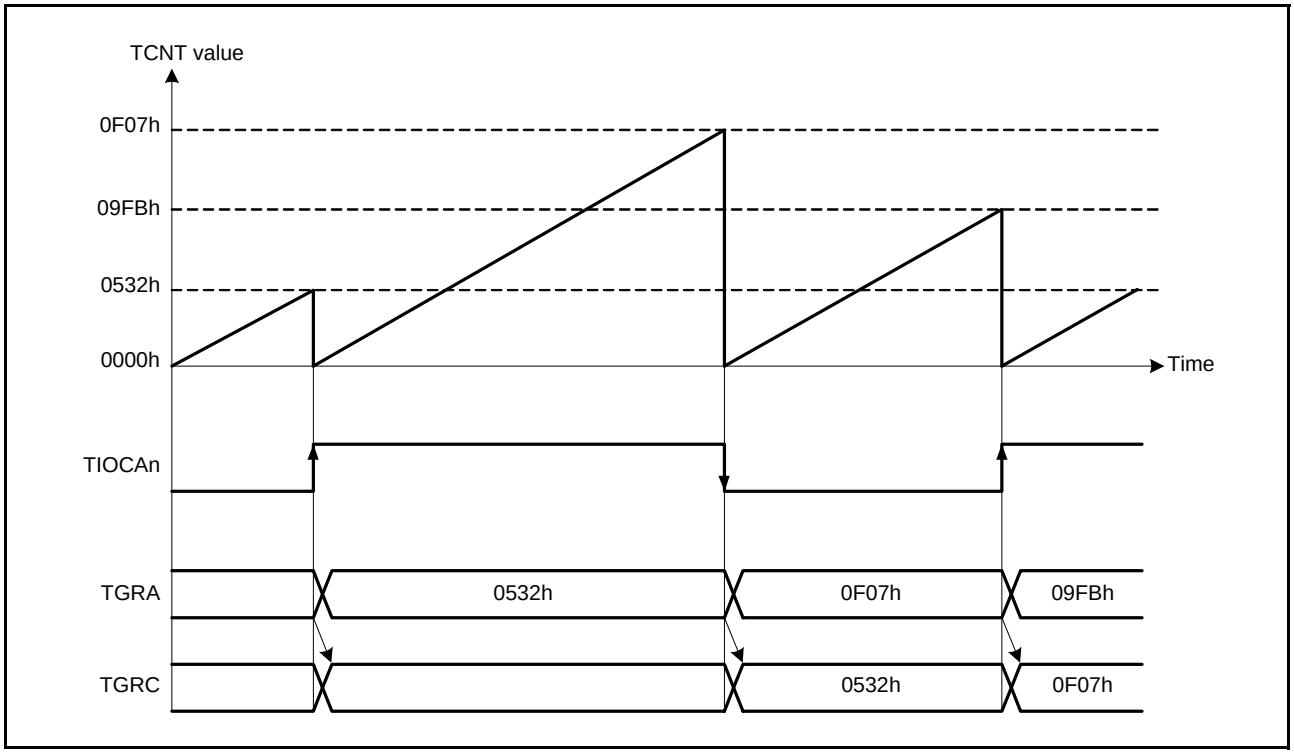


Figure 26.16 Example of Buffer Operation (2) (n = 0 to 5)

26.3.4 Cascaded Operation

In cascaded operation, two 16-bit counters for different channels are used together as a 32-bit counter.
This function works by counting the TPU1 (TPU4) count clock at overflow/underflow of TPU2.TCNT (TPU5.TCNT) as set by the TPSC[2:0] bits in TPU1.TCR (TPSC[2:0] bits in TPU4.TCR).
Underflow occurs only when the lower 16-bit TPUm.TCNT is in phase counting mode.
Table 26.22 lists the register combinations used in cascaded operation.

Note: When phase counting mode is set for TPU1 or TPU4, the count clock setting is invalid and the counter operates independently in phase counting mode.

Table 26.22 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
TPU1 and TPU2	TPU1.TCNT	TPU2.TCNT
TPU4 and TPU5	TPU4.TCNT	TPU5.TCNT

(1) Example of Cascaded Operation Setting Procedure

Figure 26.17 shows an example of the setting procedure for cascaded operation.

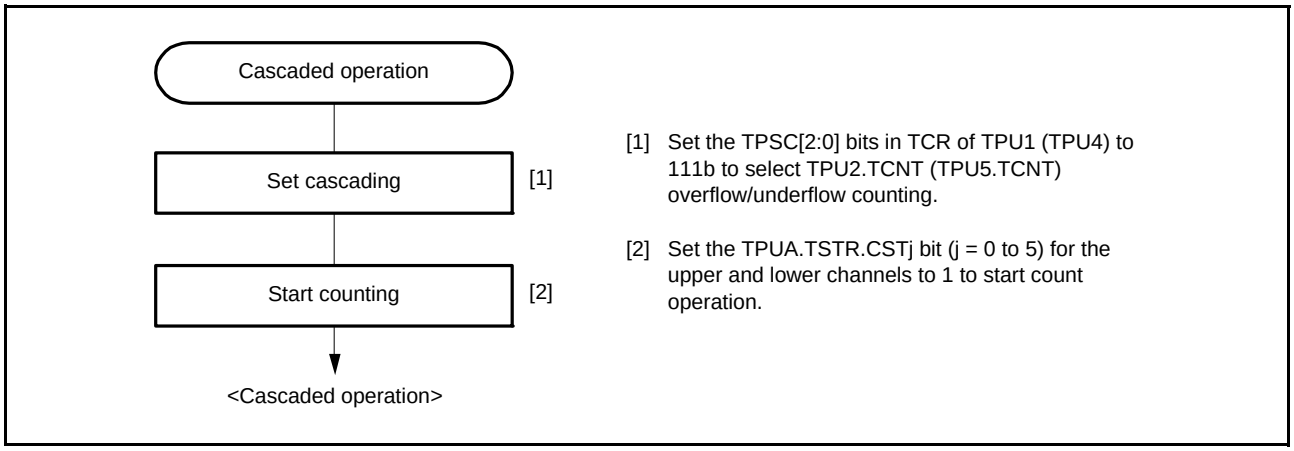


Figure 26.17 Cascaded Operation Setting Procedure

(2) Examples of Cascaded Operation

Figure 26.18 shows the operation when counting upon TPU2.TCNT overflow/underflow has been set for TPU1.TCNT, TPU1.TGRA and TPU2.TGRA have been set as input capture registers, and the rising edge of the TIOCA1 and TIOCA2 pins has been selected.

When a rising edge is input to the TIOCA1 and TIOCA2 pins simultaneously, the upper 16 bits of the 32-bit data are transferred to TPU1.TGRA, and the lower 16 bits to TPU2.TGRA. Note that a point for caution applies to simultaneous input capture in cascade operation, as described in section 26.10.11, TCNT Simultaneous Input Capture in Cascade Operation.

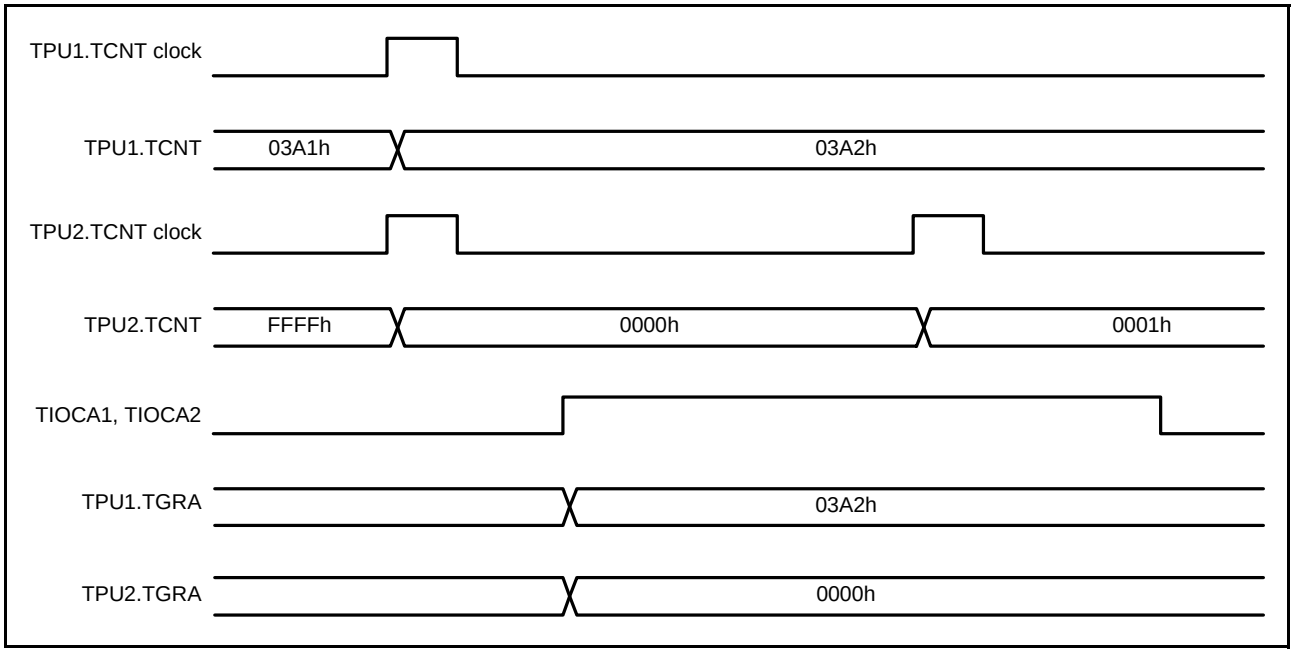


Figure 26.18 Example of Cascaded Operation (1)

Figure 26.19 shows the operation when counting upon TPU2.TCNT overflow/underflow has been set for TPU1.TCNT, and phase counting mode 1 has been specified for TPU2.

TPU1.TCNT is incremented by TPU2.TCNT overflow and decremented by TPU2.TCNT underflow.

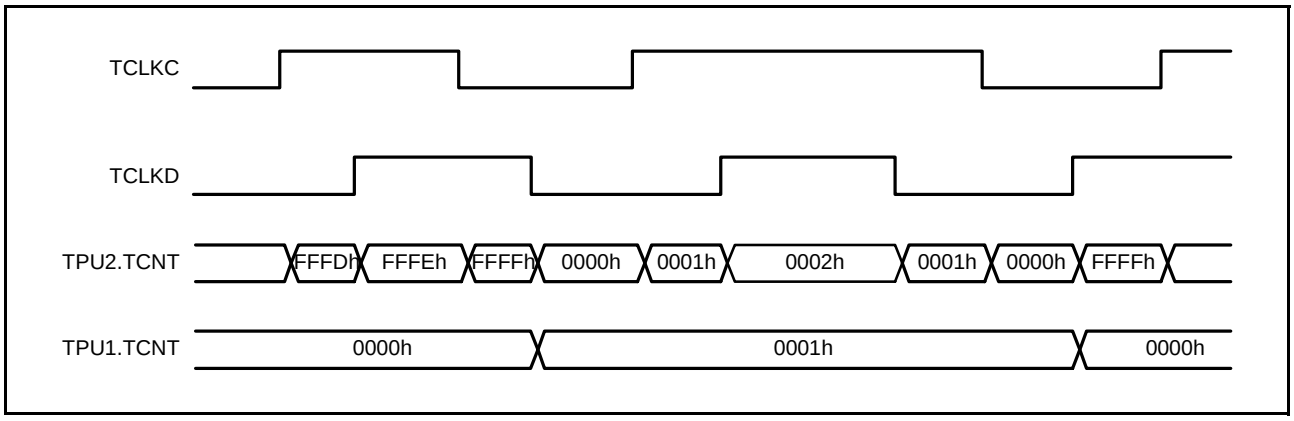


Figure 26.19 Example of Cascaded Operation (2)

26.3.5 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 26.23.

Table 26.23 PWM Output Registers and Output Pins

Channel	Register	Output Pin	
		PWM Mode 1	PWM Mode 2
TPU0	TPU0.TGRA	TIOCA0	TIOCA0
	TPU0.TGRB		TIOCB0
	TPU0.TGRC	TIOCC0	TIOCC0
	TPU0.TGRD		TIOCD0
TPU1	TPU1.TGRA	TIOCA1	TIOCA1
	TPU1.TGRB		TIOCB1
TPU2	TPU2.TGRA	TIOCA2	TIOCA2
	TPU2.TGRB		TIOCB2
TPU3	TPU3.TGRA	TIOCA3	TIOCA3
	TPU3.TGRB		TIOCB3
	TPU3.TGRC	TIOCC3	TIOCC3
	TPU3.TGRD		TIOCD3
TPU4	TPU4.TGRA	TIOCA4	TIOCA4
	TPU4.TGRB		TIOCB4
TPU5	TPU5.TGRA	TIOCA5	TIOCA5
	TPU5.TGRB		TIOCB5

Note: In PWM mode 2, PWM waveform output is not possible for the TPUm.TGRy register in which the cycle is set.

(1) Example of PWM Mode Setting Procedure

Figure 26.20 shows an example of the PWM mode setting procedure.

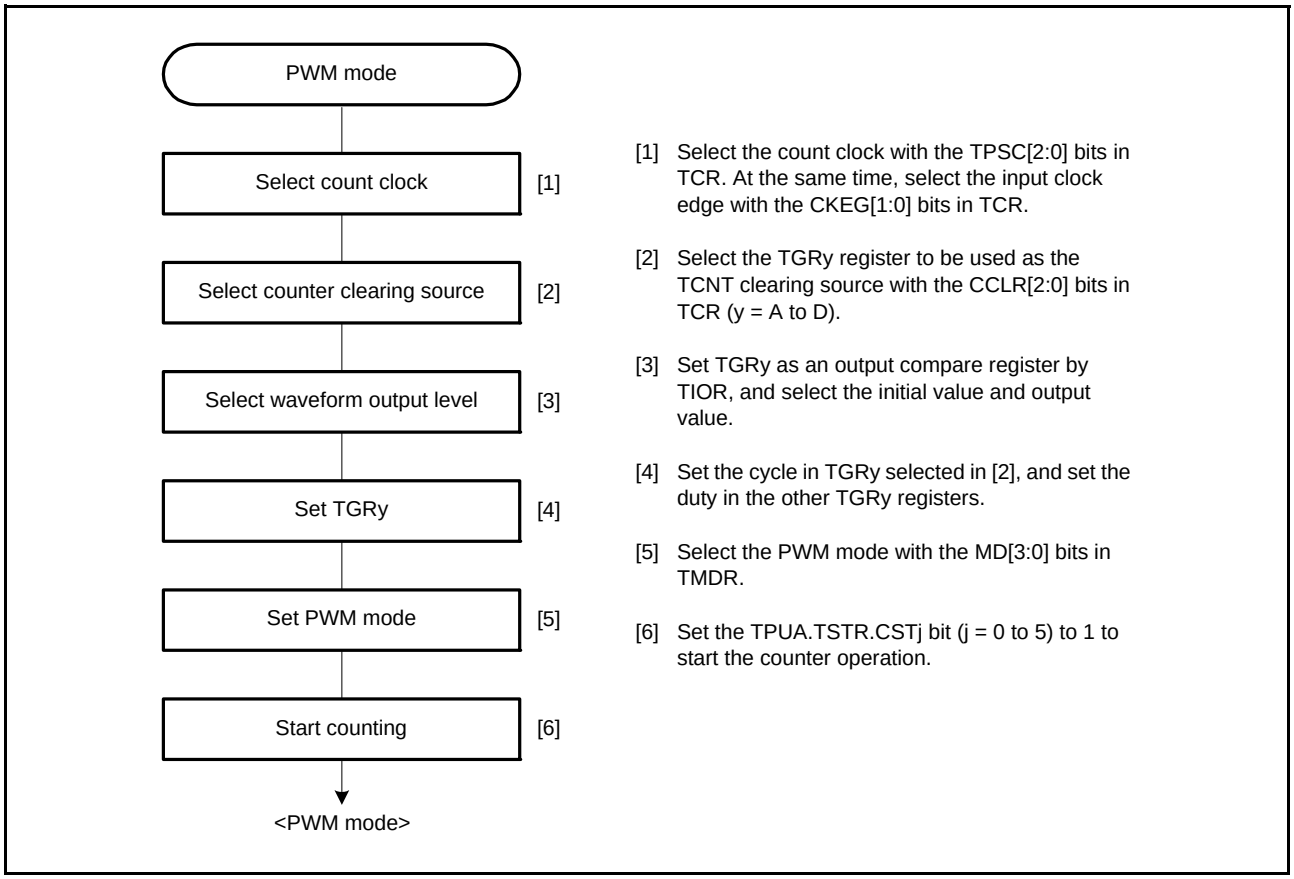


Figure 26.20 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 26.21 shows an example of PWM mode 1 operation.
In this example, TPUm.TGRA compare match is set as the TPUm.TCNT clearing source, low is set for the TGRA initial output value and output value, and high is set as the TPUm.TGRB output value.
In this case, the value set in TGRA is used as the cycle, and the value set in TGRB is used as the duty cycle.

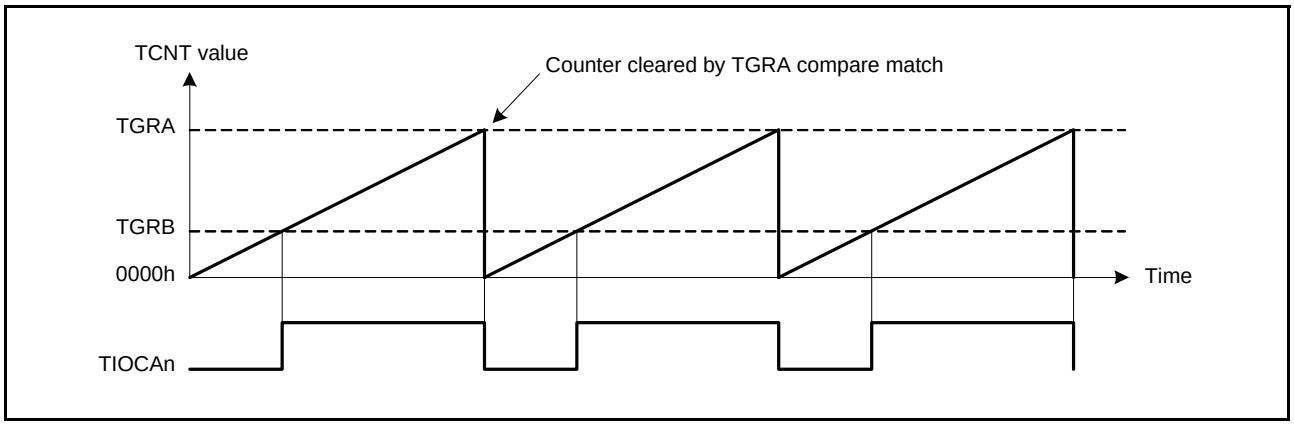


Figure 26.21 Example of PWM Mode Operation (1) (n = 0 to 5)

Figure 26.22 shows an example of PWM mode 2 operation.
In this example, synchronous operation is specified for TPU0 and TPU1, TPU1.TGRB compare match is set as the TPUm.TCNT clearing source, and low is set for the initial output value and high for the output value of the other TPUm.TGRy registers (TPU0.TGRA to TPU0.TGRD and TPU1.TGRA), to output a 5-phase PWM waveform.
In this case, the value set in TPU1.TGRB is used as the cycle, and the values set in the other TGRy registers are used as the duty cycle.

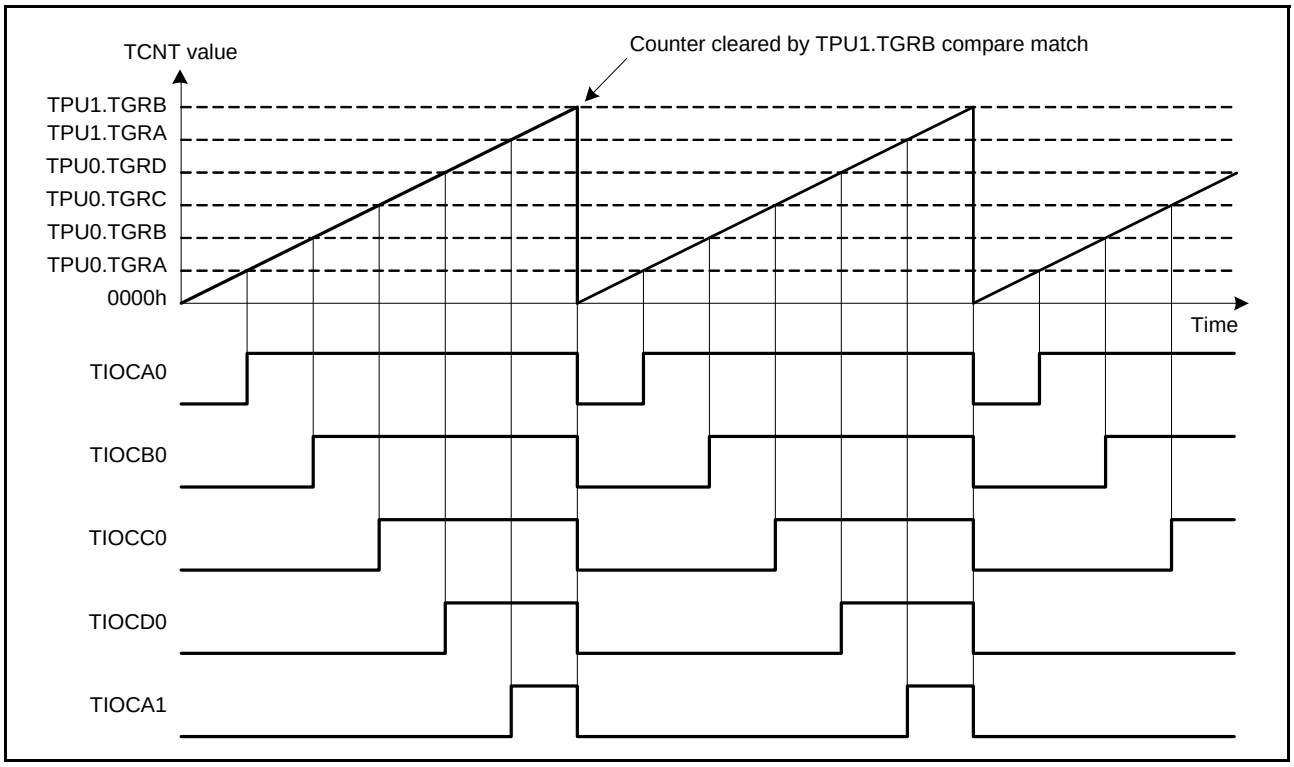


Figure 26.22 Example of PWM Mode Operation (2)

Figure 26.23 shows examples of PWM waveform output with 0% duty cycle and 100% duty cycle in PWM mode.

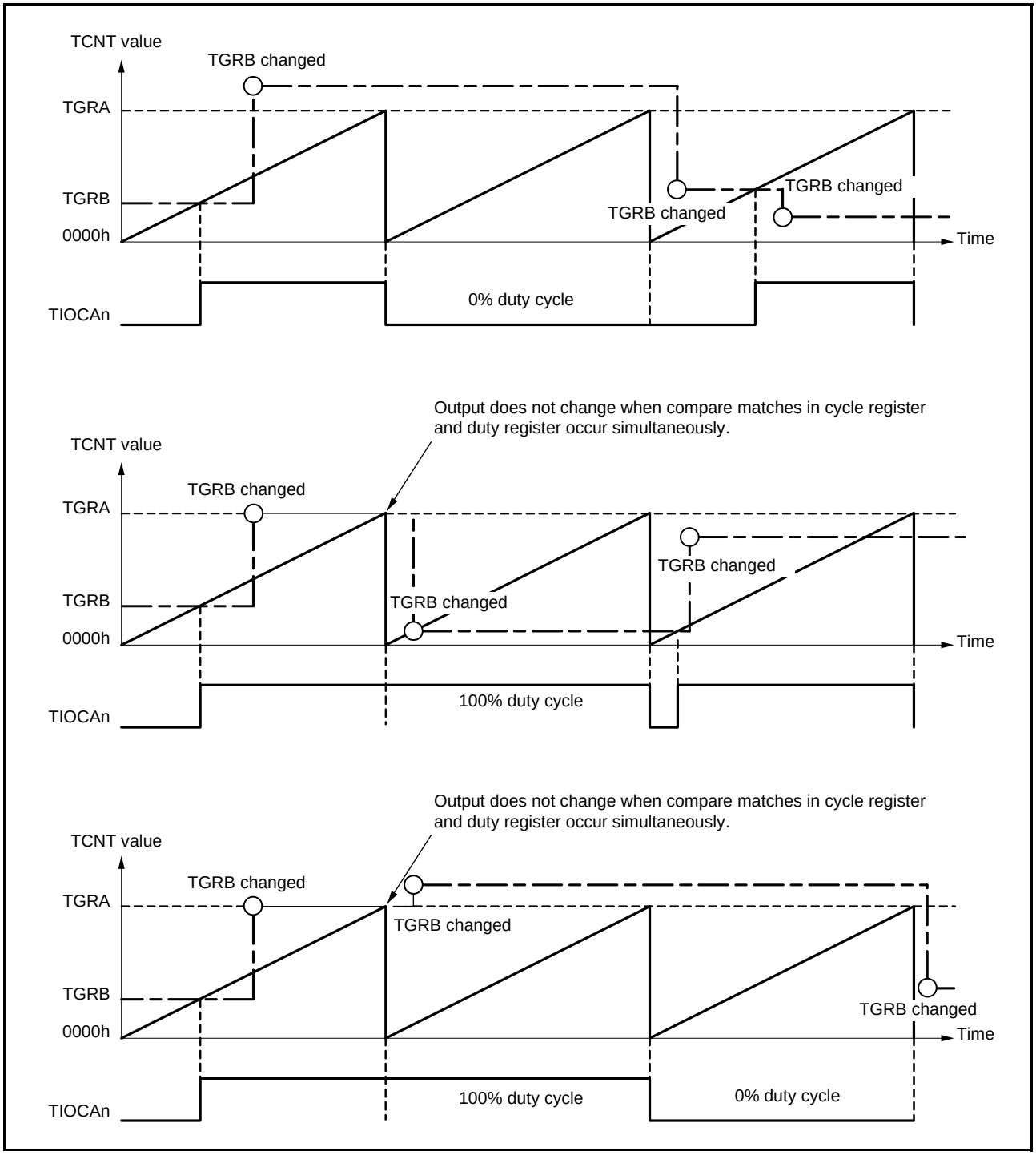


Figure 26.23 Example of PWM Mode Operation (3) (n = 0 to 5)

26.3.6 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected by the settings for channels 1, 2, 4, and 5, and TPUm.TCNT is incremented/decremented accordingly.
When phase counting mode is set, an external clock is selected as the count clock and TCNT operates as an up-/down-counter regardless of the setting of the TPSC[2:0] bits and CKEG[1:0] bits in TPUm.TCR. However, the lower 2 bits of the CCLR[2:0] bits in TPUm.TCR and the functions of TPUm.TIORH, TPUm.TIORL, TPUm.TIOR, TPUm.TIER, and TPUm.TGRy are valid, and therefore input capture/compare match and interrupt functions are available.
When an overflow occurs while TCNT is counting up, a TCIV interrupt request is generated; when an underflow occurs while TCNT is counting down, a TCIU interrupt request is generated. The TCFD bit in TPUm.TSR is the count direction flag. Reading the TCFD flag provides an indication of whether TCNT is counting up or down.
In phase counting mode, the external clock pins TCLKA, TCLKB, TCLKC, and TCLKD can be used as 2-phase encoder pulse input.
Table 26.24 lists the correspondence between external clock pins and channels.

Table 26.24 Clock Input Pins in Phase Counting Mode

	External Clock Pins	
Channel	**A-Phase**	**B-Phase**
When TPU1 or TPU5 is set to phase counting mode	TCLKA	TCLKB
When TPU2 or TPU4 is set to phase counting mode	TCLKC	TCLKD

(1) Example of Phase Counting Mode Setting Procedure

Figure 26.24 shows an example of the phase counting mode setting procedure.

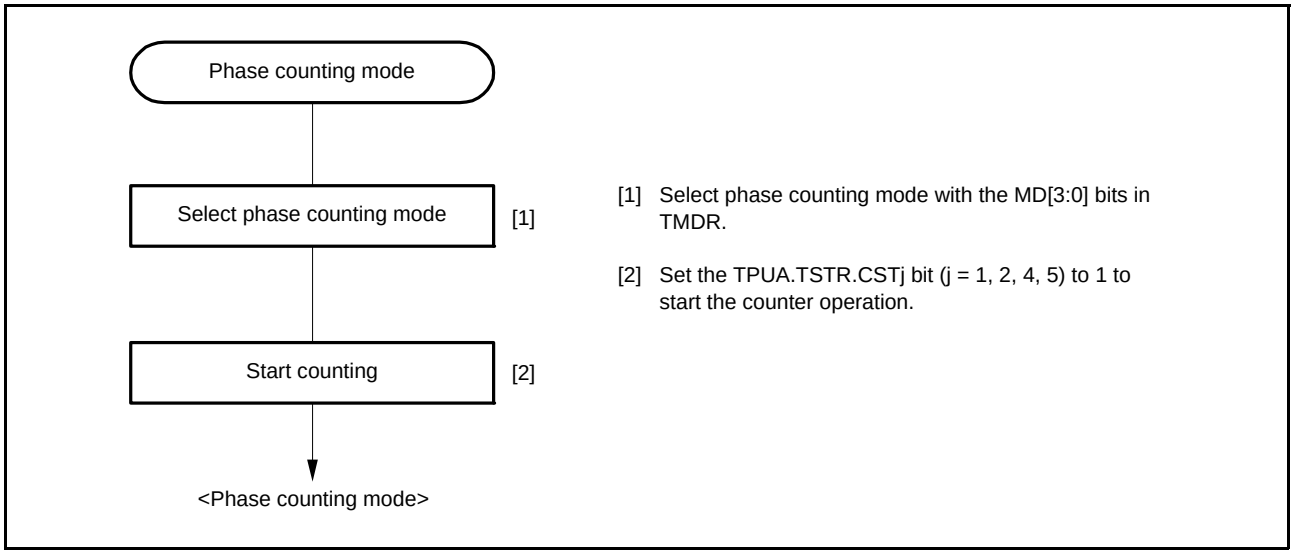


Figure 26.24 Example of Phase Counting Mode Setting Procedure

(2) Examples of Phase Counting Mode Operation

In phase counting mode, TPUm.TCNT counts up or down according to the phase difference between two external clocks. There are four modes, according to the count conditions.

(a) Phase counting mode 1

Figure 26.25 shows an example of phase counting mode 1 operation, and Table 26.25 lists the TPUm.TCNT up-/down-count conditions.

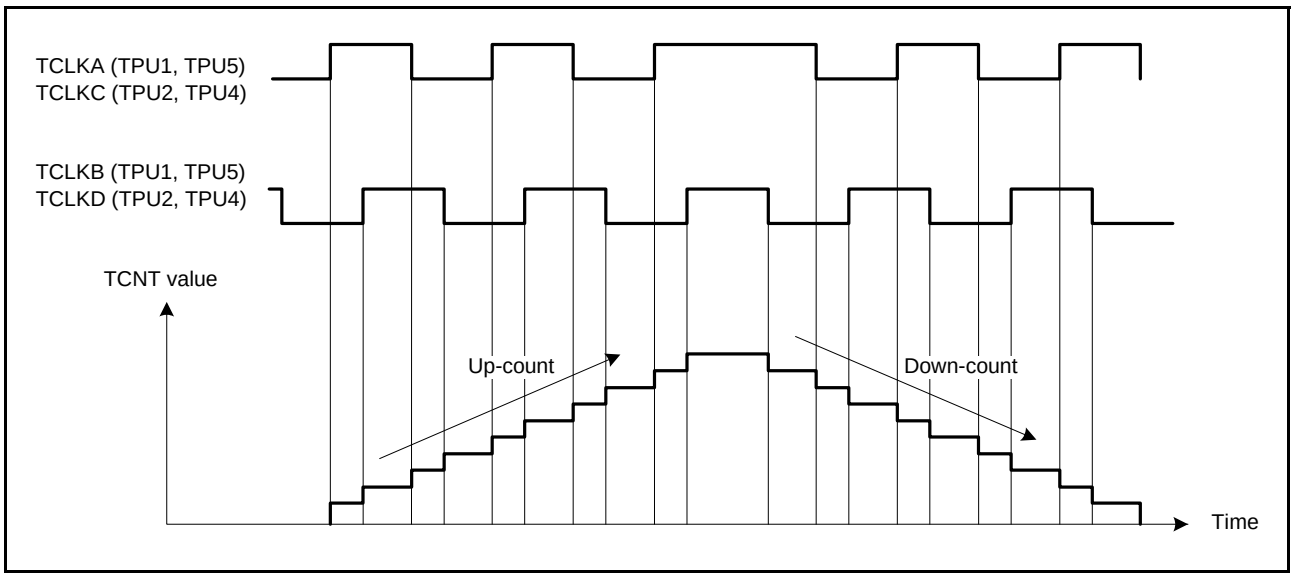


Figure 26.25 Example of Phase Counting Mode 1 Operation

Table 26.25 Up-/Down-Count Conditions in Phase Counting Mode 1

TCLKA (TPU1, TPU5) TCLKC (TPU2, TPU4)	TCLKB (TPU1, TPU5) TCLKD (TPU2, TPU4)	Operation
High	↑	Up-count
Low	↓	
↑	Low	
↓	High	
High	↓	Down-count
Low	↑	
↑	High	
↓	Low	

↑ : Rising edge

↓ : Falling edge

(b) Phase counting mode 2

Figure 26.26 shows an example of phase counting mode 2 operation, and Table 26.26 lists the TPUm.TCNT up-/down-count conditions.

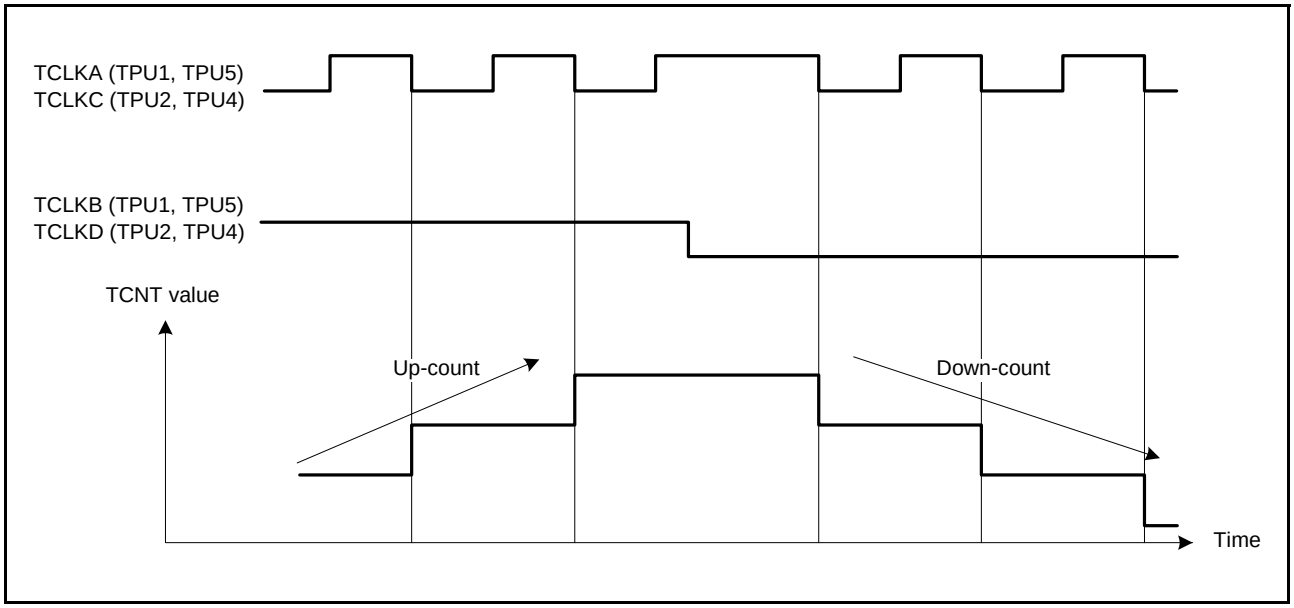


Figure 26.26 Example of Phase Counting Mode 2 Operation

Table 26.26 Up-/Down-Count Conditions in Phase Counting Mode 2

TCLKA (TPU1, TPU5) TCLKC (TPU2, TPU4)	TCLKB (TPU1, TPU5) TCLKD (TPU2, TPU4)	Operation
High	↑	Don't care
Low	↓	Don't care
↑	Low	Don't care
↓	High	Up-count
High	↓	Don't care
Low	↑	Don't care
↑	High	Don't care
↓	Low	Down-count

↑ : Rising edge

↓ : Falling edge

(c) Phase counting mode 3

Figure 26.27 shows an example of phase counting mode 3 operation, and Table 26.27 lists the TPUm.TCNT up-/down-count conditions.

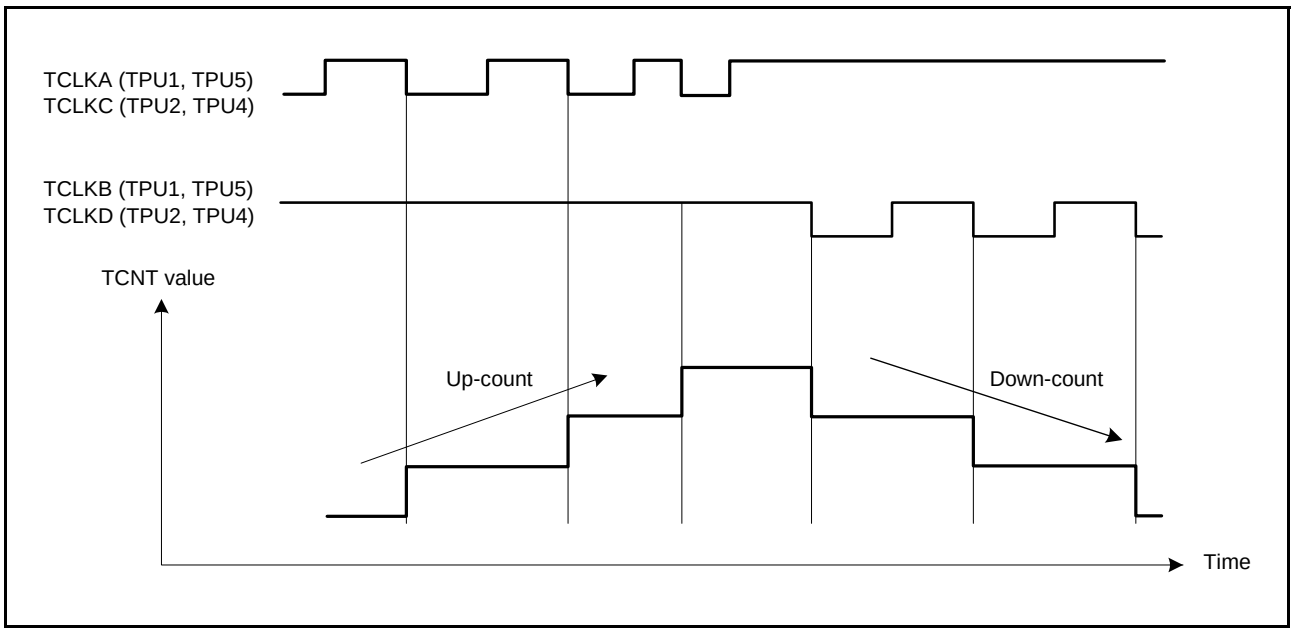


Figure 26.27 Example of Phase Counting Mode 3 Operation

Table 26.27 Up-/Down-Count Conditions in Phase Counting Mode 3

TCLKA (TPU1, TPU5) TCLKC (TPU2, TPU4)	TCLKB (TPU1, TPU5) TCLKD (TPU2, TPU4)	Operation
High	↑	Don't care
Low	↓	Don't care
↑	Low	Don't care
↓	High	Up-count
High	↓	Down-count
Low	↑	Don't care
↑	High	Don't care
↓	Low	Don't care

↑ : Rising edge

↓ : Falling edge

(d) Phase counting mode 4

Figure 26.28 shows an example of phase counting mode 4 operation, and Table 26.28 lists the TPUm.TCNT up-/down-count conditions.

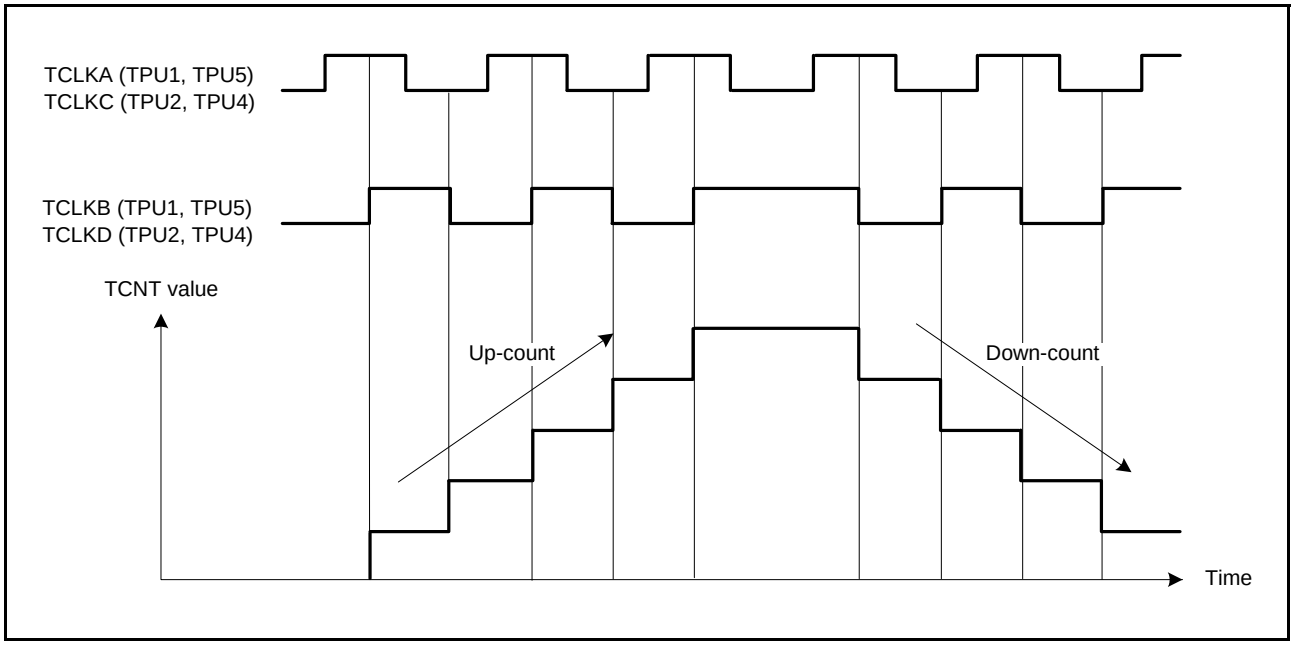


Figure 26.28 Example of Phase Counting Mode 4 Operation

Table 26.28 Up-/Down-Count Conditions in Phase Counting Mode 4

TCLKA (TPU1, TPU5) TCLKC (TPU2, TPU4)	TCLKB (TPU1, TPU5) TCLKD (TPU2, TPU4)	Operation
High	↑	Up-count
Low	↓	
↑	Low	Don't care
↓	High	
High	↓	Down-count
Low	↑	
↑	High	Don't care
↓	Low	

↑ : Rising edge

↓ : Falling edge

26.3.6.1 Phase Counting Mode Application Example

Figure 26.29 shows an example in which phase counting mode is set for TPU1, and TPU1 is coupled with TPU0 to input servo motor 2-phase encoder pulses in order to detect the position or speed.

TPU1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to the TCLKA and TCLKB pins.

TPU0 operates with TPU0.TCNT clearing by TPU0.TGRC compare match; TPU0.TGRA and TPU0.TGRC are used for the compare match function and are set with the speed control cycle and position control cycle. TPU0.TGRB is used for input capture, with TPU0.TGRB and TPU0.TGRD operating in buffer mode. The TPU1 count clock is specified as the TPU0.TGRB input capture source, and the pulse width of 2-phase encoder 4-multiplication pulses is detected.

TPU1.TGRA and TPU1.TGRB for TPU1 are specified for input capture, TPU0.TGRA and TPU0.TGRC compare matches are selected as the input capture source, and the up-/down-counter values for the control cycles are stored.

This procedure enables accurate position/speed detection to be achieved.

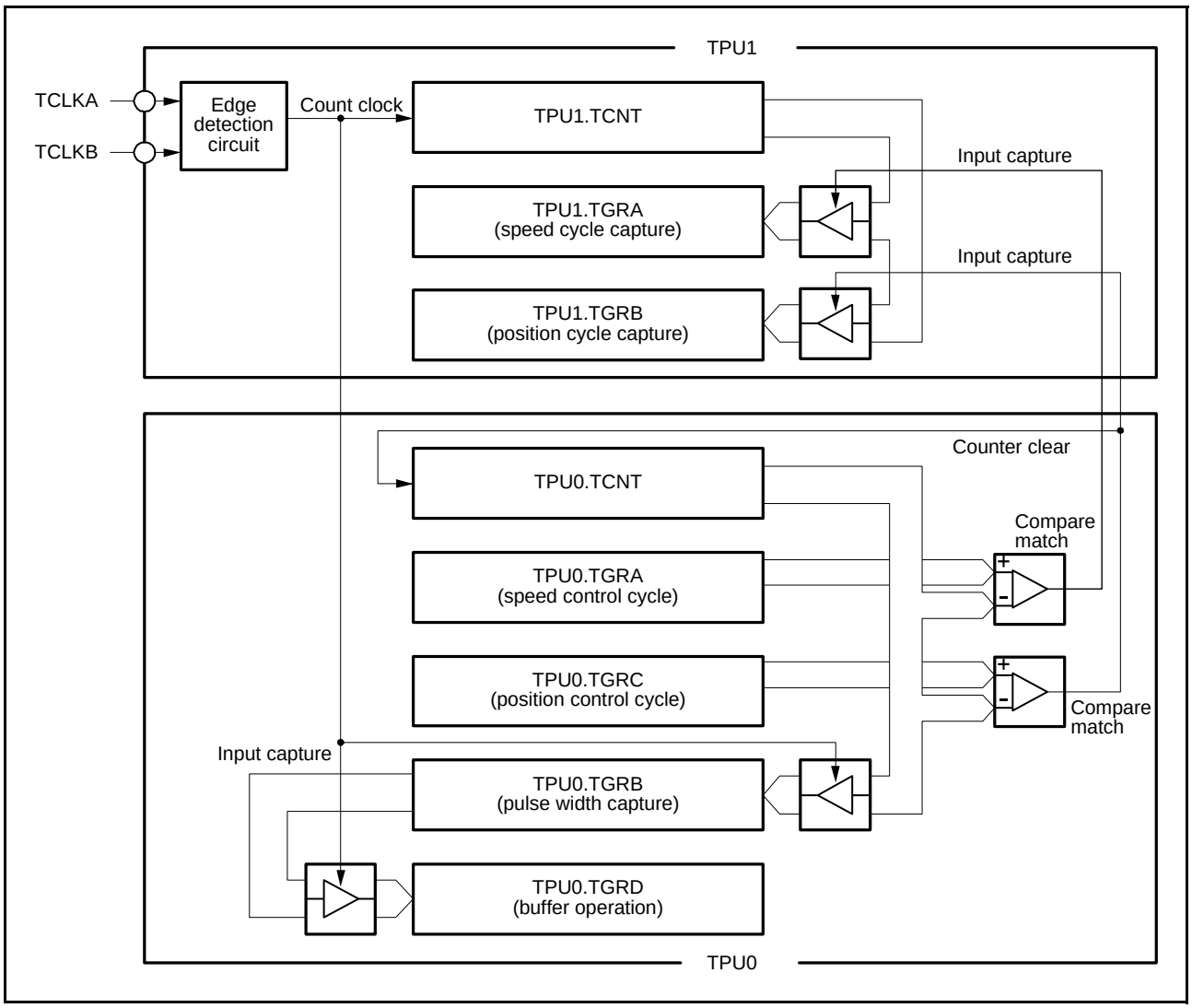


Figure 26.29 Phase Counting Mode Application Example

26.3.7 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 26.30 is a timing chart for the noise filter.

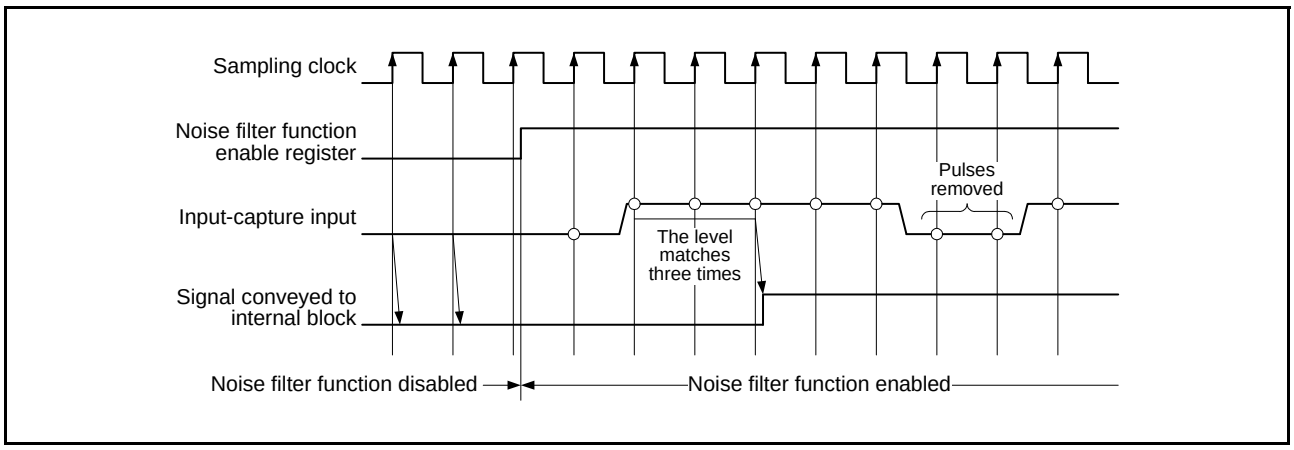


Figure 26.30 Timing Chart for the Noise Filter

If noise filtering is set, input capture operation is performed on the edges of noise-filtered signal after a minimum delay of (sampling interval × 2 + PCLK) due to noise filtering for the input capture input.

26.4 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 (ICUE).

Table 26.29 lists the TPU interrupt sources.

Table 26.29 TPU Interrupt Sources

Channel	Name	Interrupt Source	DTC Activation	DMAC Activation
TPU0	TGI0A	TPU0.TGRA input capture/compare match	Possible	Possible
	TGI0B	TPU0.TGRB input capture/compare match	Possible	Not possible
	TGI0C	TPU0.TGRC input capture/compare match	Possible	Not possible
	TGI0D	TPU0.TGRD input capture/compare match	Possible	Not possible
	TCI0V	TPU0.TCNT overflow	Not possible	Not possible
TPU1	TGI1A	TPU1.TGRA input capture/compare match	Possible	Possible
	TGI1B	TPU1.TGRB input capture/compare match	Possible	Not possible
	TCI1V	TPU1.TCNT overflow	Not possible	Not possible
	TCI1U	TPU1.TCNT underflow	Not possible	Not possible
TPU2	TGI2A	TPU2.TGRA input capture/compare match	Possible	Possible
	TGI2B	TPU2.TGRB input capture/compare match	Possible	Not possible
	TCI2V	TPU2.TCNT overflow	Not possible	Not possible
	TCI2U	TPU2.TCNT underflow	Not possible	Not possible
TPU3	TGI3A	TPU3.TGRA input capture/compare match	Possible	Possible
	TGI3B	TPU3.TGRB input capture/compare match	Possible	Not possible
	TGI3C	TPU3.TGRC input capture/compare match	Possible	Not possible
	TGI3D	TPU3.TGRD input capture/compare match	Possible	Not possible
	TCI3V	TPU3.TCNT overflow	Not possible	Not possible
TPU4	TGI4A	TPU4.TGRA input capture/compare match	Possible	Possible
	TGI4B	TPU4.TGRB input capture/compare match	Possible	Not possible
	TCI4V	TPU4.TCNT overflow	Not possible	Not possible
	TCI4U	TPU4.TCNT underflow	Not possible	Not possible
TPU5	TGI5A	TPU5.TGRA input capture/compare match	Possible	Possible
	TGI5B	TPU5.TGRB input capture/compare match	Possible	Not possible
	TCI5V	TPU5.TCNT overflow	Not possible	Not possible
	TCI5U	TPU5.TCNT underflow	Not possible	Not possible

Note: This table lists the initial state immediately after a reset. The relative channel priority levels can be changed by the interrupt controller.

(1) Input Capture/Compare Match Interrupt

An interrupt is requested when the TGIEy bit (y = A, B, C, D) in TPUm.TIER is set to 1 by the occurrence of a TPUm.TGRy input capture/compare match on a channel. The TPU has 16 input capture/compare match interrupts, four each for TPU0 and TPU3, and two each for TPU1, TPU2, TPU4, and TPU5.

(2) Overflow Interrupt

An interrupt is requested when the TCIEV bit in TPUm.TIER is set to 1 by the occurrence of a TPUm.TCNT overflow on a channel. The TPU has six overflow interrupts, one for each channel.

(3) Underflow Interrupt

An interrupt is requested when the TCIEU bit in TPUm.TIER is set to 1 by the occurrence of a TPUm.TCNT underflow on a channel. The TPU has four underflow interrupts, one each for TPU1, TPU2, TPU4, and TPU5.

26.5 DTC Activation

The DTC can be activated by the TPUm.TGRy input capture/compare match interrupt of each channel. For details, see section 20, Data Transfer Controller (DTCb).
A total of 16 input capture/compare match interrupts can be used as DTC activation sources, four each for TPU0 and TPU3, and two each for TPU1, TPU2, TPU4, and TPU5.

26.6 DMAC Activation

The DMAC can be activated by the TPUm.TGRA input capture/compare match interrupt of each channel. For details, see section 18, DMA Controller (DMACAb).
A total of six TPUm.TGRA input capture/compare match interrupts can be used as DMAC activation sources, one for each channel.

26.7 A/D Converter Activation

The TPU can activate the A/D converter by the TPUm.TGRA input capture/compare match for each channel.
When the TTGE bit in TPUm.TIER is set to 1, the TPU requests the A/D converter to start A/D conversion by the occurrence of a TPUm.TGRA input capture/compare match on a particular channel.

26.8 PPG Trigger

Input capture to or compare match with TGRA and TGRB in TPU0 to TPU3 can be made to act as a PPG1 waveform trigger. For details, see section 27, Programmable Pulse Generator (PPG).

26.9 Operation Timing

26.9.1 Input/Output Timing

(1) TPUm.TCNT Count Timing

Figure 26.31 shows TPUm.TCNT count timing in internal clock operation, and Figure 26.32 shows TCNT count timing in external clock operation.

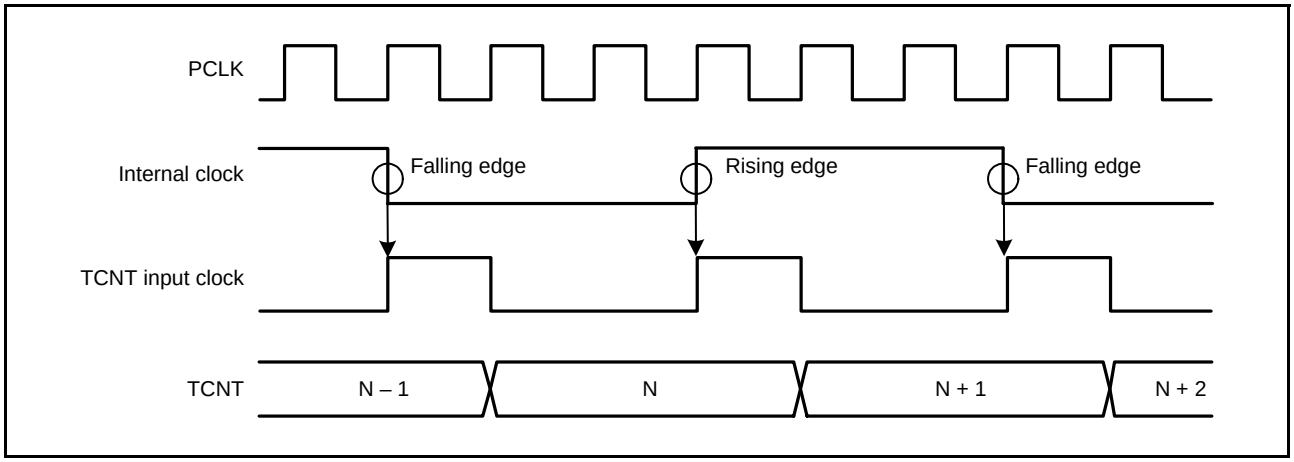


Figure 26.31 Count Timing in Internal Clock Operation

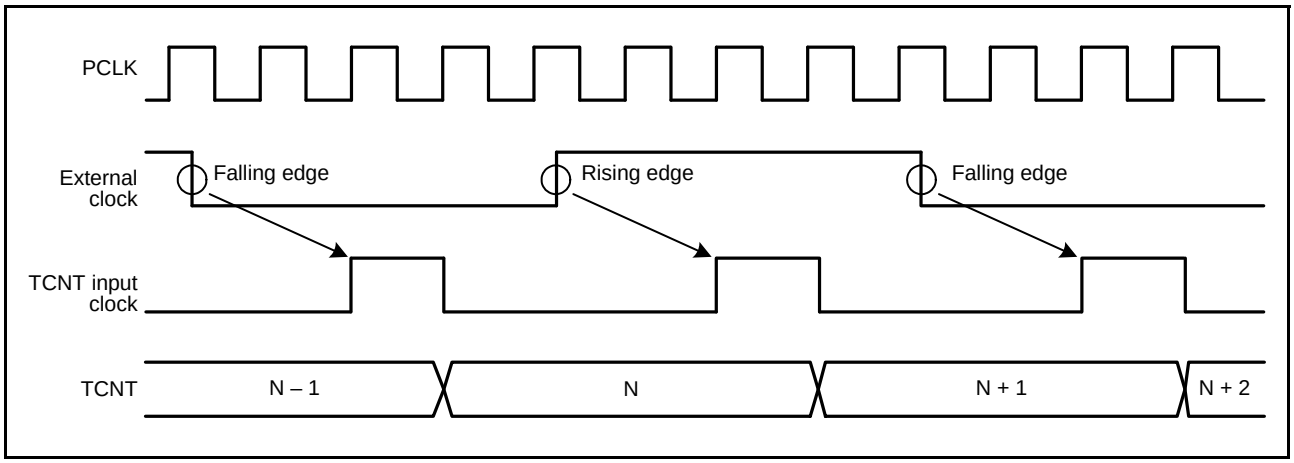


Figure 26.32 Count Timing in External Clock Operation

(2) Output Compare Output Timing

A compare match signal is generated in the final state in which TPUm.TCNT and TPUm.TGRy match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the output value set in TPUm.TIORH, TPUm.TIORL, or TPUm.TIOR is output to the output compare output pin TIOCyn (y = A to D; n = 0 to 5). After a match between TCNT and TGRy, the compare match signal is not generated until the TCNT input clock is generated.

Figure 26.33 shows output compare output timing.

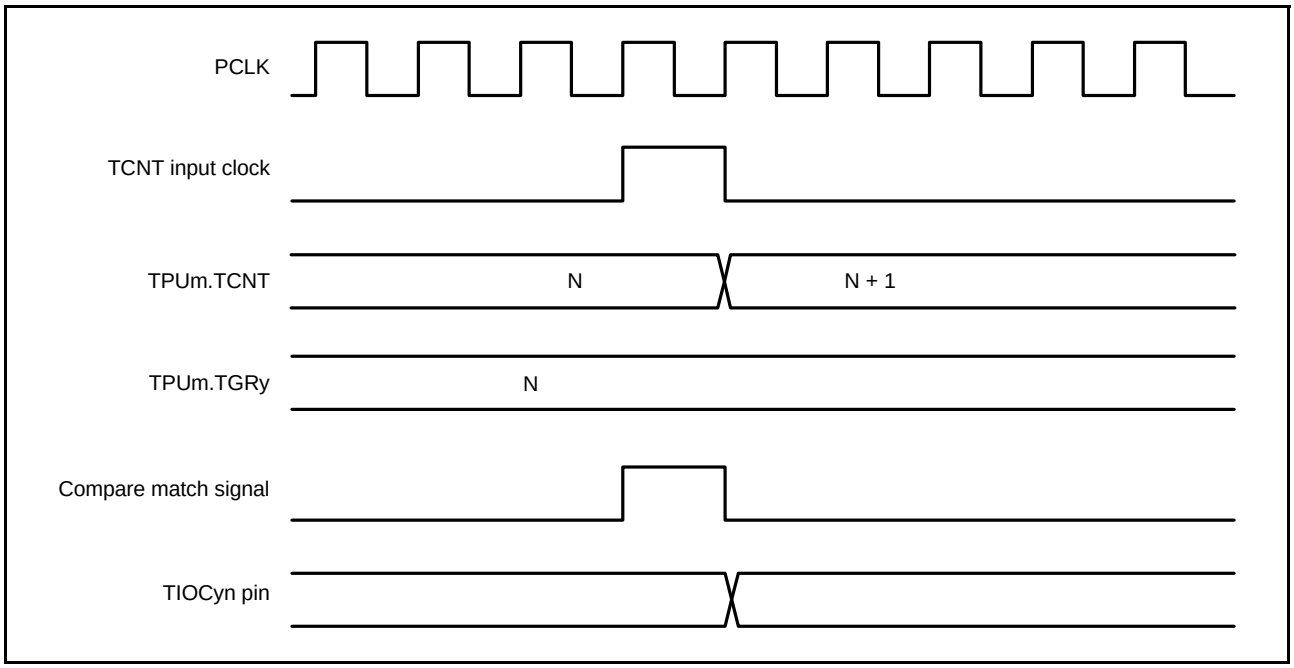


Figure 26.33 Output Compare Output Timing

(3) Input Capture Signal Timing

Figure 26.34 shows input capture signal timing.

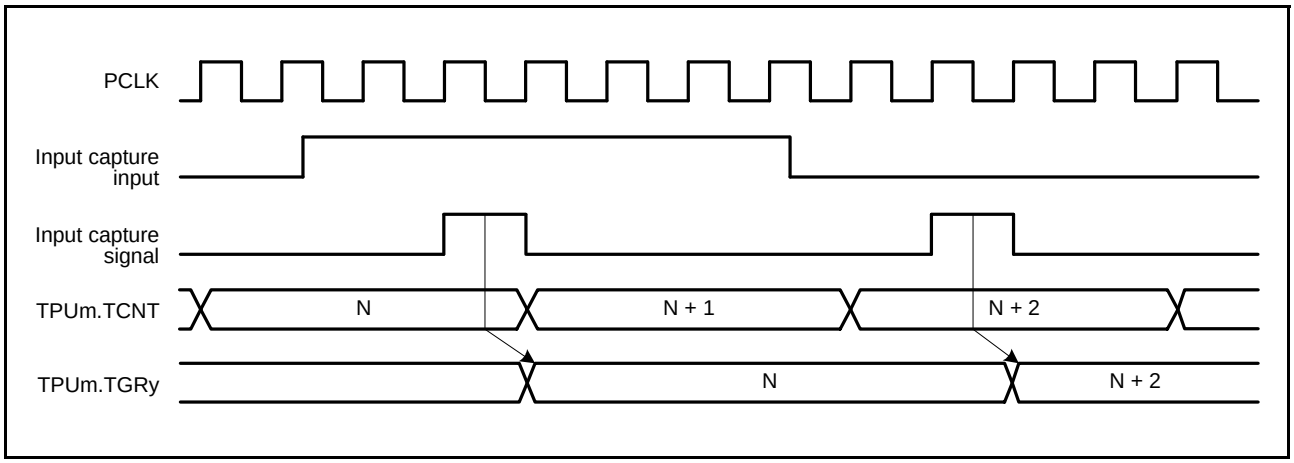


Figure 26.34 Input Capture Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figure 26.35 shows the timing when counter clearing by compare match occurrence is specified, and Figure 26.36 shows the timing when counter clearing by input capture occurrence is specified.

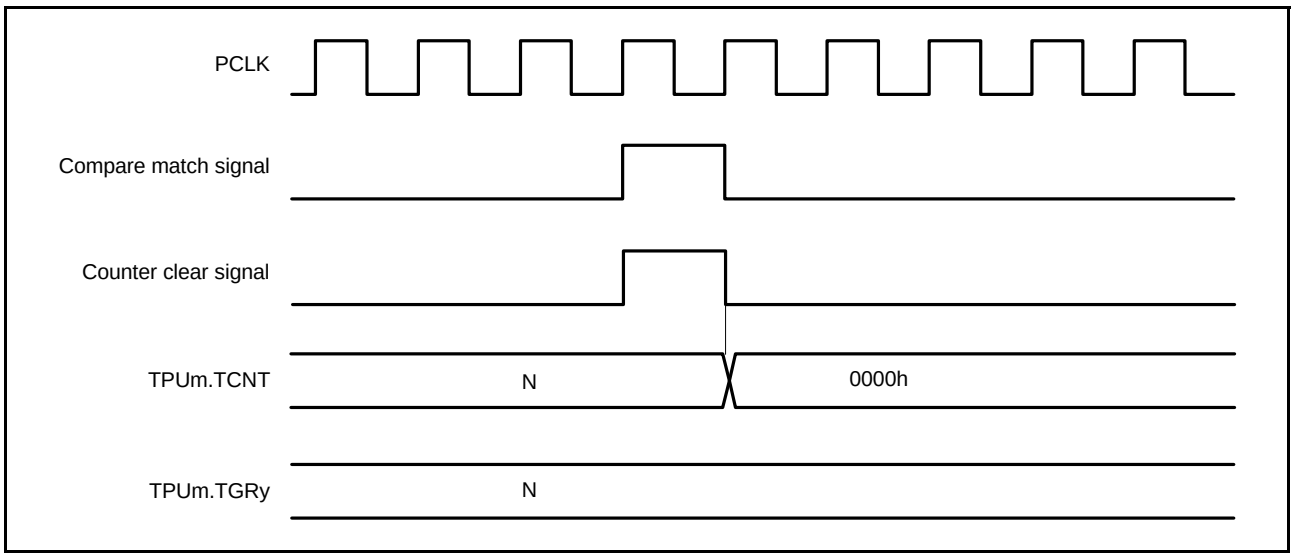


Figure 26.35 Counter Clear Timing (Compare Match)

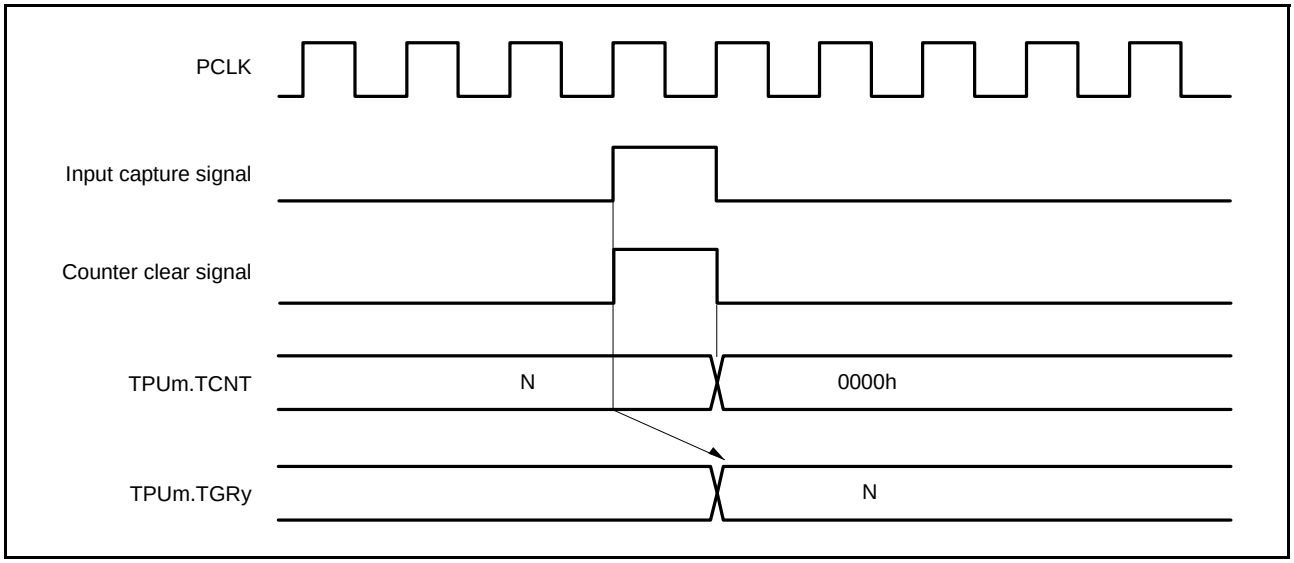


Figure 26.36 Counter Clear Timing (Input Capture)

(5) Buffer Operation Timing

Figure 26.37 and Figure 26.38 show the timings in buffer operation.

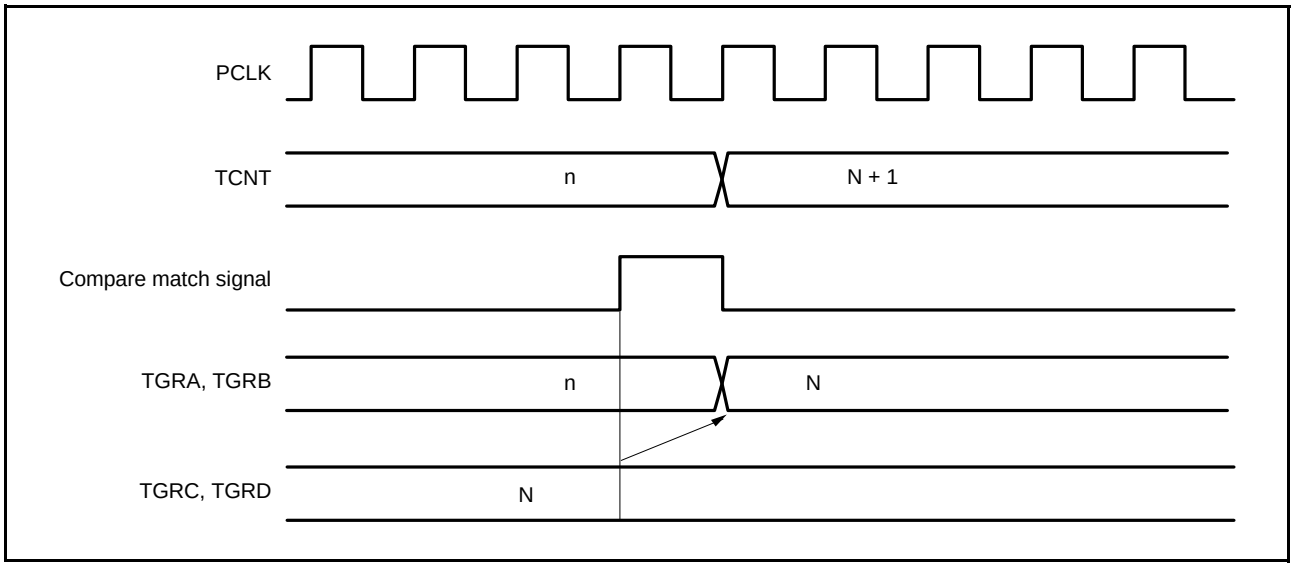


Figure 26.37 Buffer Operation Timing (Compare Match)

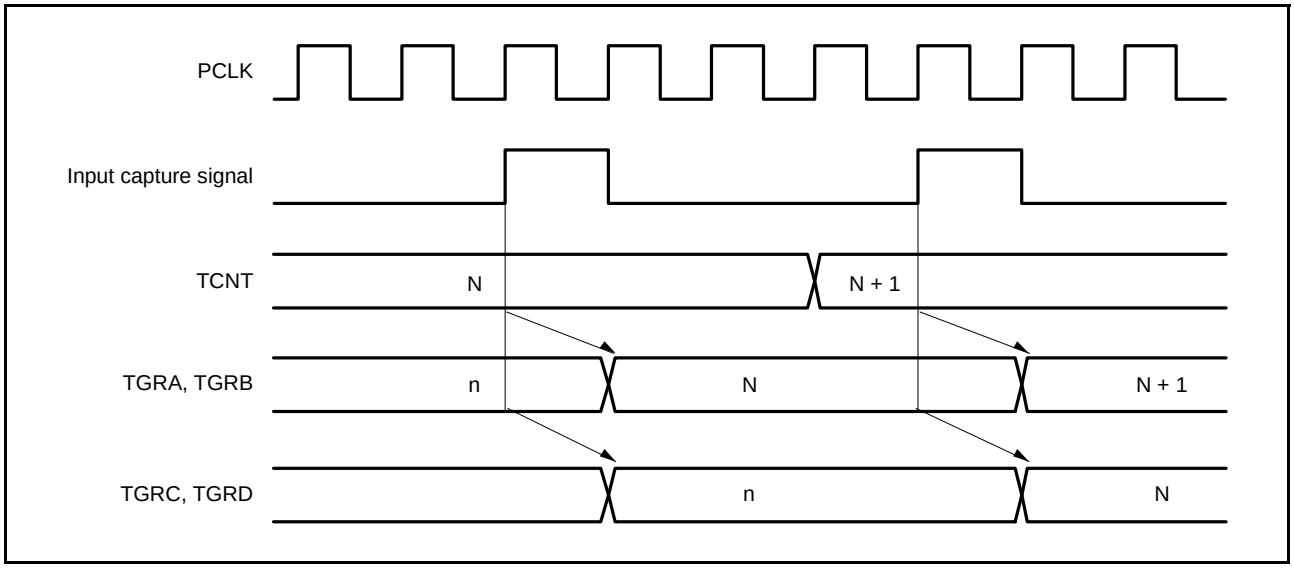


Figure 26.38 Buffer Operation Timing (Input Capture)

26.9.2 Interrupt Signal Timing

(1) Timing of Interrupt Signal Setting on Compare Match

Figure 26.39 shows the timing for setting the interrupt signal by compare match occurrence.

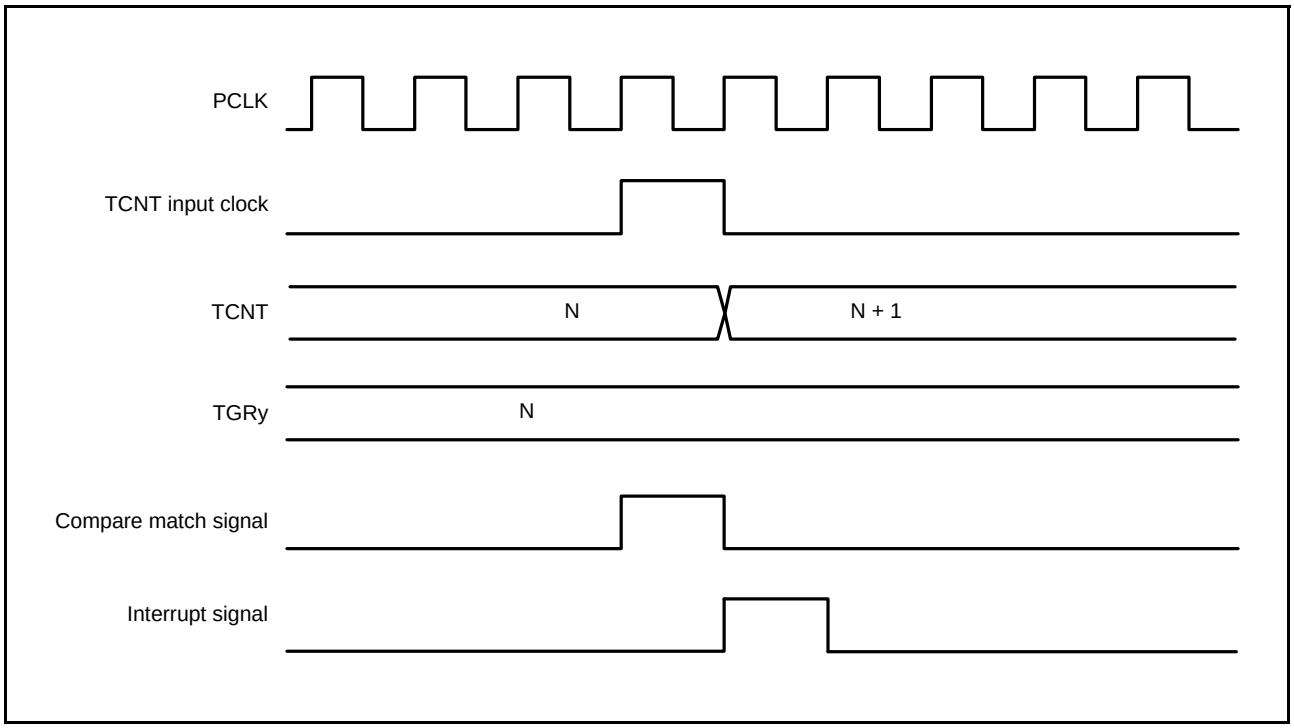


Figure 26.39 TGImy Interrupt Timing (Compare Match)

(2) Timing of Interrupt Signal Setting on Input Capture

Figure 26.40 shows the timing for setting the interrupt signal by input capture occurrence.

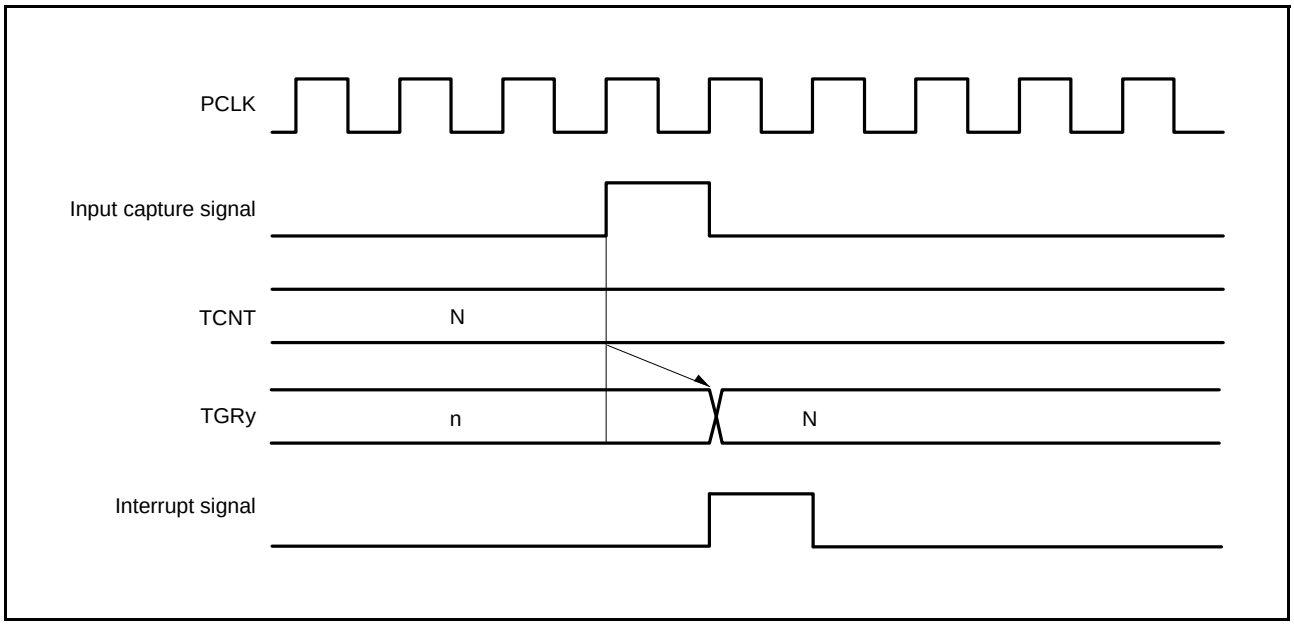


Figure 26.40 TGImy Interrupt Timing (Input Capture)

(3) Timing of TCImV/TCImU Interrupt Signal Setting

Figure 26.41 shows the timing for generating the TCImV interrupt signal by overflow occurrence.
Figure 26.42 shows the timing for generating the TCImU interrupt signal by underflow occurrence.

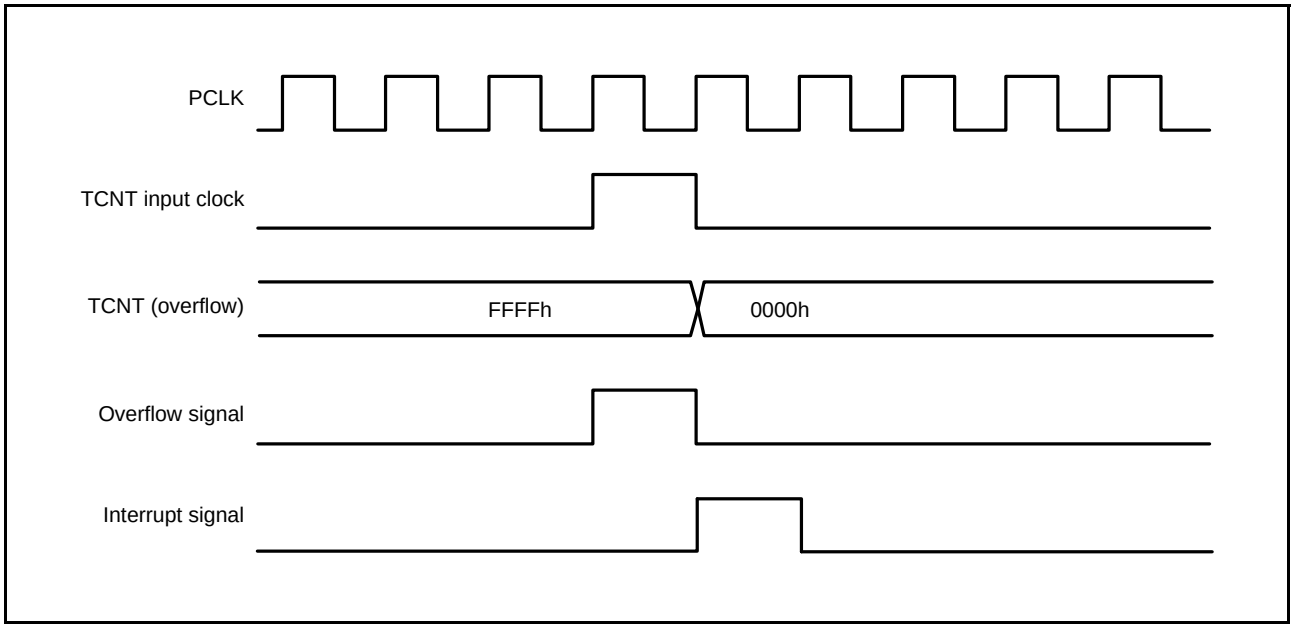


Figure 26.41 TCImV Interrupt Setting Timing

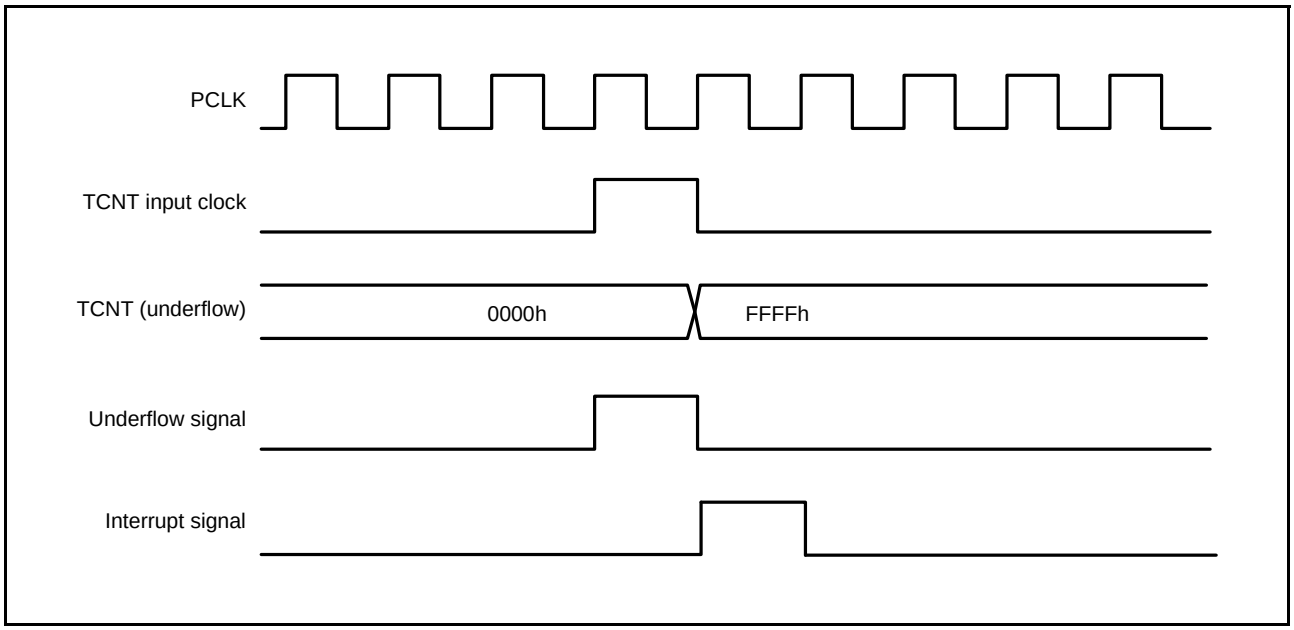


Figure 26.42 TCImU Interrupt Setting Timing

26.10 Usage Notes

26.10.1 Module Stop Function Setting

Operation of the TPU can be disabled or enabled using the module stop control register. The TPU does not operate with the initial setting. Register access is enabled by releasing the module stop state. For details, see section 11, Low Power Consumption.

26.10.2 Input Clock Restrictions

The input clock pulse width must be at least 1.5 PCLK cycles in the case of single-edge detection, and at least 2.5 PCLK cycles in the case of both-edge detection. The TPU will not operate properly with a narrower pulse width.
In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 PCLK cycles, and the pulse width must be at least 2.5 PCLK cycles. Figure 26.43 shows the input clock conditions in phase counting mode.

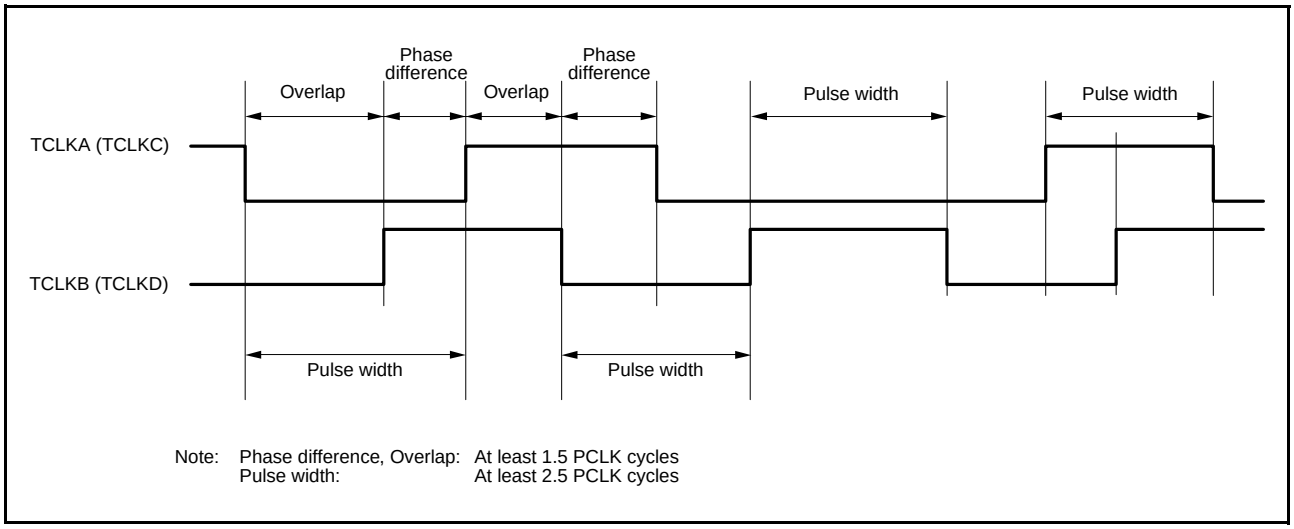


Figure 26.43 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

26.10.3 Notes on Cycle Setting

When counter clearing by compare match is set, TPUm.TCNT is cleared in the final state in which it matches the TPUm.TGRy value (the point at which the count value matched by TCNT is updated). Consequently, the actual counter frequency is given by the following formula:

$$f = \frac{f_{TCNT\_CLK}}{(N + 1)}$$

f: Counter frequency
f_{TCNT_CLK}: Count clock frequency
N: TGRy set value

26.10.4 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 26.44 shows the timing in this case.

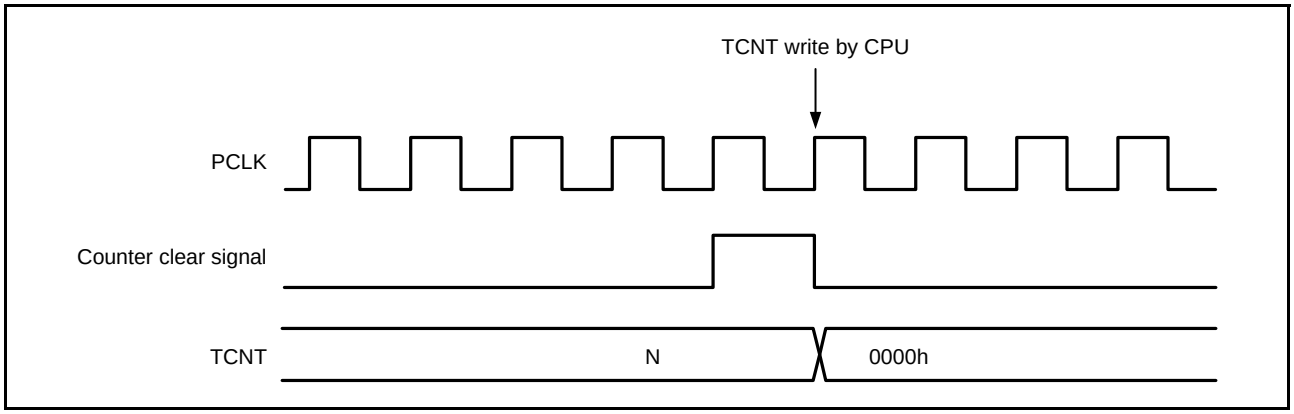


Figure 26.44 Conflict between TPUm.TCNT Write and Clear Operations

26.10.5 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 26.45 shows the timing in this case.

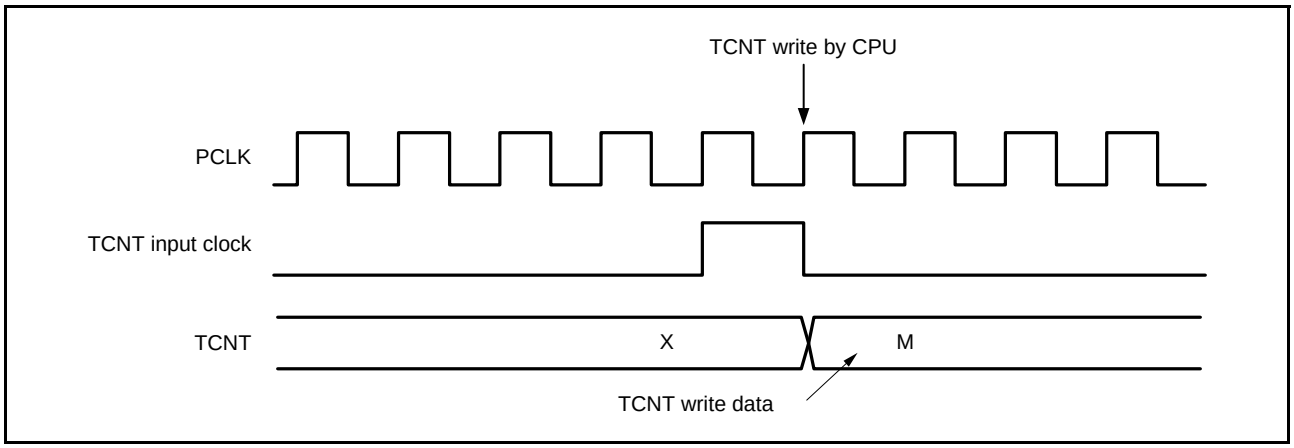


Figure 26.45 Conflict between TPUm.TCNT Write and Increment Operations

26.10.6 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 26.46 shows the timing in this case.

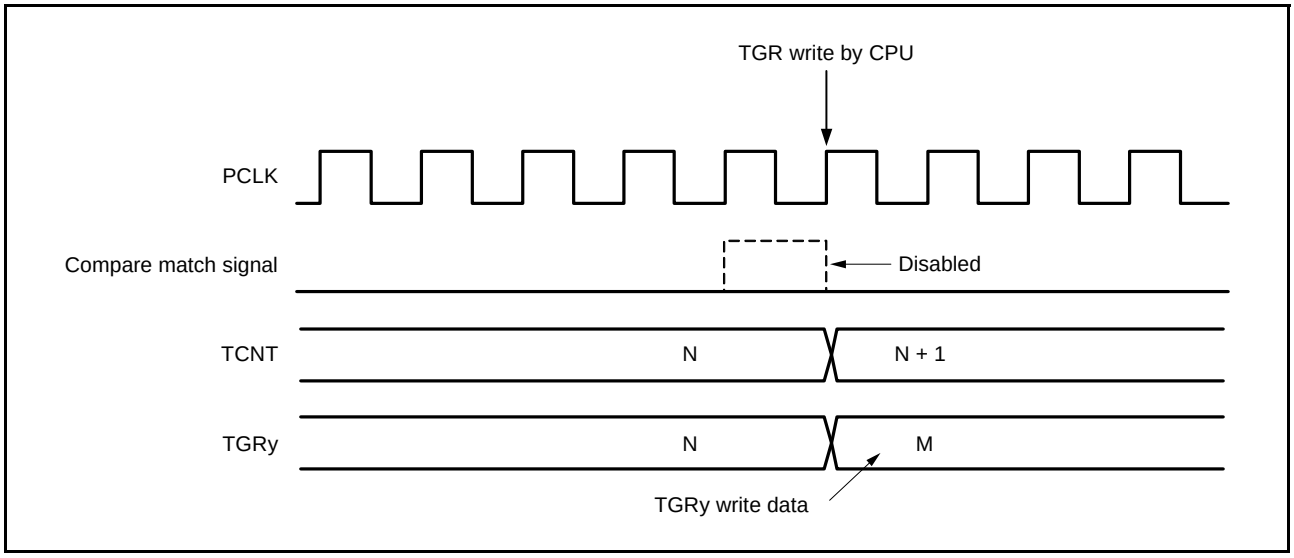


Figure 26.46 Conflict between TPUm.TGRy Write and Compare Match

26.10.7 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 26.47 shows the timing in this case.

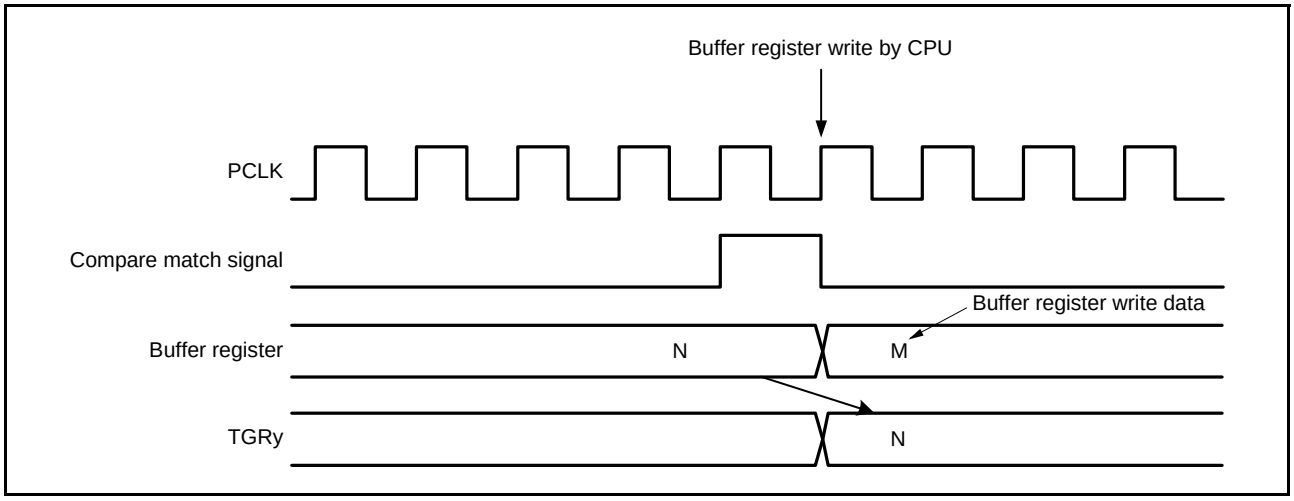


Figure 26.47 Conflict between Buffer Register Write and Compare Match

26.10.8 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 26.48 shows the timing in this case.

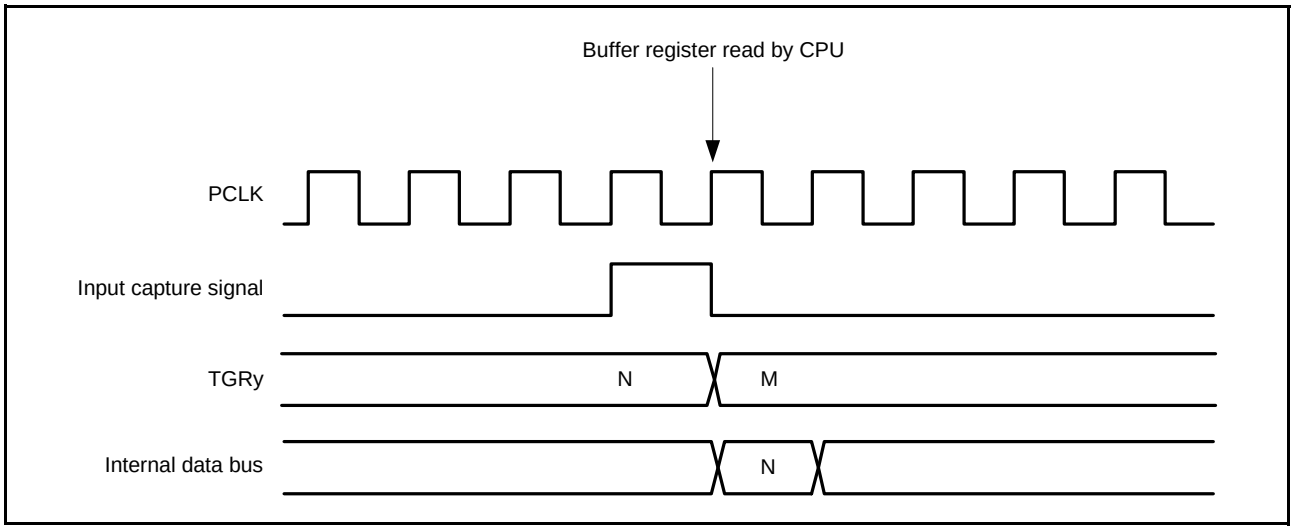


Figure 26.48 Conflict between TPUm.TGRy Read and Input Capture

26.10.9 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 26.49 shows the timing in this case.

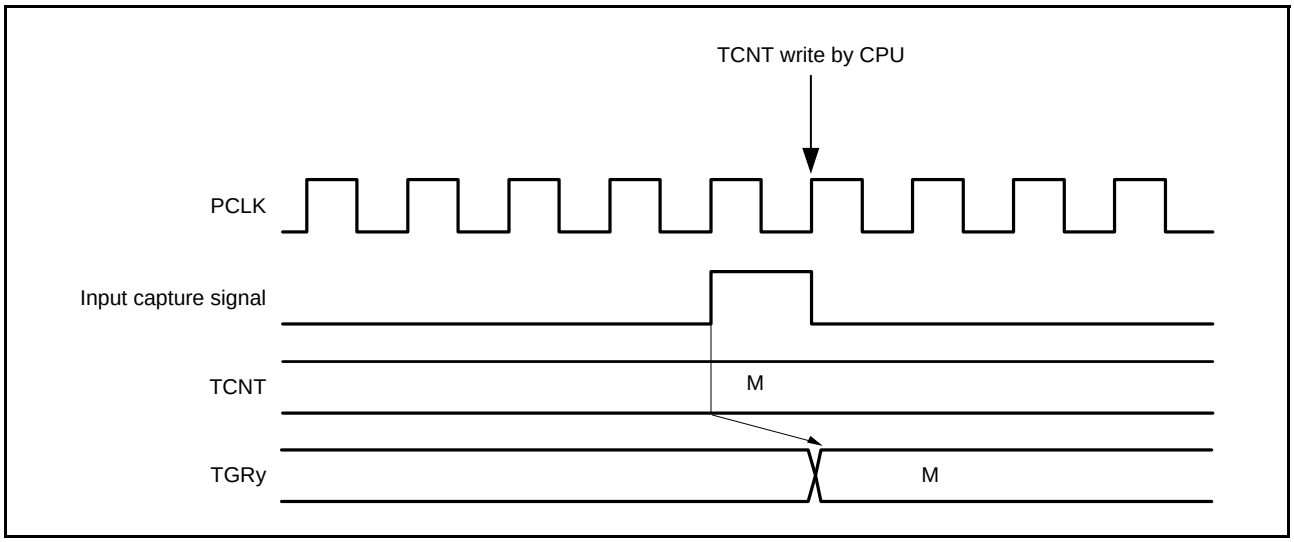


Figure 26.49 Conflict between TPUm.TGRy Write and Input Capture

26.10.10 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 26.50 shows the timing in this case.

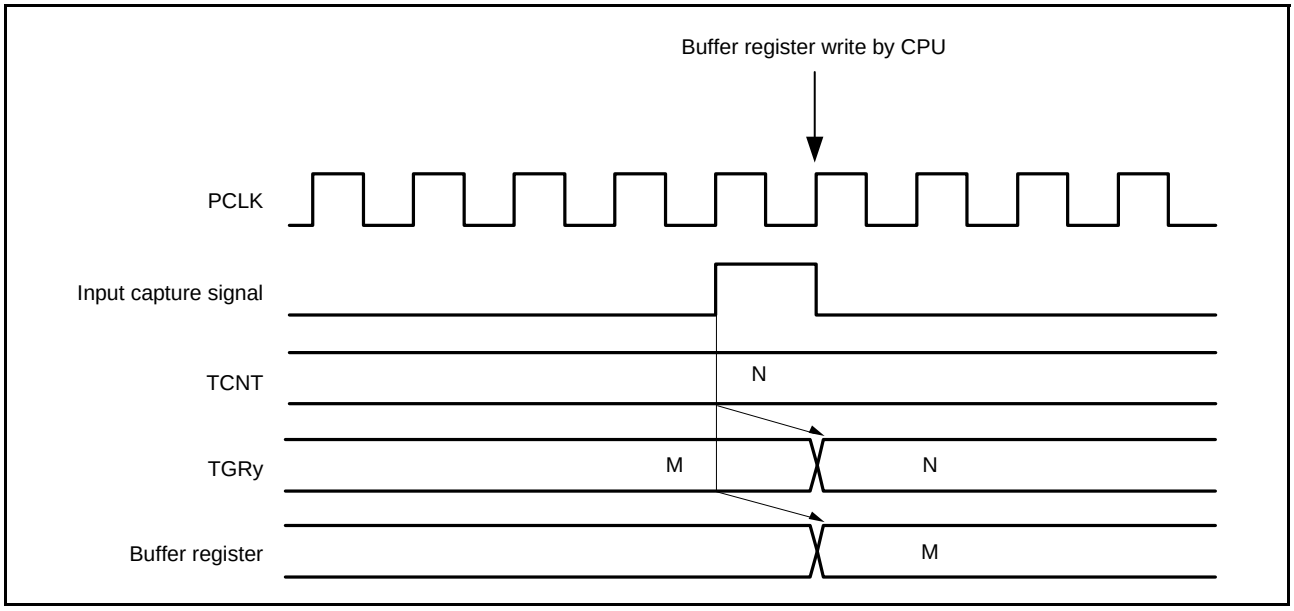


Figure 26.50 Conflict between Buffer Register Write and Input Capture

26.10.11 TCNT Simultaneous Input Capture in Cascade Operation

When TPU1.TCNT and TPU2.TCNT are cascaded for operation as a 32-bit counter, the counter value may not be captured correctly even if the input capture signal is input to the TIOC1A and TIOC2A pins or the TIOC1B and TIOC2B pins at the same time. This is because a difference of up to 1 clock cycle in the timing of the capture signal to input to TPU1.TCNT and TPU2.TCNT may occur due to internal delays.

For example, the counter value is captured at an overflow of TPU2.TCNT as in counting up from 0A31 FFFFh to 0A32 0000h, the value captured may be 0A31 0000h or 0A32 FFFFh.

The same applies to cascaded operation of TPU4.TCNT and TPU5.TCNT.

26.10.12 Conflict between Overflow/Underflow and Counter Clearing

If overflow/underflow and counter clearing*1 occur simultaneously, TPUm.TCNT is cleared with the generation of the compare match interrupt and an overflow interrupt is generated.

Figure 26.51 shows the operation timing when a TPUm.TGRy compare match is specified as the clearing source and FFFFh is set in TGRy.

Note 1. There are four counter clearing sources:
- Compare match
- Input capture
- Synchronous clearing
- Counter restart operation by an event signal

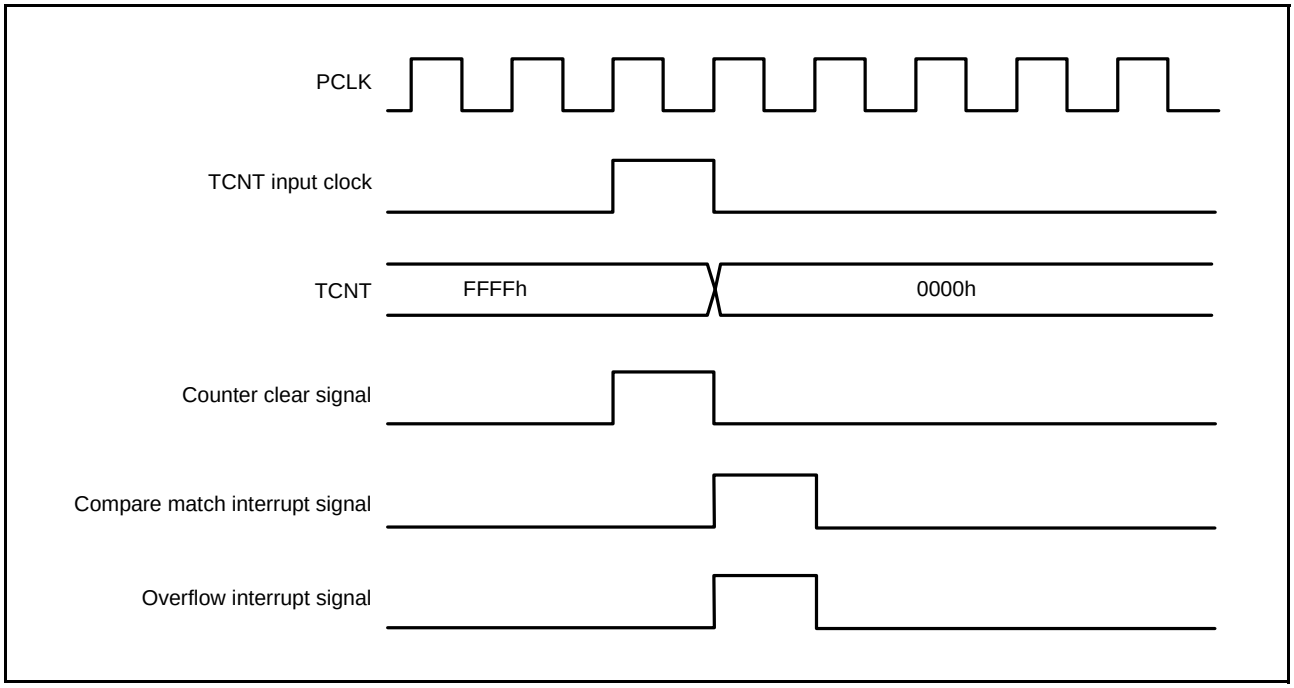


Figure 26.51 Conflict between Overflow and Counter Clearing

26.10.13 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 26.52 shows the operation timing when there is conflict between TCNT write and overflow.

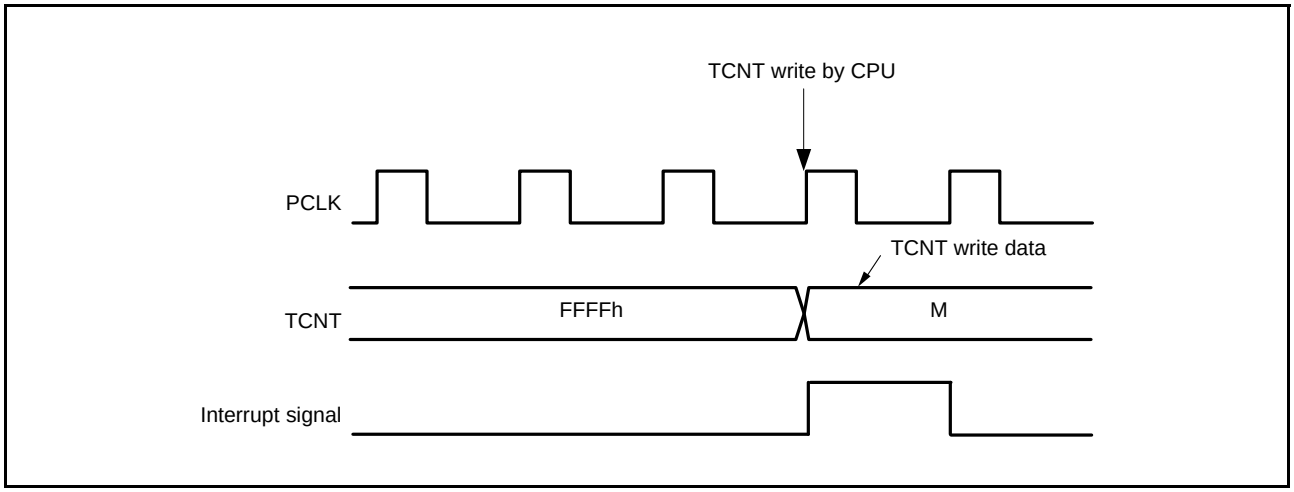


Figure 26.52 Conflict between TPUm.TCNT Write and Overflow

26.10.14 Multiplexing of I/O Pins

In this MCU, the TCLKA input pin is multiplexed with the TIOCB5 I/O pin, the TCLKB input pin with the TIOCB2 I/O pin, the TCLKC input pin with the TIOCB1 I/O pin, the TCLKD input pin with the TIOCB0 I/O pin, the TCLKB input pin with the TIOCD0 I/O pin, the TCLKC input pin with the TIOCC3 I/O pin, and the TCLKD input pin with the TIOCD3 I/O pin. When an external clock is input, compare match output should not be performed from a multiplexed pin.

26.10.15 Continuous Output of Compare-Match Pulse Interrupt Signal

When TGR is set to 0000h, PCLK/1 is set as the count clock, and compare match is set as the counter clear source, the TCNT remains 0000h and is not updated, and a compare-match pulse interrupt signal is output continuously to form a flat signal level.

When a pulse interrupt signal is used, the interrupt controller cannot detect the second and subsequent interrupts.

Figure 26.53 shows an operation timing when the compare-match pulse interrupt signal is continuously output.

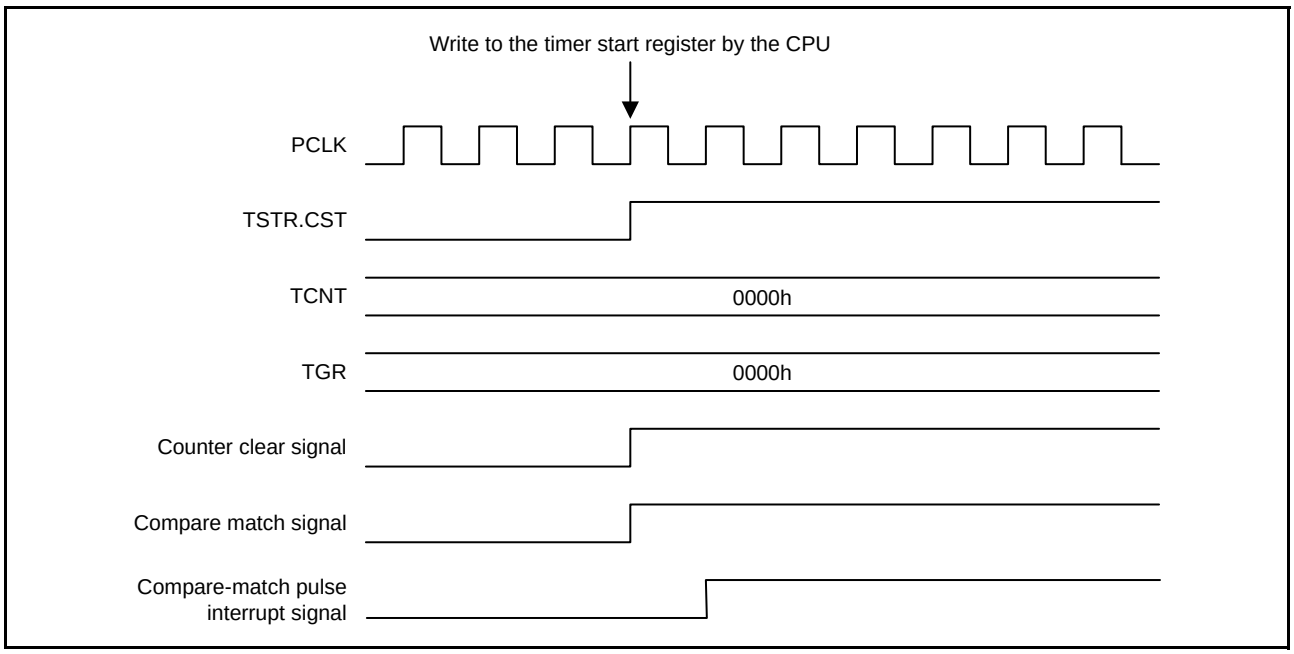


Figure 26.53 Continuous Output of Compare-Match Pulse Interrupt Signal

26.10.16 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 26.54 shows an operation timing when the input-capture pulse interrupt signal is output continuously.

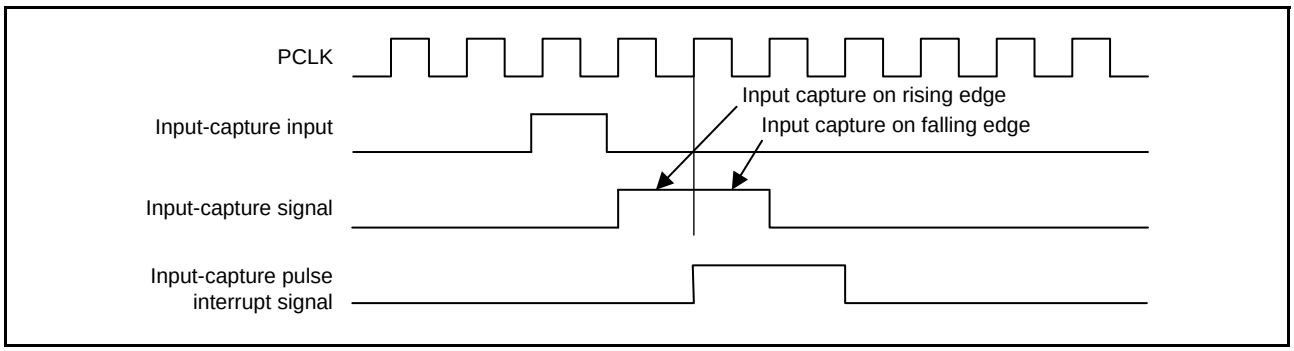


Figure 26.54 Continuous Output of Input-Capture Pulse Interrupt Signal

26.10.17 Continuous Output of Underflow Pulse Interrupt Signal

If two external clock signals' same direction edges to be phase counted are generated within two PCLK cycles in phase counting mode 1, with TGR being 0000h, and compare match set as the counter clear source, the TCNT remains 0000h and is not updated, and a compare-match pulse interrupt signal and an underflow interrupt signal are output continuously to form a flat signal level.

When a pulse interrupt signal is used, the interrupt controller cannot detect the second and subsequent interrupts.

Figure 26.55 shows an operation timing when the underflow pulse interrupt signal is output continuously.

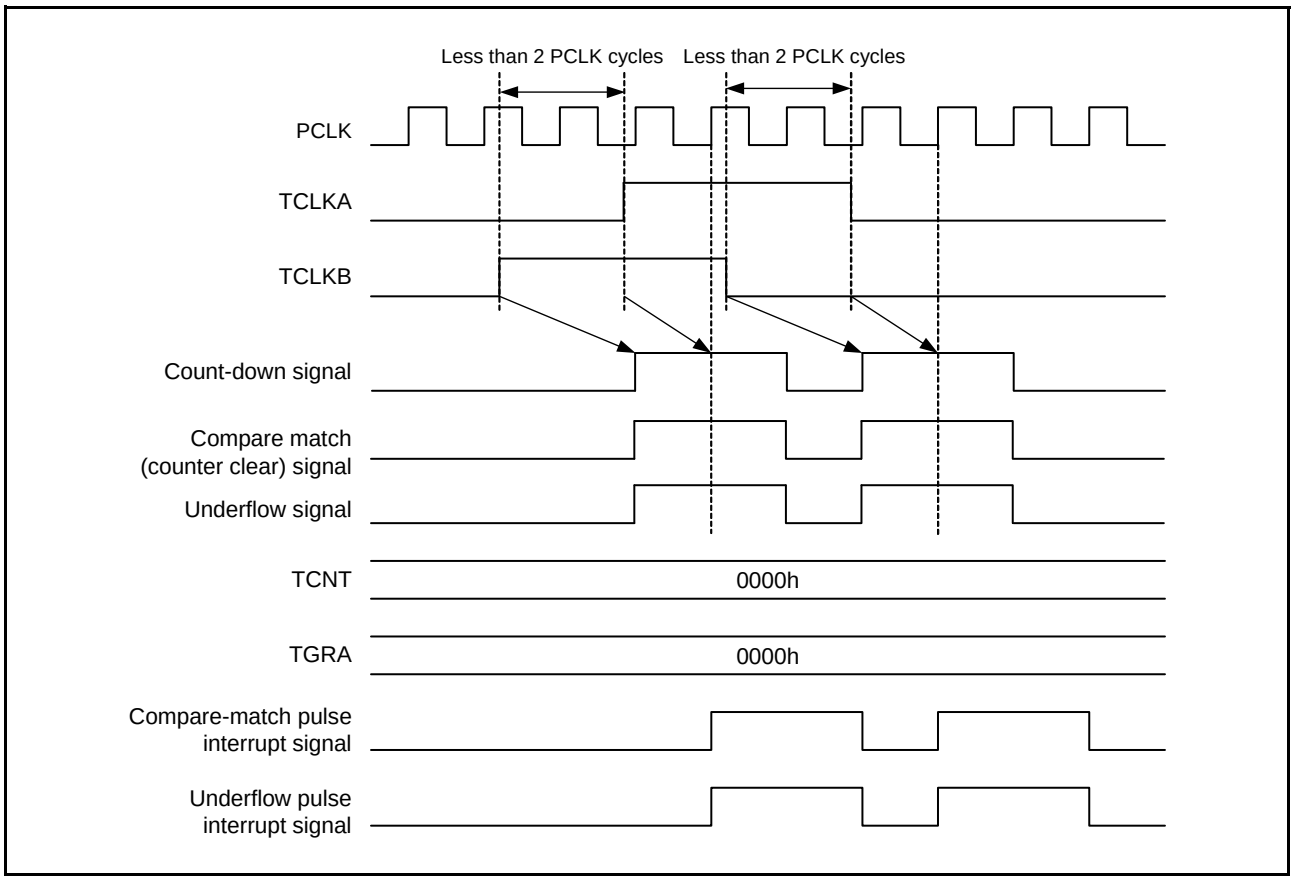


Figure 26.55 Continuous Output of Underflow Pulse Interrupt Signal

26.11 Event Link Operation

26.11.1 Event Signal Output to ELC

The TPU uses the ELC (event link controller) to perform link operation to the previously specified module using the interrupt request signal as the event signal.
The event signal can be output regardless of the setting of the corresponding interrupt request enable bits (TGIEA, TGIEB, TGIEC, or TGIED, and TCIEV, or TCIEU).

26.11.2 Event Signal Input from ELC

The TPU can perform any of the following three operations using the event link setting register of the ELC (event link controller).

(1) Start Counting

When an event signal is input while the TPU count start operation is selected, the CSTn bit in TSTRA register (the timer start register) is set to 1 and counting starts.
However, if this event is generated for the channels when the CSTn bit is set to 1, the event is ignored.
Table 26.30 lists the TSTRA.CSTn bit used for each channel.
Figure 26.56 shows the timing of the count start operation.

For details on the setting procedure to start counting, see section 26.3.1, (1) Counter Operation.

Table 26.30 Correspondence between Channels and TSTRA.CSTn bit

Channel No.	TSTRA.CSTn bit
TPU0	CST0
TPU1	CST1
TPU2	CST2
TPU3	CST3

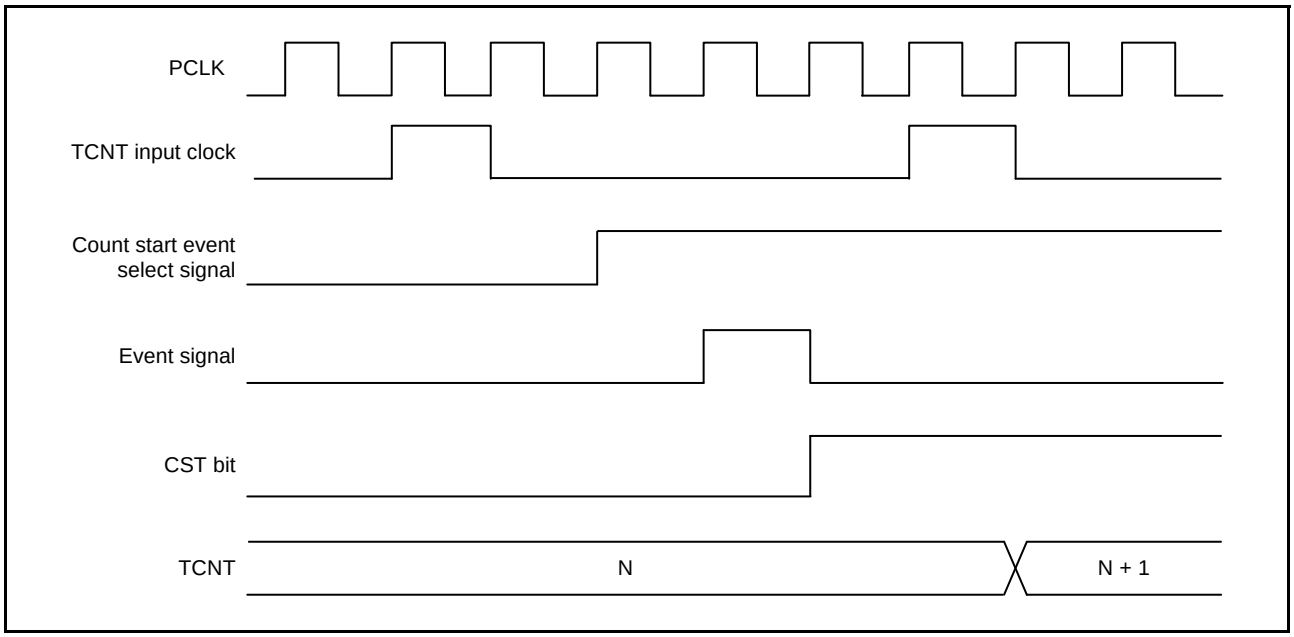


Figure 26.56 Start Counting on Input of the Event Signal

(2) Restart Counting

When an event signal is input while the TPU count restart operation is selected, the value of TCNTn counter (the timer start register) is returned to its initial value (0000h). If the CSTn bit in TSTRA register (the timer start register) is set to 1, however, counting operation will then continue.
Table 26.30 lists the TSTRA.CSTn bit used for each channel.
Figure 26.57 shows the timing of the count restart operation.

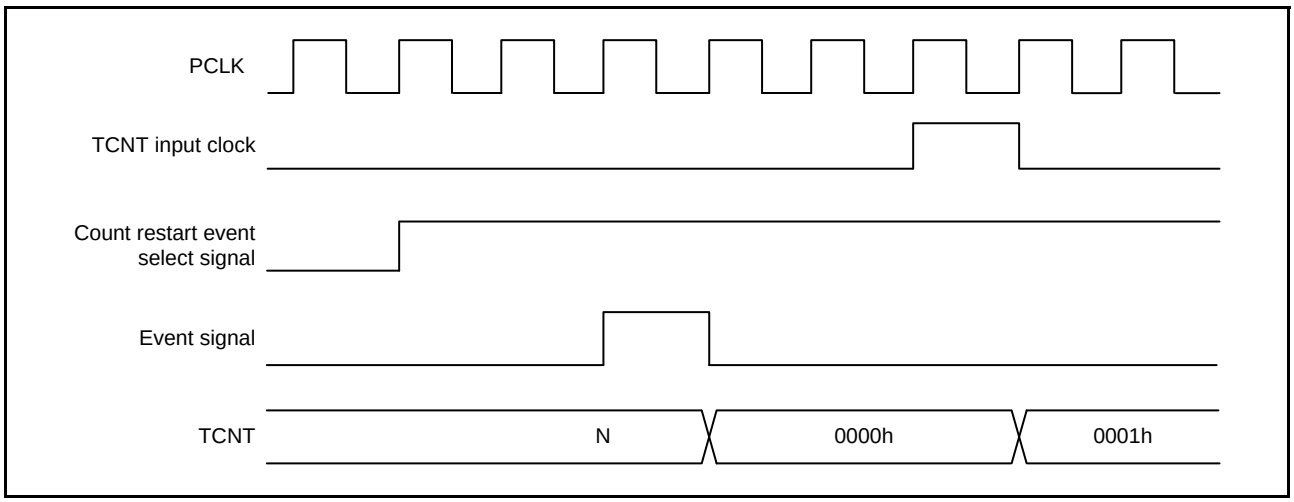


Figure 26.57 Restart Counting on Input of the Event Signal

(3) Input Capture Operation

When an event signal is input while the TPU input capture operation is selected, the value of TCNT counter (the timer count register) for the corresponding channel is captured in TGR register (the timer general register). When using input capture due to the event link, set the bit in TIOR (the timer I/O control register) to specify input capture, and then set the CSTn bit in TSTRA register (the timer start register) to 1 to start counting.
Table 26.31 lists TGR register and TIOR register bits used for each channel. For the TSTRA.CSTn bit used for each channel, see Table 26.30.
Figure 26.58 shows the timing of input capture operation.
When input capture operation by event linking is selected, the setting of TIOR register and the corresponding input capture (the linkage of the TIOCnA pin (input capture pin) input with the specific operation of other channels) are not effective (this also applies when the event select signal is set to 1 at the same time).

For details on the setting procedure for input capture, see section 26.3.1, (3) Input Capture Function.

Table 26.31 TGR and TIOR Used for Input Capture by ELC

Channel No.	Capture Destination Registers	Bits in TIOR
TPU0	TGRA register (channel 0)	IOA[3:0] bits (TIORH0)
TPU1	TGRA register (channel 1)	IOA[3:0] bits (TIOR1)
TPU2	TGRA register (channel 2)	IOA[3:0] bits (TIOR2)
TPU3	TGRA register (channel 3)	IOA[3:0] bits (TIORH3)

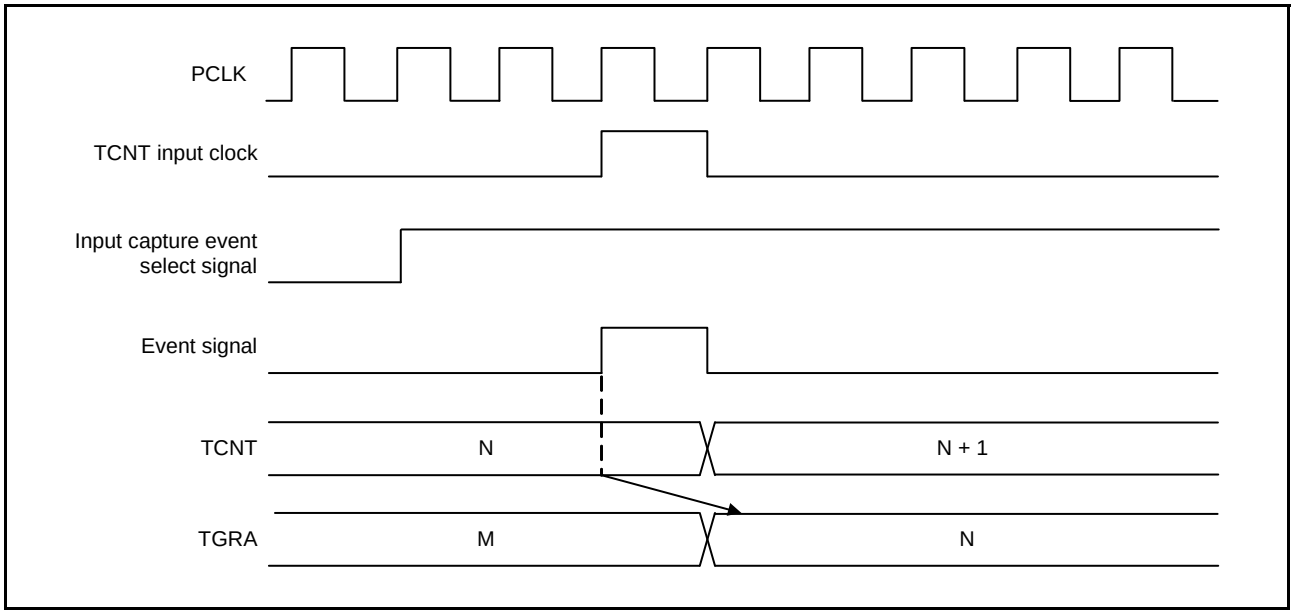


Figure 26.58 Input Capture on Input of the Event Signal

26.11.3 Usage Notes on Operation on Input of the Event Signal

The followings are the notes on using the TPU for event link operations.

(1) Start Counting

When writing to the TSTRA.CSTn bit (the timer start register) and a counting start are in contention, writing to the CSTn bit does not proceed since setting of the CSTn bit to 1 in response to the event takes priority.
Figure 26.59 shows the timing in this case.
Furthermore, even when a counting start due to the event link is selected, CPU writing to the TSTRA.CSTn bit proceeds if the event signal is low.

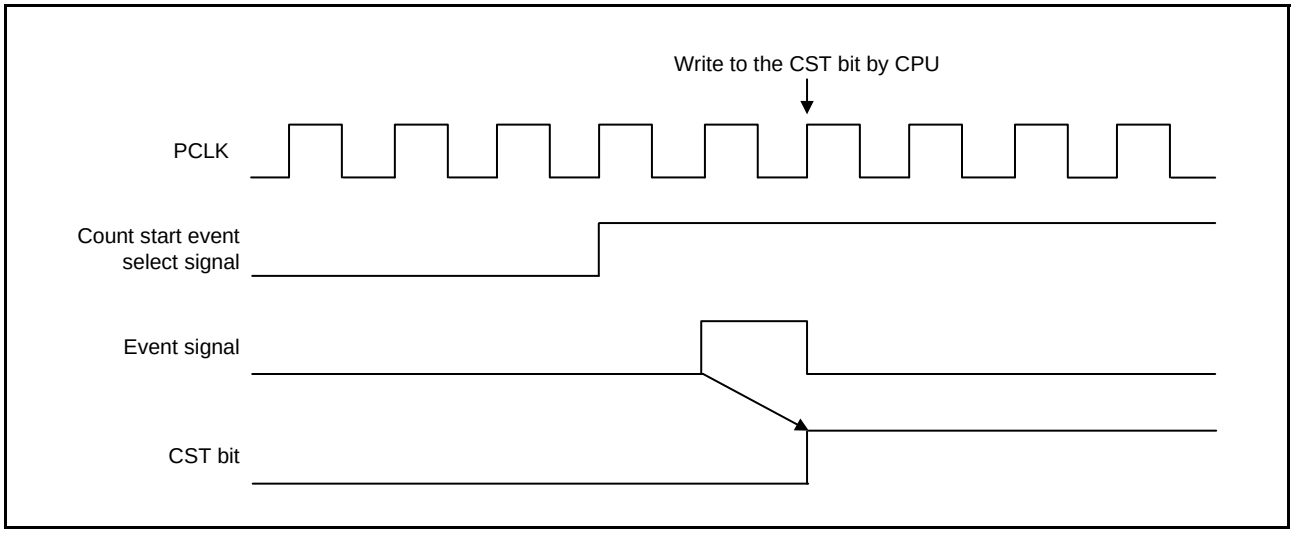


Figure 26.59 Conflict between Writing to the CSTn Bit and Counting Start

(2) Restart Counting

When a TCNTn counter (the timer count register) write cycle and a counting restart are in contention, writing to TCNTn counter does not proceed since the counter value initialization in response to the counting restart takes priority.
Figure 26.60 shows the timing in this case.
Furthermore, even when a counting restart due to the event link is selected, CPU writing to TCNTn counter proceeds if the event signal is low.

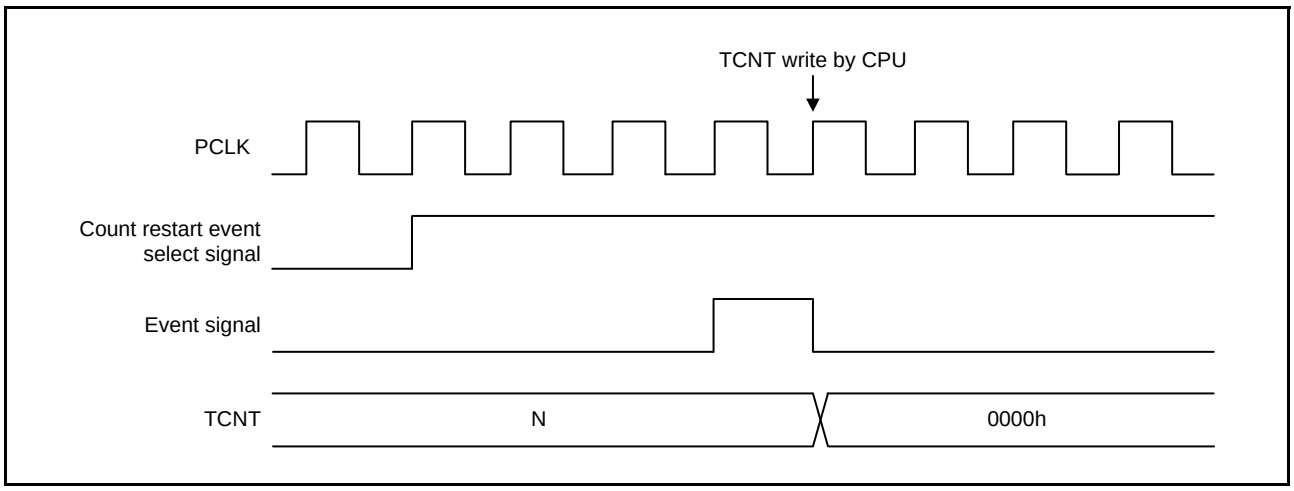


Figure 26.60 Conflict between TCNTn Write Cycle and Counting Restart

(3) Input Capture Operation

If a TGRA register (the timer general register) read/write cycle and input capture operation are in contention, operation proceeds as follows:

(a) Conflict between TGR Read Cycle and Input Capture

The internal data bus reads the data before input capture transfer.
Figure 26.61 shows the timing in this case.

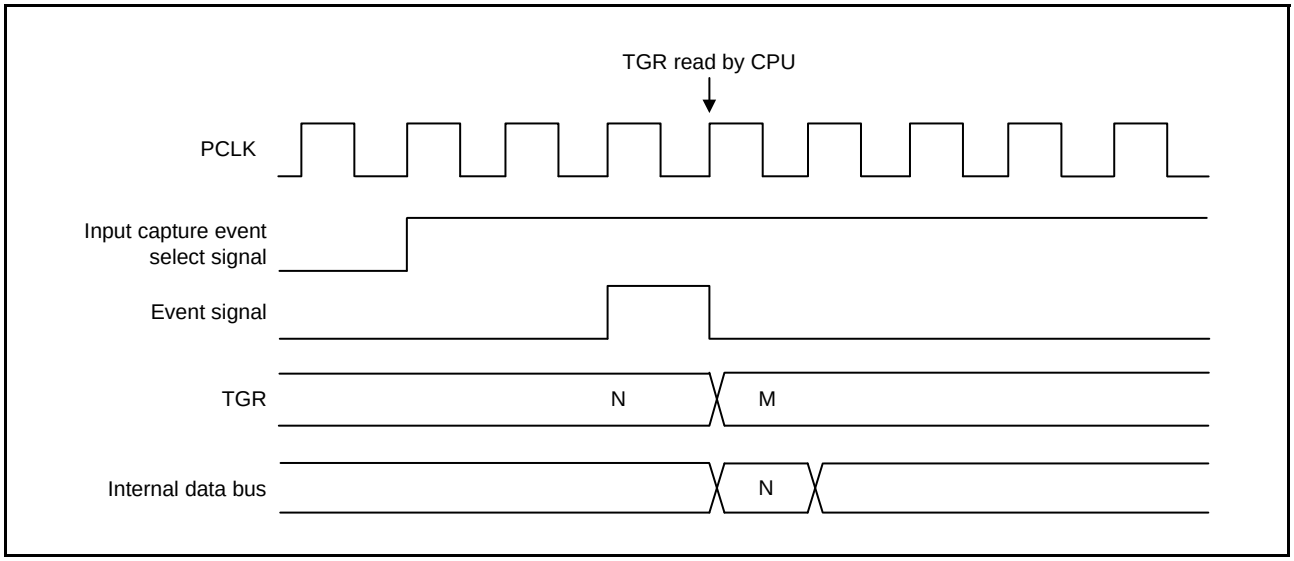


Figure 26.61 Conflict between TGR Read Cycle and Input Capture Operation

(b) Conflict between TGR Write Cycle and Input Capture

Writing to TGRA register does not proceed since input capture takes priority.

Figure 26.62 shows the timing in this case.

Furthermore, even when input capture operation due to the event link is selected, CPU writing to TGR proceeds if the event signal is low.

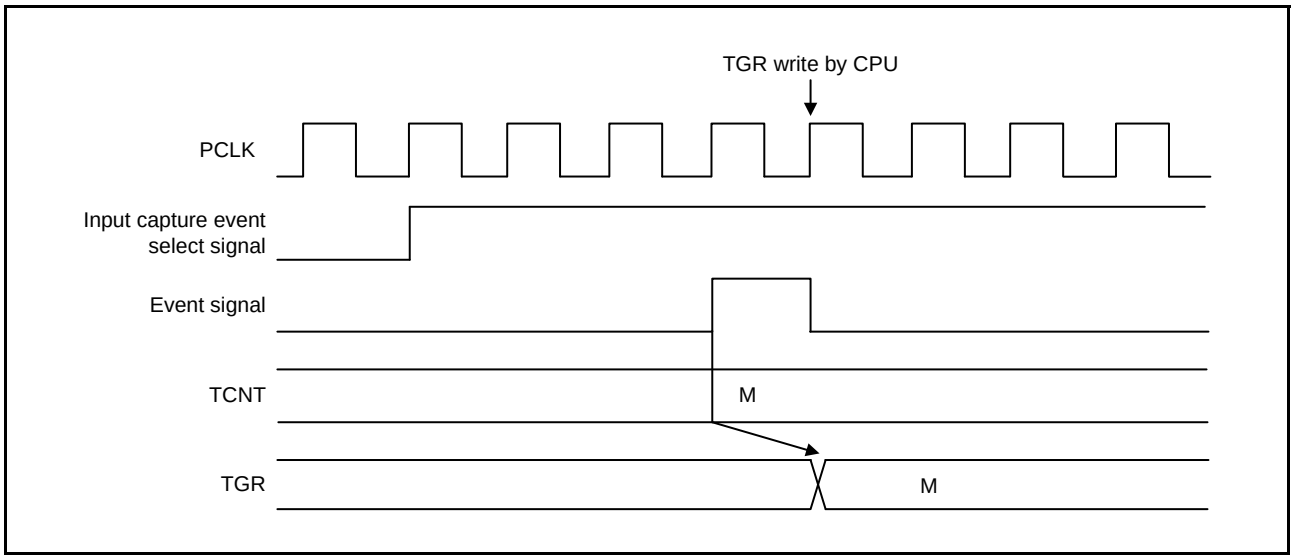


Figure 26.62 Conflict between TGR Write Cycle and Input Capture Operation

26.11.4 Notes on Output of the Event Signal

The followings are the notes on output of the event signal.

(1) Output of the Compare Match Event Signal

When the TGR register is set to 0000h, PCLK/1 is set as the counter clock, and compare match is set as the trigger for clearing of the counter clock (TCRn.TPSC[2:0] = 000b), the value of the TCNT remains 0000h, and the event output signal will be output continuously (i.e. its level will be flat) rather than output over a single cycle.
Figure 26.63 shows the timing for continuous output of the event output signal in response to a compare match.

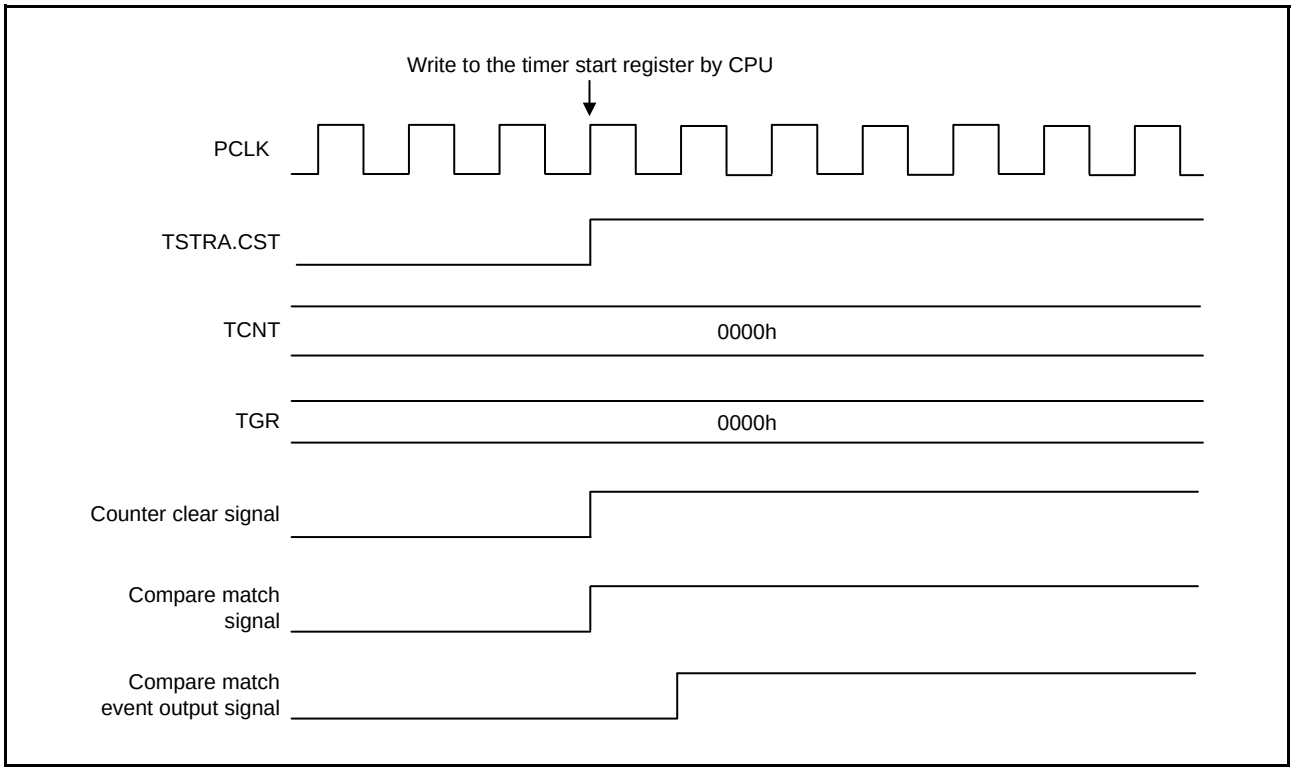


Figure 26.63 Continuous Output of the Compare Match Event Output Signal

(2) Output of the Underflow Event Signal

If two external clock signals' same direction edges to be phase counted are generated within two PCLK cycles in phase counting mode 1, with TGR being 0000h, and compare match set as the counter clear source, the TCNT counter remains 0000h, and a compare-match event signal and an underflow event signal are output continuously to form a flat signal level.

Figure 26.64 shows the timing for continuous output of the event output signal in response to underflow.

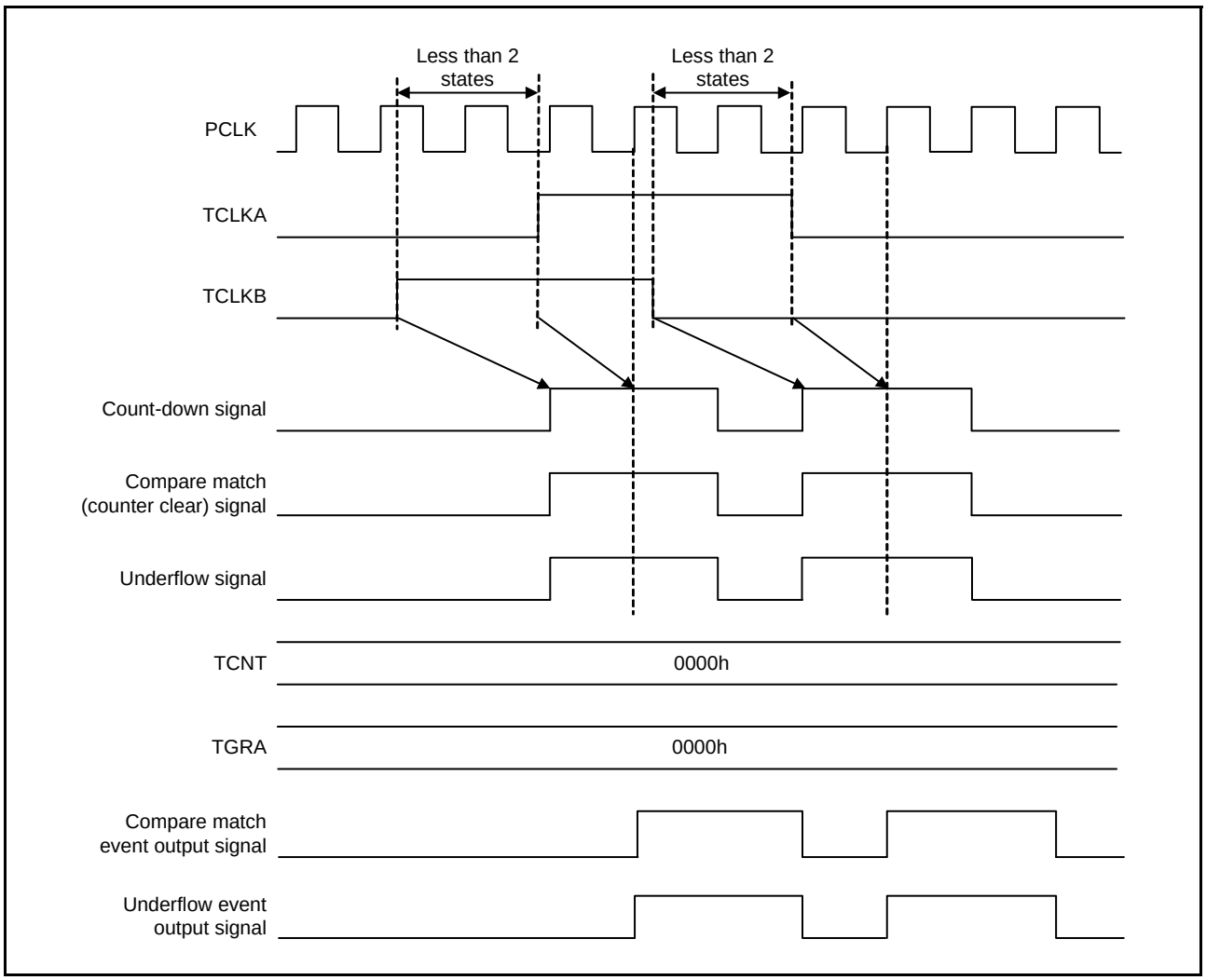


Figure 26.64 Continuous Output of the Underflow Event Output Signal

27. 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 (MTU) as a time base.
This MCU has two PPG units, each of which controls up to 16 pulse output pins. The pulse outputs from the PPGs are divided into 4-bit groups that can operate all simultaneously and independently.

27.1 Overview

Table 27.1 lists the specifications of the PPG and Table 27.2 lists PPG functions.
Figure 27.1 and Figure 27.2 show block diagrams of the PPGs.

Table 27.1 Specifications of PPG

Item	Specifications
Number of output bits	Up to 32 bits*1
Pulse output	• Two units, each capable of output through four pin groups • Output trigger signals are selectable. • Non-overlapping operation is possible. • Inverted output is selectable.
Output data transfer	Can operate together with the DTC and DMAC (when TPU and MTU interrupts are in use)
Power consumption reducing function	Module-stop state can be set for each unit.

Note 1. When setting PPG output trigger in MTU, make settings so that PCLKA run at the same frequency as PCLKB.

Table 27.2 List of PPG Functions

Item			PPG0	PPG1
PPG output trigger	MTU channels 0 to 3 (MTU0 to MTU3)*1	Compare match	✓	✓
		Input capture	✓	✓
	TPU (unit 0) channels 0 to 3 (TPU0 to TPU3)	Compare match	—	✓
		Input capture	—	✓
Non-overlapping operation			✓	✓
Output data transfer	DTC		✓	✓
	DMAC		✓	✓
Selecting inverted output			✓	✓
Setting the module-stop state*2			The MSTPA11 bit in MSTPCRA	The MSTPA10 bit in MSTPCRA

✓: Possible
—: Not possible
Note 1. When setting PPG output trigger in MTU, make settings so that PCLKA run at the same frequency as PCLKB.
Note 2. For details, refer to section 11, Low Power Consumption.

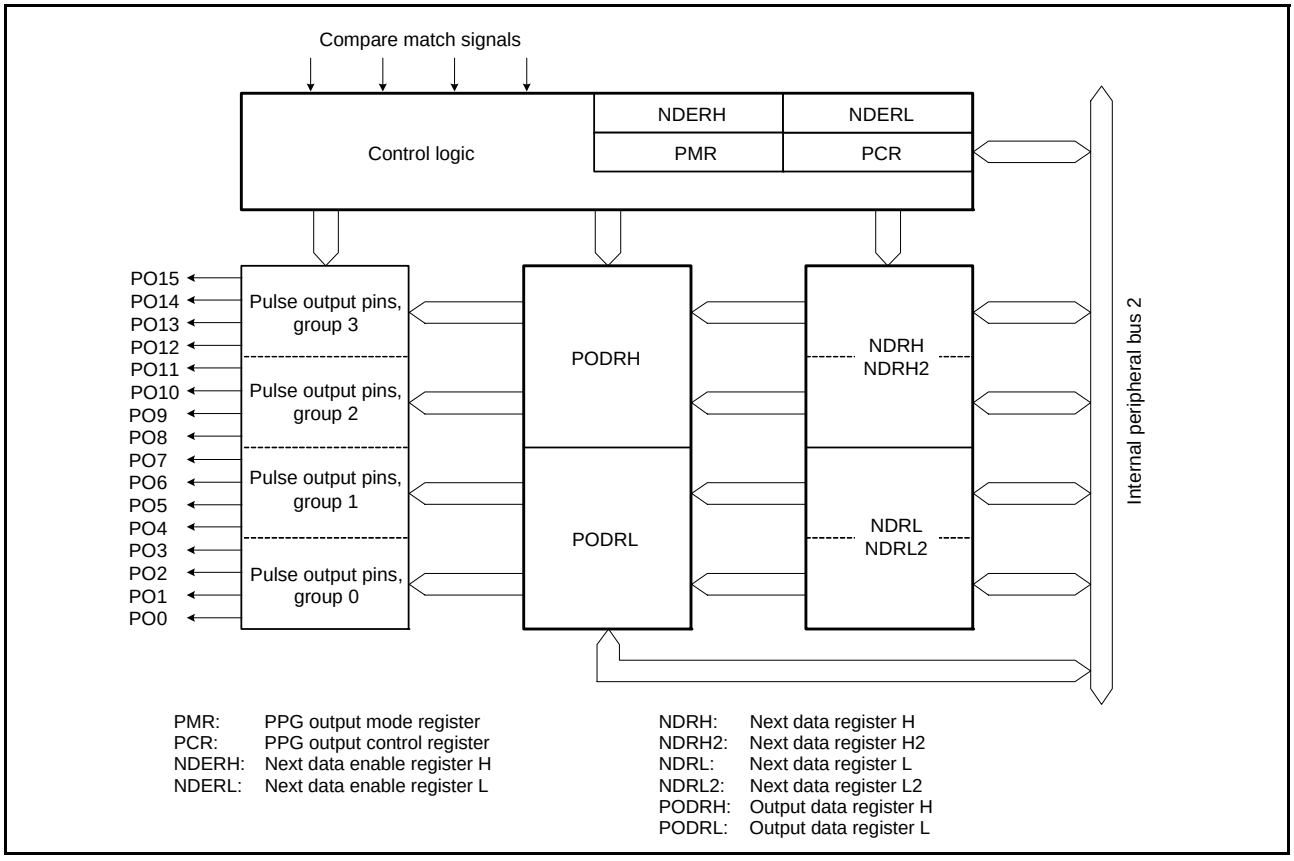


Figure 27.1 Block Diagram of PPG0

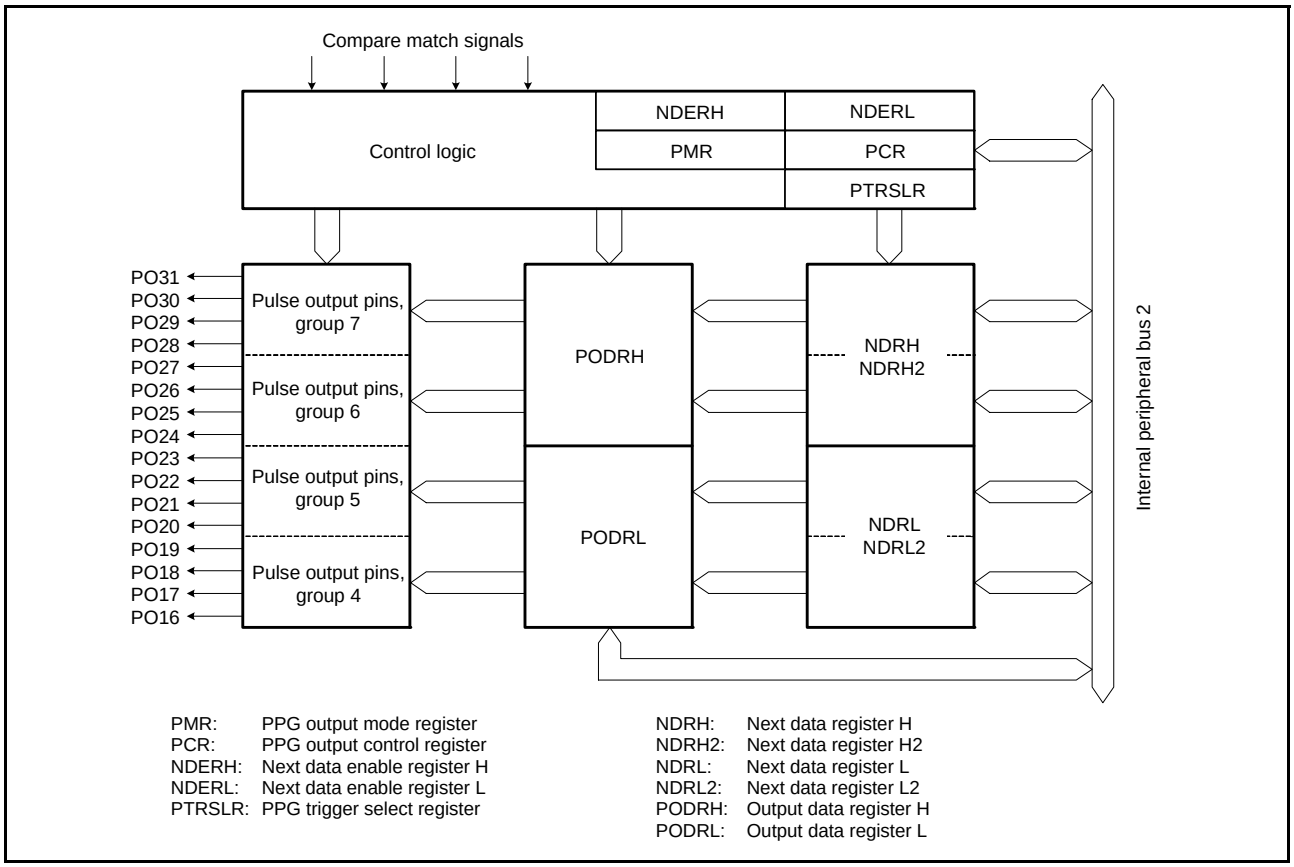


Figure 27.2 Block Diagram of PPG1

Table 27.3 lists the pin configuration of the PPG.

Table 27.3 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	

27.2 Register Descriptions

27.2.1 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
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

PTRSL Bit (PPG Trigger Select)

This bit selects either MTU0 to MTU3 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.

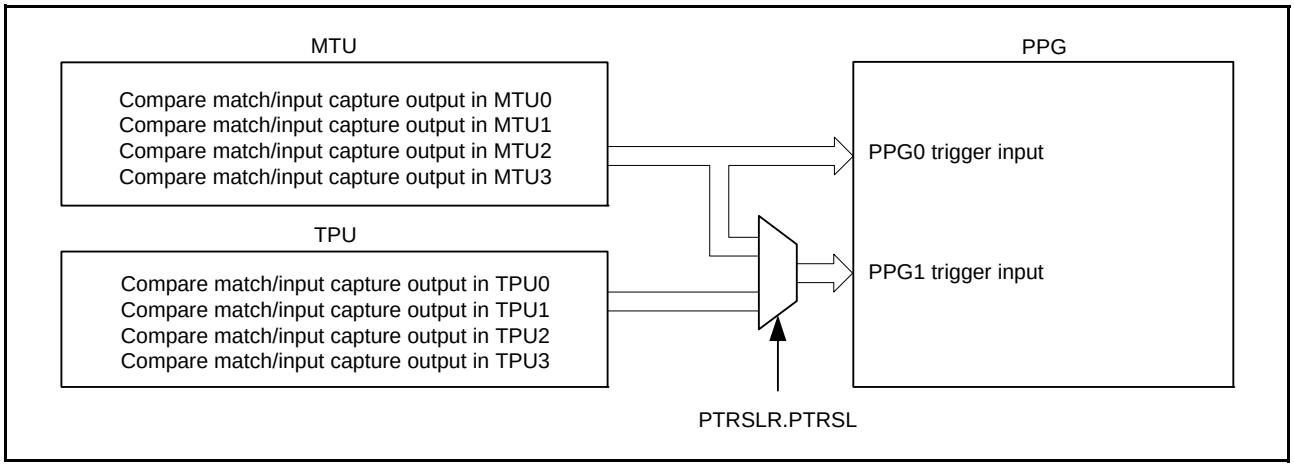


Figure 27.3 Block Diagram of PPG Trigger Selection

27.2.2 Next Data Enable Register H (NDERH), Next Data Enable Register 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)

For NDERi bits with the setting 1, the trigger specified by PPG0.PCR leads to the transfer of the values from the corresponding bits of the PPG0.NDRH or PPG0.NDRH2 register to those of the PPG0.PODRH register. For NDERi bits with the setting 0, the values are not transferred from the corresponding bits of the PPG0.NDRH or PPG0.NDRH2 register to those of the PPG0.PODRH register.

- 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)

For NDERi bits with the setting 1, the trigger specified by PPG0.PCR leads to the transfer of the values from the corresponding bits of the PPG0.NDRL or PPG0.NDRL2 register to those of the PPG0.PODRL register. For NDERi bits with the setting 0, the values are not transferred from the corresponding bits of the PPG0.NDRL or PPG0.NDRL2 register to those of the PPG0.PODRL register.

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)

For NDERi bits with the setting 1, the trigger specified by PPG1.PCR leads to the transfer of the values from the corresponding bits of the PPG1.NDRH or PPG1.NDRH2 register to those of the PPG1.PODRH register. For NDERi bits with the setting 0, the values are not transferred from the corresponding bits of the PPG1.NDRH or PPG1.NDRH2 register to those of the PPG1.PODRH register.

- 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)

For NDERi bits with the setting 1, the trigger specified by PPG1.PCR leads to the transfer of the values from the corresponding bits of the PPG1.NDRL or PPG1.NDRL2 register to those of the PPG1.PODRL register. For NDERi bits with the setting 0, the values are not transferred from the corresponding bits of the PPG1.NDRL or PPG1.NDRL2 register to those of the PPG1.PODRL register.

27.2.3 Output Data Register H (PODRH), Output Data Register 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 or PPG0.NDRH2 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 or PPG0.NDRH2 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 or PPG0.NDRL2 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 or PPG0.NDRL2 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 or PPG1.NDRH2 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 or PPG1.NDRH2 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 or PPG1.NDRL2 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 or PPG1.NDRL2 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.

27.2.4 Next Data Register H (NDRH), Next Data Register L (NDRL), Next Data Register H2 (NDRH2), Next Data Register L2 (NDRL2)

Address(es): 0008 81ECh

	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

	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

Address(es): 0008 81EEh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG0.NDRH2	—	—	—	—	NDR11	NDR10	NDR9	NDR8
Value after reset:	0	0	0	0	0	0	0	0

Address(es): 0008 81EFh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG0.NDRL2	—	—	—	—	NDR3	NDR2	NDR1	NDR0
Value after reset:	0	0	0	0	0	0	0	0

- PPG0.NDRH

The PPG0.NDRH register holds the next pulse output value to the one stored in the PODRH register. The assignment of bits in the PPG0.NDRH register depends on whether pulse output groups are set as having 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 NDRH register and can be accessed in a single operation.

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

(2) When pulse output groups 2 and 3 have different output triggers
If different output triggers are selected for pulse output groups 2 and 3, the pulse output group 3 are mapped to the upper 4 bits.

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

- PPG0.NDRH2

If different output triggers are selected for pulse output groups 2 and 3, the pulse output group 2 are mapped to the lower 4 bits of the PPG0.NDRH2 register.
If the same output trigger is selected for pulse output groups 2 and 3, the PPG0.NDRH2 register is not used. With such settings, the value read is FFh and writing is prohibited.

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

The PPG0.NDRL register holds the next pulse output value to the one stored in the PODRL register. The assignment of bits in the PPG0.NDRL register depends on whether pulse output groups are set as having 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 NDRL register and can be accessed in a single operation.

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

(2) When pulse output groups 0 and 1 have different output triggers

If different output triggers are selected for pulse output groups 0 and 1, the pulse output group 1 are mapped to the upper 4 bits.

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

- PPG0.NDRL2

If different output triggers are selected for pulse output groups 0 and 1, the pulse output group 0 are mapped to the lower 4 bits of the PPG0.NDRL2 register.

If the same output trigger is selected for pulse output groups 0 and 1, the PPG0.NDRL2 register is not used. With such settings, the value read is FFh and writing is prohibited.

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

	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

	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

Address(es): 0008 81FEh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG1.NDRH2	—	—	—	—	NDR27	NDR26	NDR25	NDR24
Value after reset:	0	0	0	0	0	0	0	0

Address(es): 0008 81FFh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG1.NDRL2	—	—	—	—	NDR19	NDR18	NDR17	NDR16
Value after reset:	0	0	0	0	0	0	0	0

- PPG1.NDRH

The PPG1.NDRH register holds the next pulse output value to the one stored in the PODRH register. The assignment of bits in the PPG1.NDRH register depends on whether pulse output groups are set as having 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 NDRH register and can be accessed in a single operation.

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

(2) When pulse output groups 6 and 7 have different output triggers
If different output triggers are selected for pulse output groups 6 and 7, the pulse output group 7 are mapped to the upper 4 bits.

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

- PPG1.NDRH2

If different output triggers are selected for pulse output groups 6 and 7, the pulse output group 6 are mapped to the lower 4 bits of the PPG1.NDRH2 register.
If the same output trigger is selected for pulse output groups 6 and 7, the PPG1.NDRH2 register is not used. With such settings, the value read is FFh and writing is prohibited.

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

The PPG1.NDRL register holds the next pulse output value to the one stored in the PODRL register. The assignment of bits in the PPG1.NDRL register depends on whether pulse output groups are set as having 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 NDRL register and can be accessed in a single operation.

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

(2) When pulse output groups 4 and 5 have different output triggers

If different output triggers are selected for pulse output groups 4 and 5, the pulse output group 5 are mapped to the upper 4 bits.

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

- PPG1.NDRL2

If different output triggers are selected for pulse output groups 4 and 5, the pulse output group 4 are mapped to the lower 4 bits of the PPG1.NDRL2 register.

If the same output trigger is selected for pulse output groups 4 and 5, the PPG1.NDRL2 register is not used. With such settings, the value read is FFh and writing is prohibited.

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

27.2.5 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, refer to section 27.2.6, PPG Output Mode Register (PMR).

27.2.6 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.
For details, refer to section 27.3.4, Pulse Output in Non-Overlapping Operation.

27.3 Operation

Figure 27.4 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, PPGn.NDRL, PPGn.NDRH2, and PPGn.NDRL2 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, PPGn.NDRL, PPGn.NDRH2, and PPGn.NDRL2 before the next compare match.

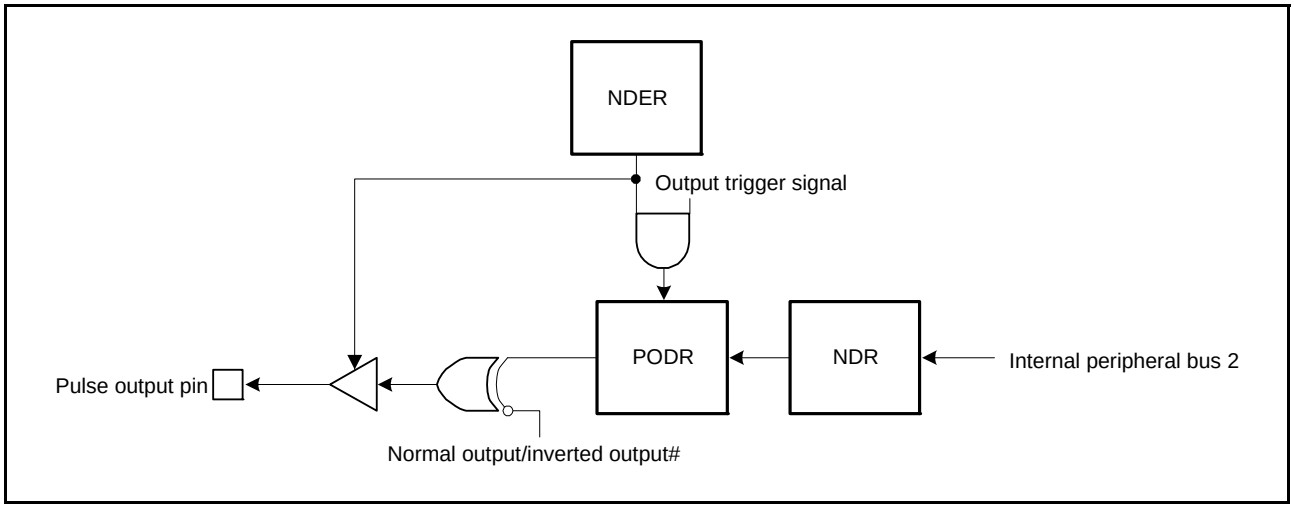


Figure 27.4 Schematic Diagram of PPG

27.3.1 Output Timing

When the selected compare match event occurs while pulse output is enabled, the values in PPGn.NDRH, PPGn.NDRL, PPGn.NDRH2, and PPGn.NDRL2 (n = 0, 1) are transferred to PPGn.PODRH and PPGn.PODRL, respectively, and then output on the corresponding pins.

Figure 27.5 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.

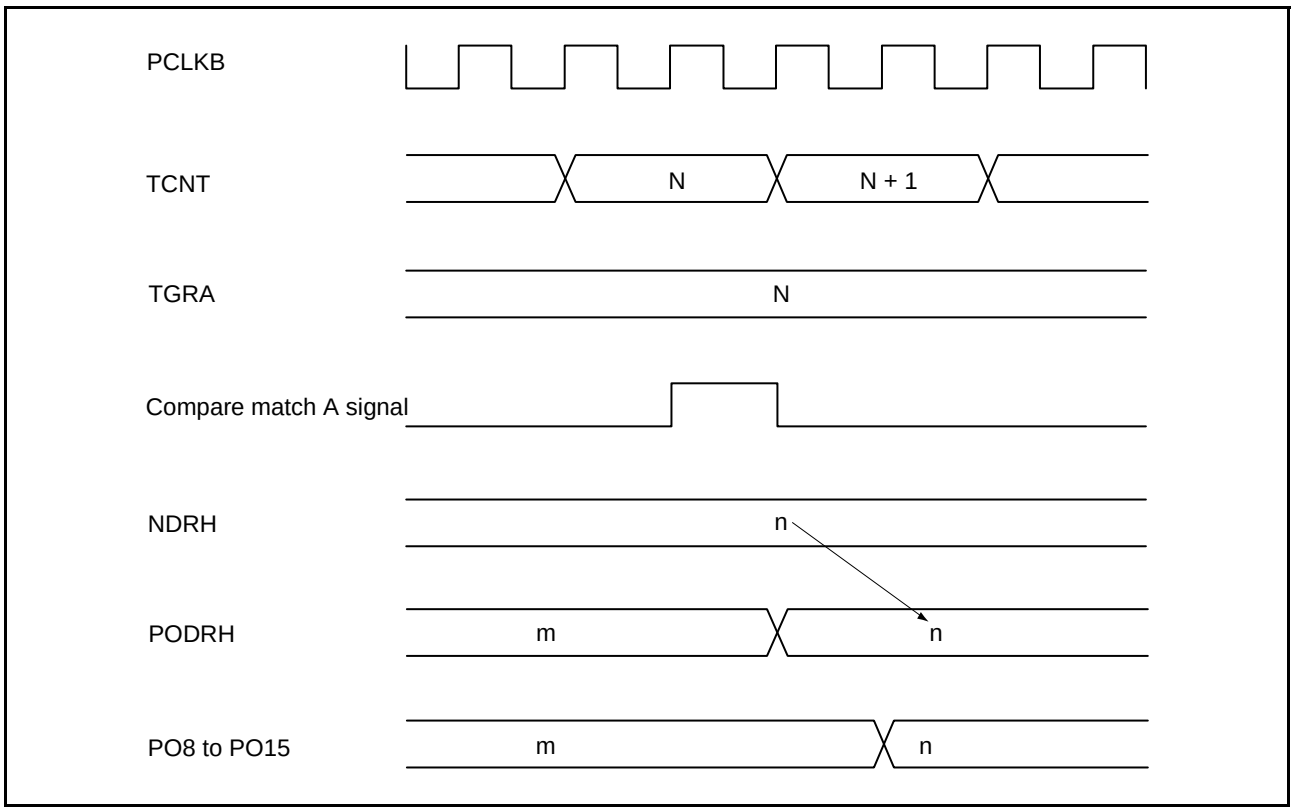


Figure 27.5 Timing of Transfer and Output of the Values in NDR (Example)

27.3.2 Example of Setup Procedure for Pulse Output in Normal Operation

Figure 27.6 and Figure 27.7 show examples of setup procedures for pulse output in normal operation.

(1) PPG0 Setting

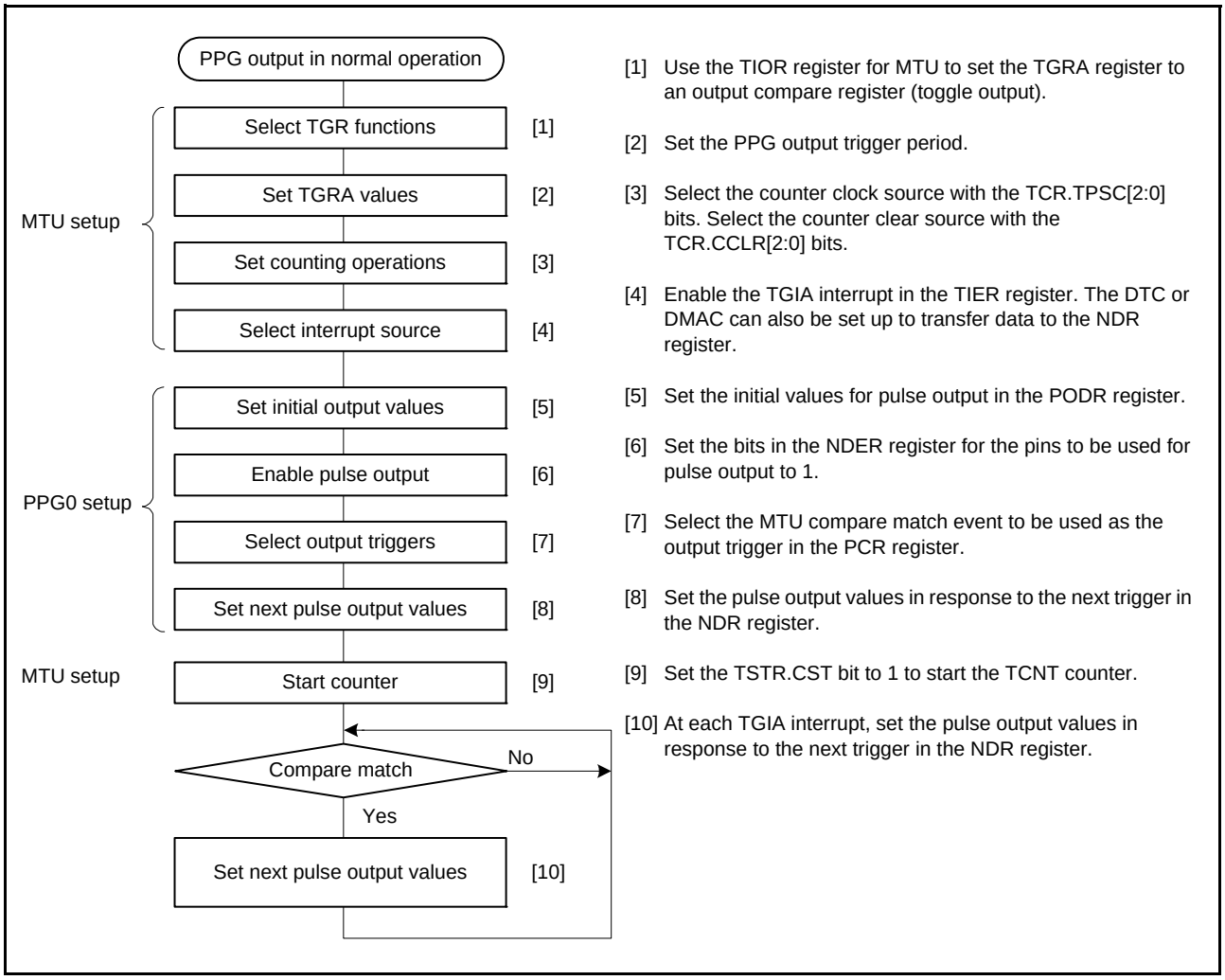
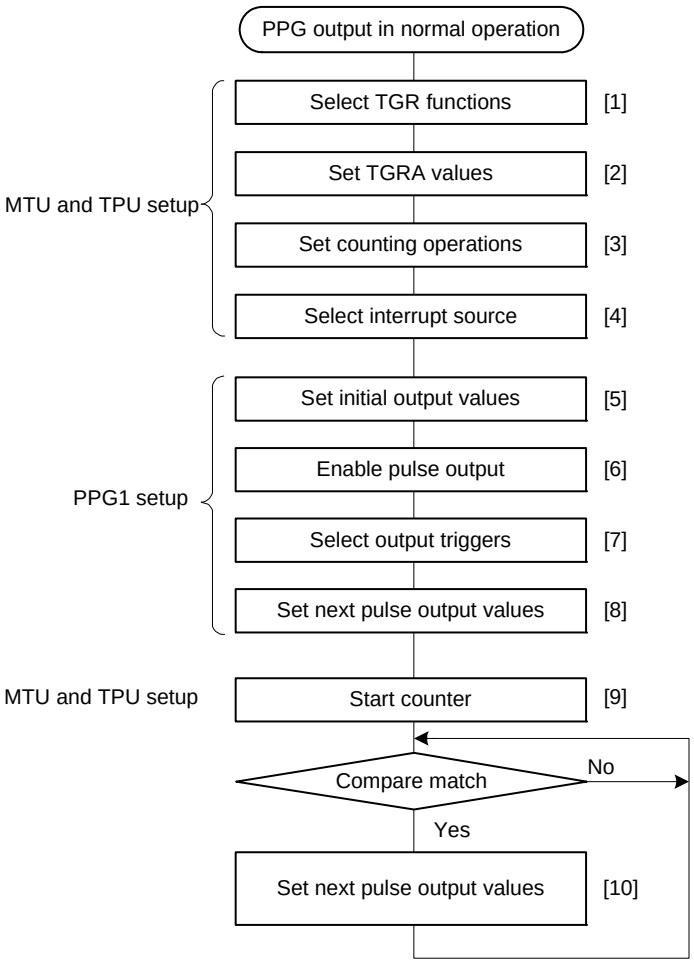


Figure 27.6 Example of Setup Procedure for Pulse Output in Normal Operation (PPG0 Setting)

(2) PPG1 Setting



[1] In the case of MTU, use the TIOR register to set the TGRA register to an output compare register (toggle output).
In the case of TPU, use the TIOR register to set the TGRA register an output compare register (toggle output).

[2] Set the PPG output trigger period.

[3] Select the counter clock source with the TCR.TPSC[2:0] bits. Select the counter clear source with the TCR.CCLR[2:0] bits.

[4] Enable the TGIA interrupt in the TIER register. The DTC or DMAC can also be set up to transfer data to the NDR register.

[5] Set the initial values for pulse output in the PODR register.

[6] Set the bits in the NDER register for the pins to be used for pulse output to 1.

[7] Select the MTU/TPU compare match event to be used as the output trigger in the PTRSLR and PCR registers.

[8] Set the pulse output values in response to the next trigger in the NDR register.

[9] Set the TSTR.CST bit to 1 to start the TCNT counter.

[10] At each TGIA interrupt, set the pulse output values in response to the next trigger in the NDR register.

Figure 27.7 Example of Setup Procedure for Pulse Output in Normal Operation (PPG1 Setting)

27.3.3 Example of Pulse Output in Normal Operation (Example of Five-Phase Pulse Output)

Figure 27.8 shows an example in which pulse output from the PPG0 is used for cyclic five-phase pulse output.

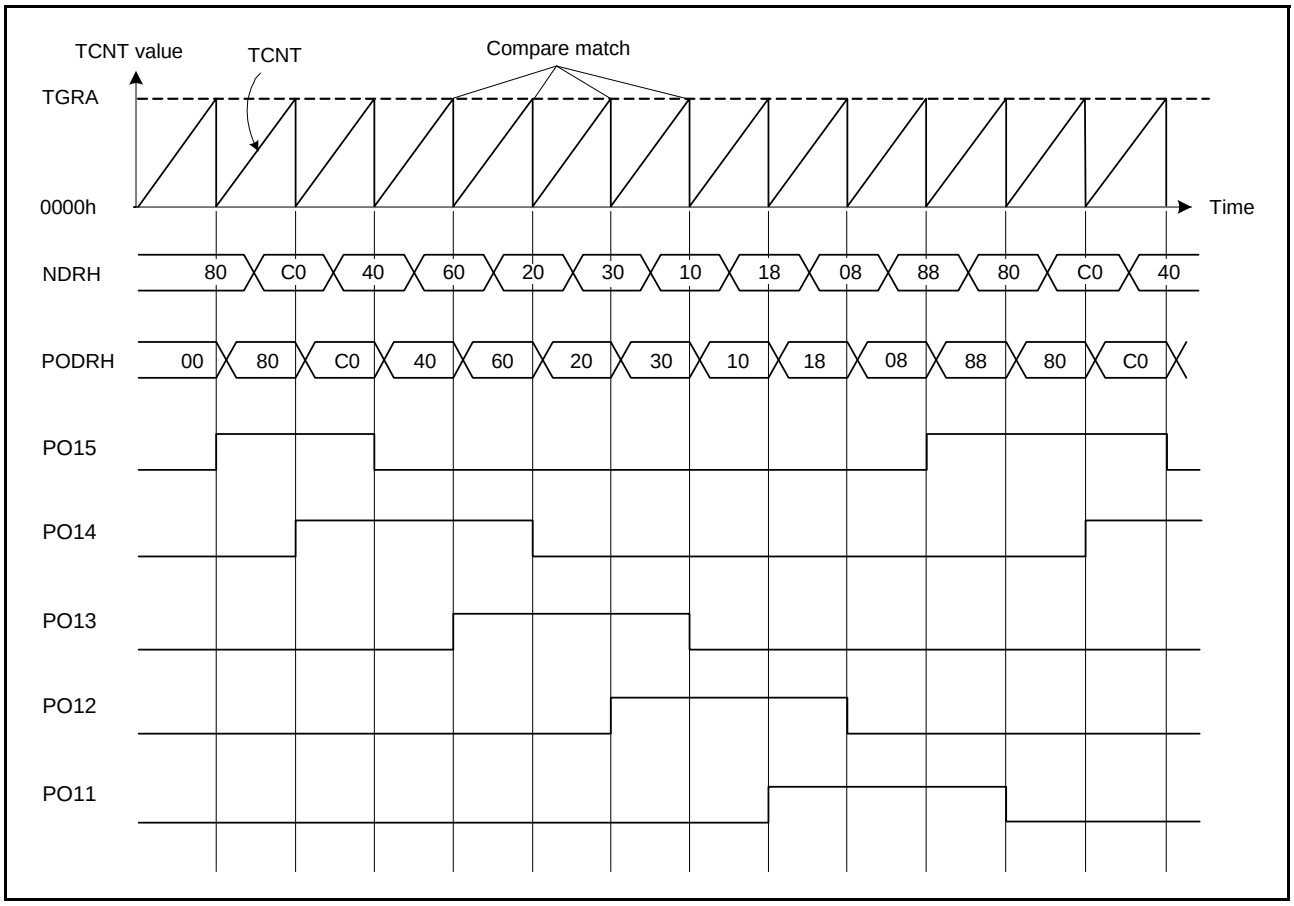


Figure 27.8 Example of Pulse Output in Normal Operation (Example of Five-Phase Pulse Output)

1. Set an output compare register of the MTUn.TGRA (n = 0 to 3) 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.

27.3.4 Pulse Output in Non-Overlapping Operation

During non-overlapping operation, data transfer from PPGn.NDRH, PPGn.NDRL, PPGn.NDRH2, and PPGn.NDRL2 (n = 0, 1) to PPGn.PODRH and PPGn.PODRL is performed as follows.

- On compare match A, the values in PPGn.NDRH, PPGn.NDRL, PPGn.NDRH2, and PPGn.NDRL2 are transferred to PPGn.PODRH and PPGn.PODRL.
- On compare match B, data transfer proceeds for bits in PPGn.NDRH, PPGn.NDRL, PPGn.NDRH2, and PPGn.NDRL2 that have the value 0. It does not proceed for bits having the value 1.

Figure 27.9 shows the pulse output in non-overlapping operation.

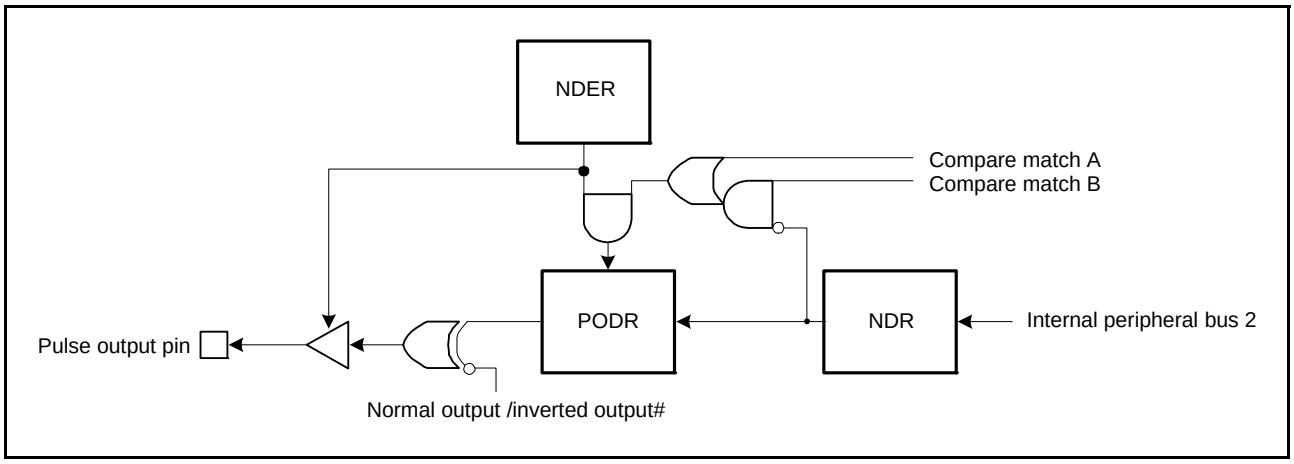


Figure 27.9 Pulse Output in Non-Overlapping Operation

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, PPGn.NDRL, PPGn.NDRH2, and PPGn.NDRL2 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, PPGn.NDRL, PPGn.NDRH2, and PPGn.NDRL2 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 27.10 shows the timing of the above procedure.

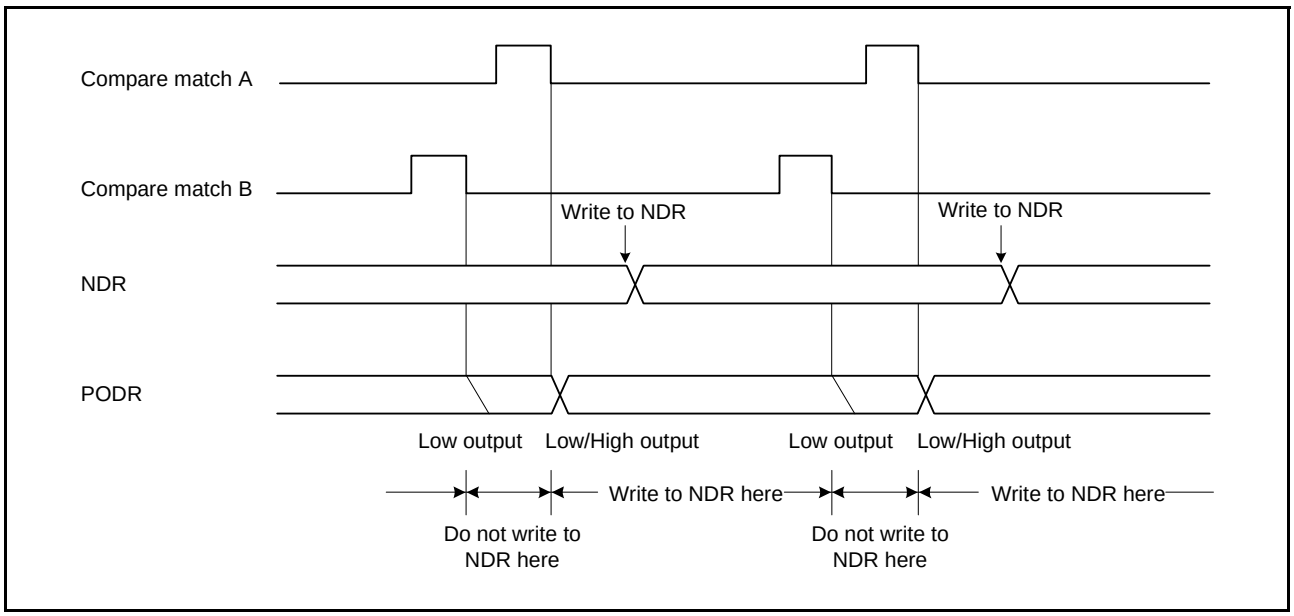


Figure 27.10 Non-Overlapping Operation and Write Timing to PPGn.NDRH, PPGn.NDRL, PPGn.NDRH2, and PPGn.NDRL2

27.3.5 Example of Setup Procedure for Pulse Output in Non-Overlapping Operation

Figure 27.11 and Figure 27.12 show examples of setup procedure for pulse output in non-overlapping operation.

(1) PPG0 Setting

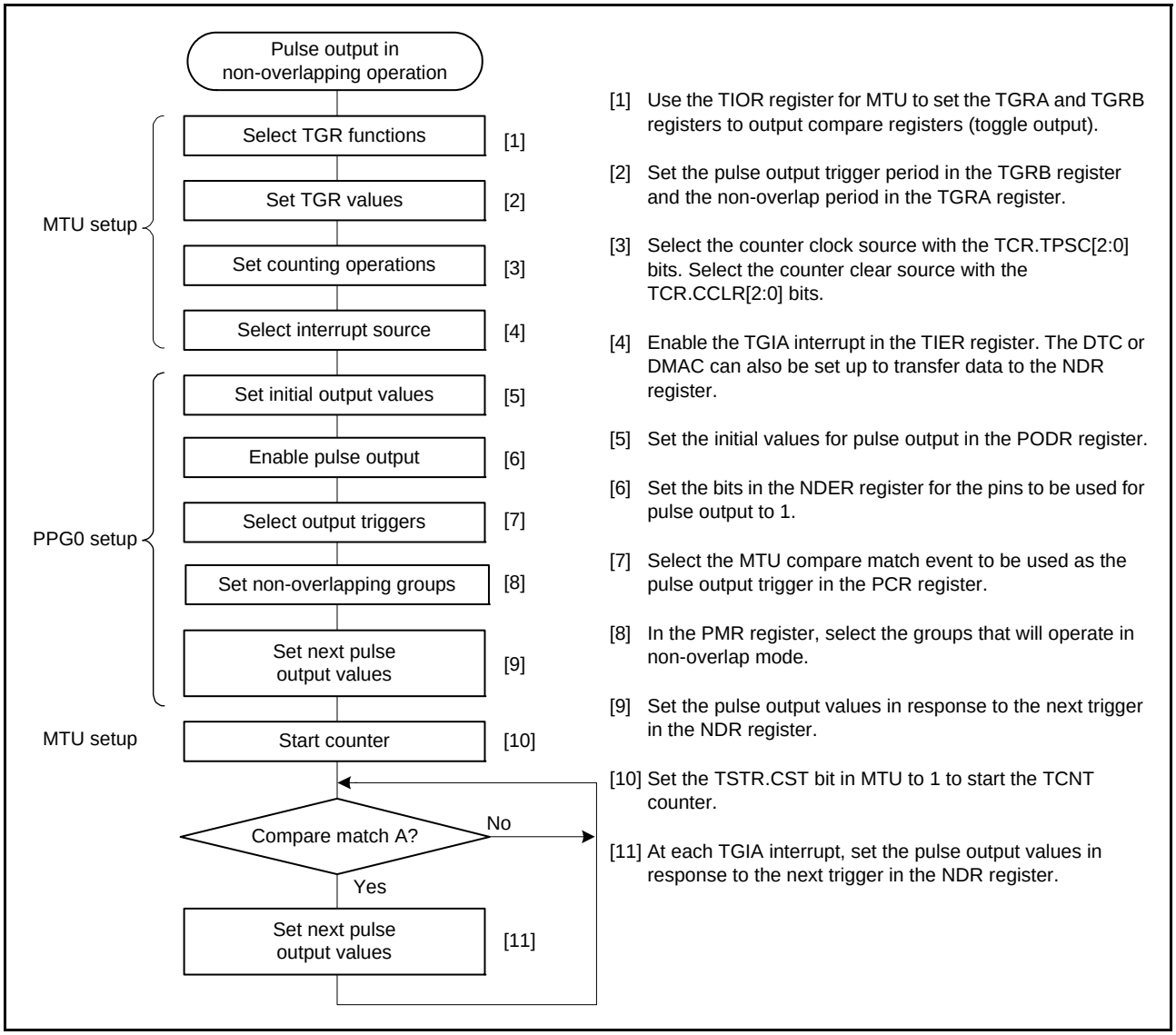
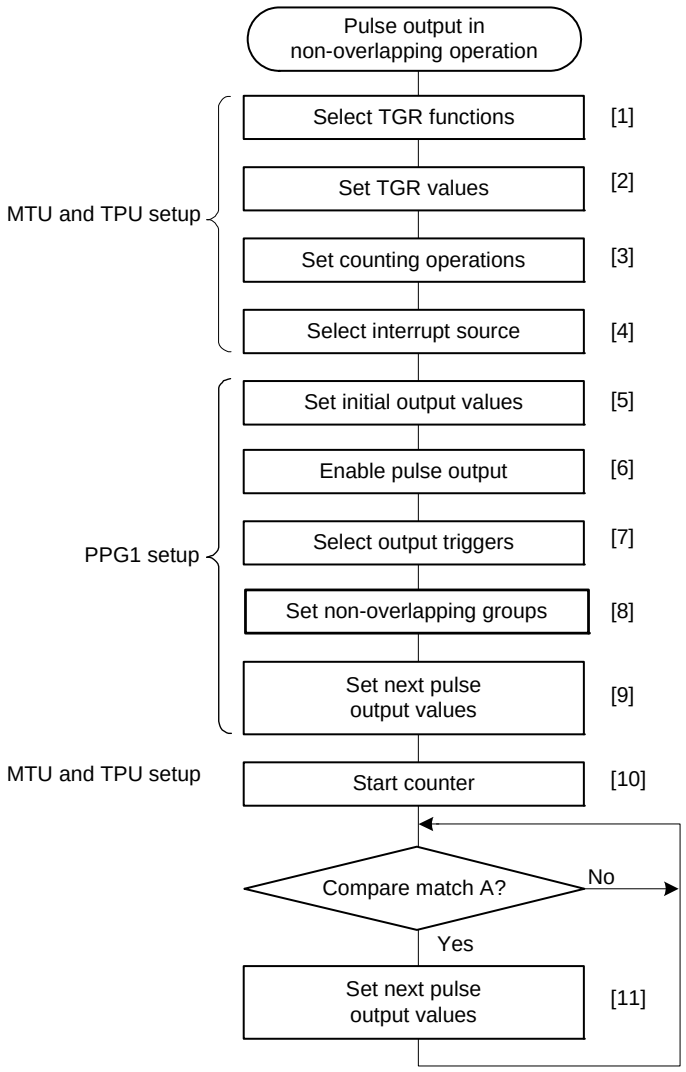


Figure 27.11 Example of Setup Procedure for Pulse Output in Non-Overlapping Operation (PPG0 Setting)

(2) PPG1 Setting



[1] In the case of MTU, use the TIOR register to set the TGRA and TGRB registers to output compare registers (toggle output).
In the case of TPU, use the TIOR register to set the TGRA and TGRB registers to output compare registers (toggle output).

[2] Set the pulse output trigger period in the TGRB register and the non-overlap period in the TGRA register.

[3] Select the counter clock source with the TCR.TPSC[2:0] bits.
Select the counter clear source with the TCR.CCLR[2:0] bits.

[4] Enable the TGIA interrupt in the TIER register. The DTC or DMAC can also be set up to transfer data to the NDR register.

[5] Set the initial values for pulse output in the PODR register.

[6] Set the bits in the NDER register for the pins to be used for pulse output to 1.

[7] Select the MTU/TPU compare match event to be used as the pulse output trigger in the PTRSLR and PCR registers.

[8] In the PMR register, select the groups that will operate in non-overlap mode.

[9] Set the pulse output values in response to the next trigger in the NDR register.

[10] Set the TSTR.CST bit to 1 to start the TCNT counter.

[11] At each TGIA interrupt, set the pulse output values in response to the next trigger in the NDR register.

Figure 27.12 Example of Setup Procedure for Pulse Output in Non-Overlapping Operation (PPG1 Setting)

27.3.6 Example of Pulse Output in Non-Overlapping Operation (Example of Four-Phase Complementary Non-Overlapping Output)

Figure 27.13 shows an example in which pulse output from the PPG0 is used for four-phase complementary non-overlapping pulse output.

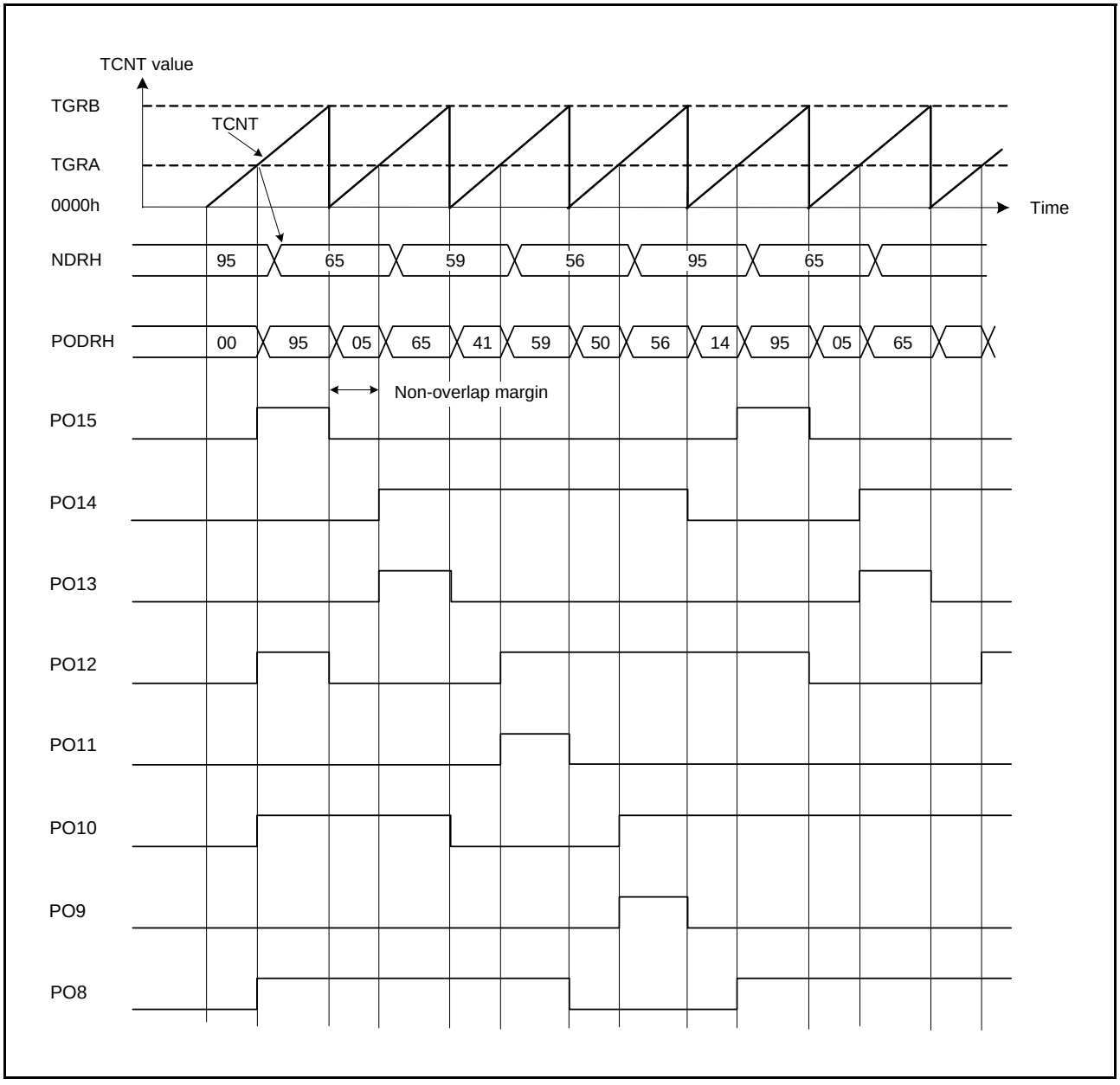


Figure 27.13 Example of Pulse Output in Non-Overlapping Operation (Four-Phase Complementary Non-Overlapping Output)

1. Set output compare registers of the MTUn.TGRA and MTUn.TGRB (n = 0 to 3) 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 the PPG0.NDERH register, 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 the PPG0.NDRH register.
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 the PPG0.NDRH register.
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.

27.3.7 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 27.14 shows the outputs when the G3INV and G2INV bits are cleared to 0 in addition to the settings in Figure 27.13.

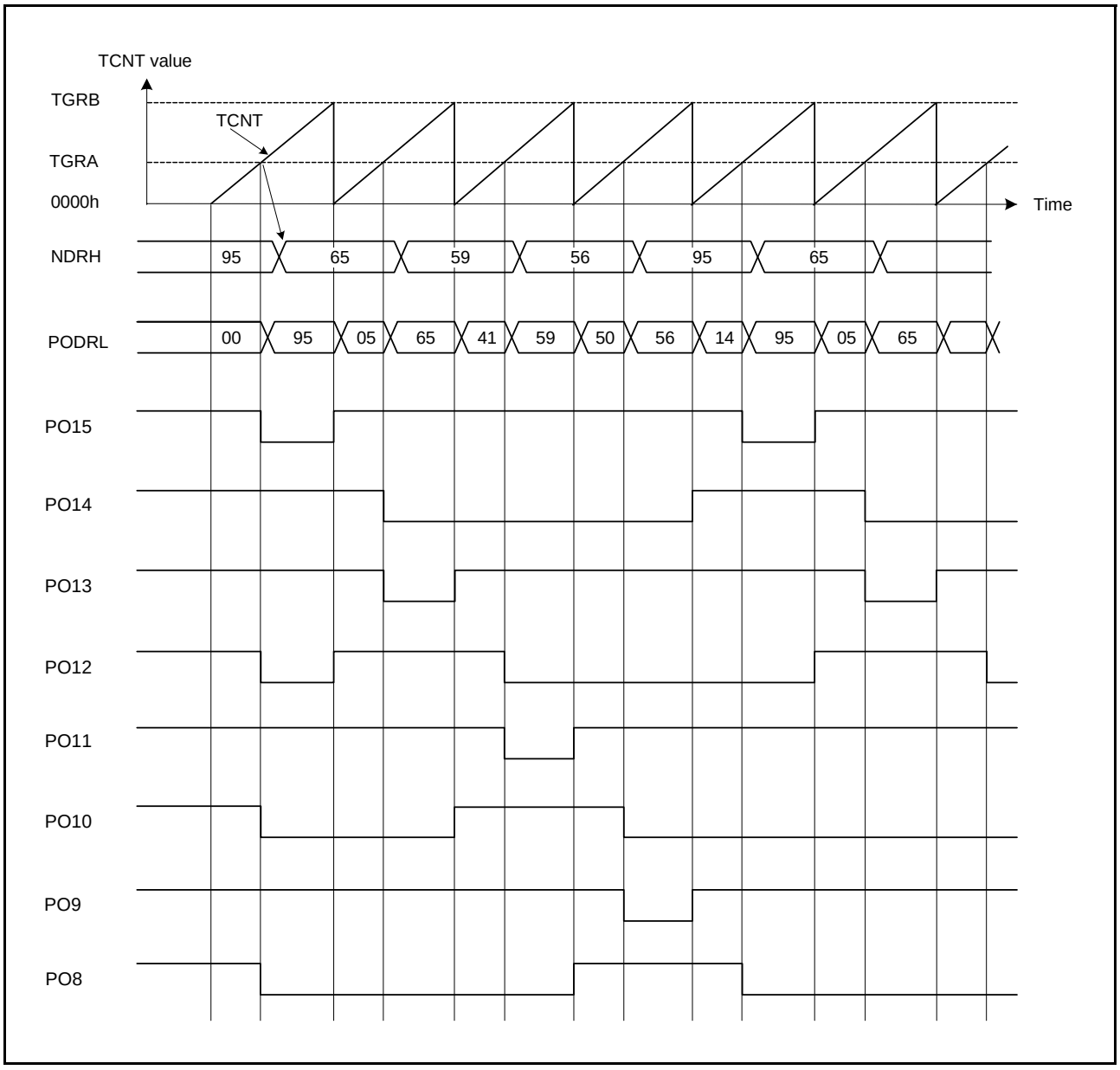


Figure 27.14 Inverted Pulse Output (Example)

27.3.8 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 selected by PPG0.PCR, pulse output is triggered by the input capture signal.

Figure 27.15 shows the timing of pulse output triggered by input capture.

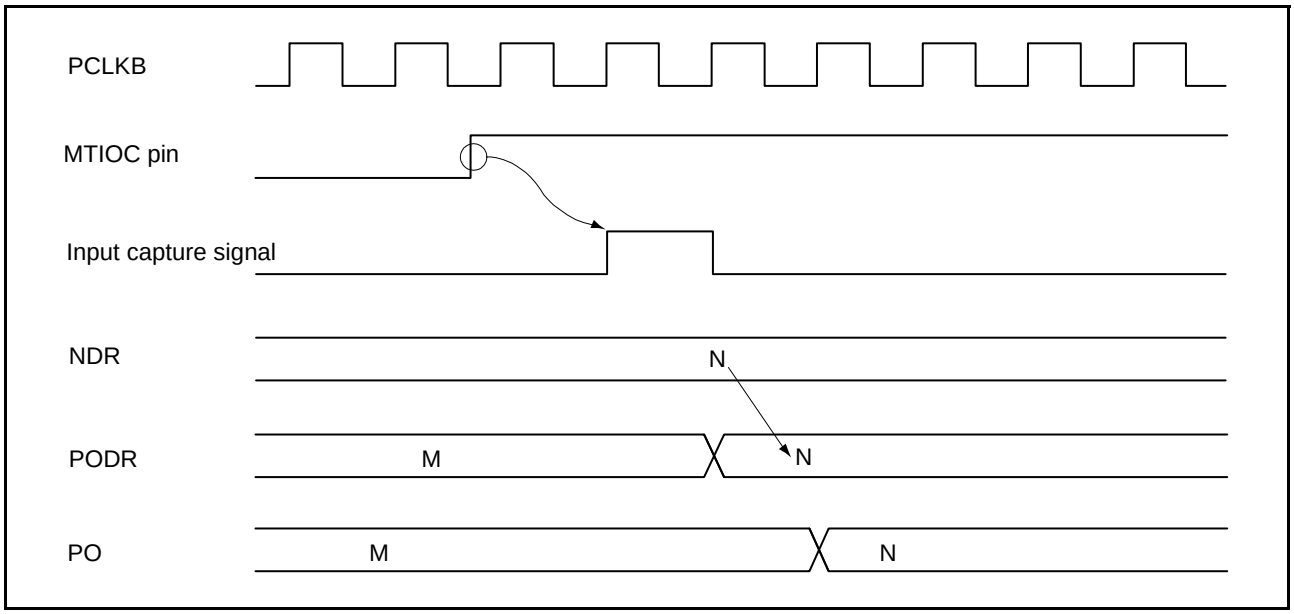


Figure 27.15 Timing of Pulse Output Triggered by Input Capture (Example)

27.4 Usage Note

27.4.1 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, refer to section 11, Low Power Consumption.

28. 8-Bit Timer (TMRb)

This MCU has two units (unit 0, unit 1) of an on-chip 8-bit timer (TMR) module that comprise two 8-bit counter channels, totaling four channels. The 8-bit timer module can be used to count external events and also be used as a multi-function timer in a variety of applications, such as generation of counter reset signal, interrupt requests, and pulse output with a desired duty cycle using a compare-match signal with two registers.
Unit 0 and unit 1 have the same functions, and can generate a base clock for the SCI and an operating clock for the remote control signal receiver (REMC).
In this section, “PCLK” is used to refer to PCLKB.

28.1 Overview

Table 28.1 lists the specifications of the TMR. Table 28.2 lists the TMR functions.
Figure 28.1 shows a block diagram of the 8-bit timer module (unit 0), and Figure 28.2 shows that of the 8-bit timer module (unit 1).

Table 28.1 Specifications of TMR

Item	Description
Count clock	• Internal clock: PCLK/1, PCLK/2, PCLK/8, PCLK/32, PCLK/64, PCLK/1024, PCLK/8192 • External clock: external count clock
Number of channels	(8 bits × 2 channels) × 2 units
Compare match	• 8-bit mode (compare match A, compare match B) • 16-bit mode (compare match A, compare match B)
Counter clear	Selected by compare match A or B, or an external counter reset signal.
Timer output	Output pulses with a desired duty cycle or PWM output
Cascading of two channels	• 16-bit count mode 16-bit timer using TMR0 for the upper 8 bits and TMR1 for the lower 8 bits (TMR2 for the upper 8 bits and TMR3 for the lower 8 bits) • Compare match count mode TMR1 can be used to count TMR0 compare matches (TMR3 can be used to count TMR2 compare matches).
Interrupt sources	Compare match A, compare match B, and overflow
Event link function (Output)	Compare match A, compare match B, and overflow (TMR0 to TMR3)
Event link function (Input)	One of the following three operations proceeds in response to an event reception: (1) Counting start operation (TMR0 to TMR3) (2) Event counting operation (TMR0 to TMR3) (3) Counting restart operation (TMR0 to TMR3)
DTC activation	DTC can be activated by compare match A interrupts or compare match B interrupts.
A/D conversion start trigger of the A/D converter	Compare match A of TMR0 and TMR2
Capable of generating base clock for SCI	Generates base clock for SCI.*1
Capable of generating operating clock for REMC	Generates operating clock for remote control signal receiver (REMC)*2
Low power consumption function	Each unit can be placed in a module stop state

Note 1. For details, refer to section 35, Serial Communications Interface (SCIk, SCIm, SCIh).
Note 2. For details, refer to section 46, Remote Control Signal Receiver (REMCa).

Table 28.2 TMR Functions

Item		Unit 0			Unit 1		
Counter mode		**8 Bits**		**16 Bits**	**8 Bits**		**16 Bits**
Channel		**TMR0**	**TMR1**	**TMR0 + TMR1**	**TMR2**	**TMR3**	**TMR2 + TMR3**
Count clock		PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI0	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI1	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI1	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI2	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI3	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMCI3
Counter clear		TMR0.TCORA TMR0.TCORB TMRI0	TMR1.TCORA TMR1.TCORB TMRI1	TMR0.TCORA + TMR1.TCORA TMR0.TCORB + TMR1.TCORB TMRI0	TMR2.TCORA TMR2.TCORB TMRI2	TMR3.TCORA TMR3.TCORB TMRI3	TMR2.TCORA + TMR3.TCORA TMR2.TCORB + TMR3.TCORB TMRI2
Compare match	Compare match A	✓	✓	✓	✓	✓	✓
	Compare match B	✓	✓	✓	✓	✓	✓
Timer output	Low output	✓	✓	✓	✓	✓	✓
	High output	✓	✓	✓	✓	✓	✓
	Toggle output	✓	✓	✓	✓	✓	✓
DTC activation	Compare match A	✓	✓	✓	✓	✓	✓
	Compare match B	✓	✓	✓	✓	✓	✓
	TCNT overflow	—	—	—	—	—	—
Interrupt	Compare match A	CMIA0	CMIA1	CMIA0	CMIA2	CMIA3	CMIA2
	Compare match B	CMIB0	CMIB1	CMIB0	CMIB2	CMIB3	CMIB2
	TCNT overflow	OVI0	OVI1	OVI0	OVI2	OVI3	OVI2
Cascaded connection		TMR1 overflow	TMR0 compare match A	—	TMR3 overflow	TMR2 compare match A	—
A/D conversion start trigger of the A/D converter*1		✓	—	✓	✓	—	✓
SCI base clock generation*2		✓		—	✓		—
Capable of generating operating clock for REMC*3		✓	—	—	—	—	—
ELC output event	Compare match A	✓	✓	✓	✓	✓	✓
	Compare match B	✓	✓	✓	✓	✓	✓
	TCNT overflow	✓	✓	✓	✓	✓	✓
ELC input event	Counting start	✓	✓	—	✓	✓	—
	Event counting	✓	✓	—	✓	✓	—
	Counting restart	✓	✓	—	✓	✓	—
Module stop setting*4		MSTPCRA.MSTPA5 bit (unit 0), MSTPCRA.MSTPA4 bit (unit 1)					

✓: Possible

—: Impossible

Note 1. For details, refer to section 50, 12-Bit A/D Converter (S12ADFa).
Note 2. For details, refer to section 35, Serial Communications Interface (SCIk, SCIm, SCIh).
Note 3. For details, refer to section 46, Remote Control Signal Receiver (REMCa).
Note 4. For details, refer to section 11, Low Power Consumption.

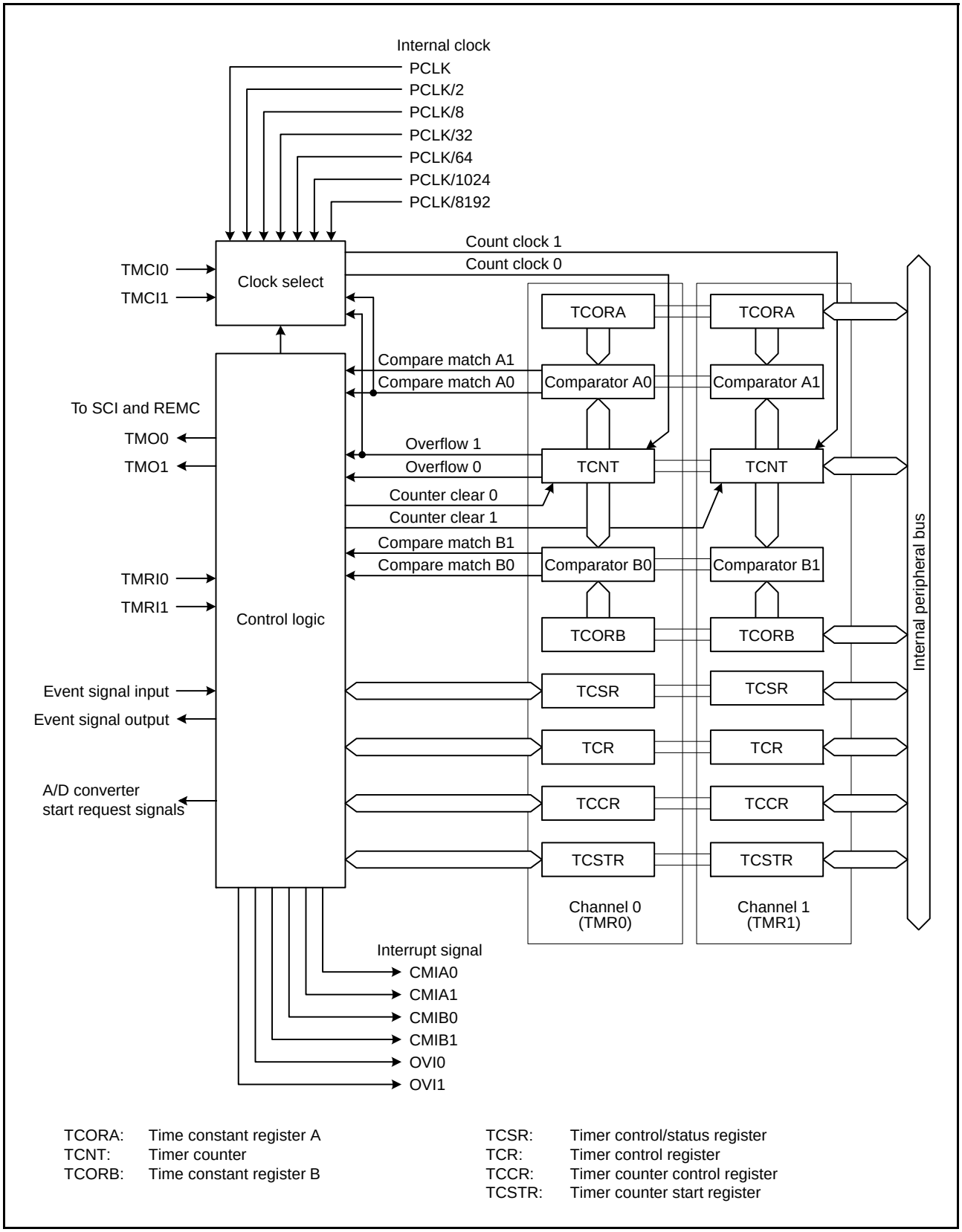


Figure 28.1 Block Diagram of TMR (Unit 0)

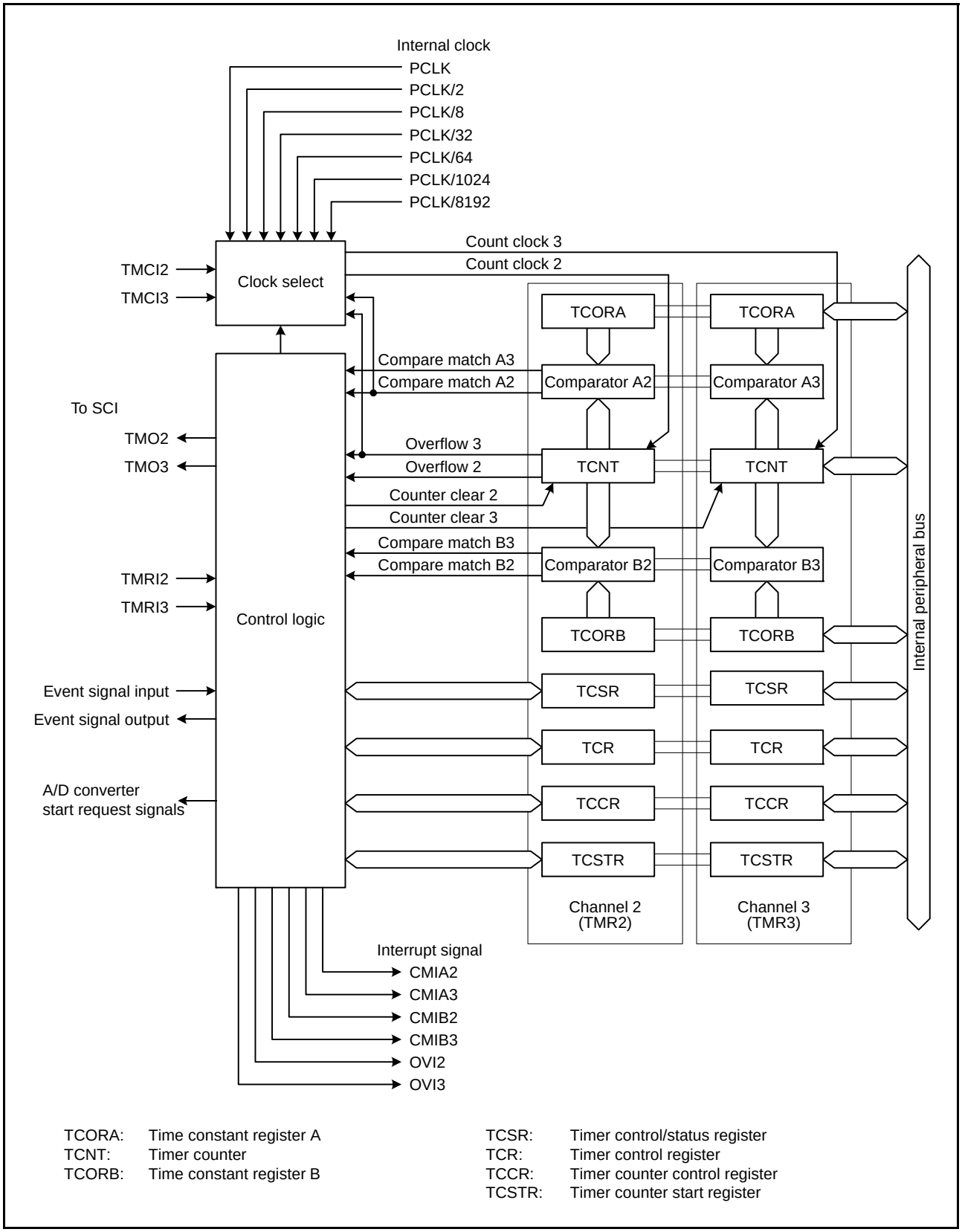


Figure 28.2 Block Diagram of TMR (Unit 1)

Table 28.3 lists the I/O pins of the TMR.

Table 28.3 Pin Configuration of TMR

Unit	Channel	Pin Name	I/O	Description
0	TMR0	TMO0	Output	Outputs compare match
		TMCI0	Input	Inputs external count clock
		TMRI0	Input	Inputs external counter reset
	TMR1	TMO1	Output	Outputs compare match
		TMCI1	Input	Inputs external count clock
		TMRI1	Input	Inputs external counter reset
1	TMR2	TMO2	Output	Outputs compare match
		TMCI2	Input	Inputs external count clock
		TMRI2	Input	Inputs external counter reset
	TMR3	TMO3	Output	Outputs compare match
		TMCI3	Input	Inputs external count clock
		TMRI3	Input	Inputs external counter reset

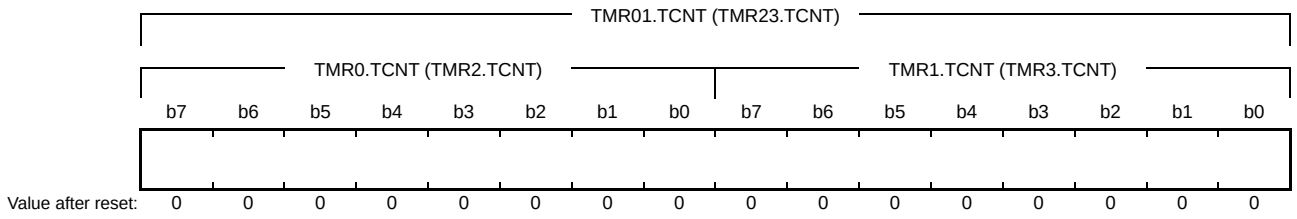
28.2 Register Descriptions

Table 28.4 Register Allocation for 16-Bit Access

Address	Register	Upper 8 Bits	Lower 8 Bits
0008 8208h	TMR01.TCNT	TMR0.TCNT	TMR1.TCNT
0008 8204h	TMR01.TCORA	TMR0.TCORA	TMR1.TCORA
0008 8206h	TMR01.TCORB	TMR0.TCORB	TMR1.TCORB
0008 820Ah	TMR01.TCCR	TMR0.TCCR	TMR1.TCCR
0008 8218h	TMR23.TCNT	TMR2.TCNT	TMR3.TCNT
0008 8214h	TMR23.TCORA	TMR2.TCORA	TMR3.TCORA
0008 8216h	TMR23.TCORB	TMR2.TCORB	TMR3.TCORB
0008 821Ah	TMR23.TCCR	TMR2.TCCR	TMR3.TCCR

28.2.1 Timer Counter (TCNT)

Address(es): TMR0.TCNT 0008 8208h, TMR1.TCNT 0008 8209h, TMR2.TCNT 0008 8218h, TMR3.TCNT 0008 8219h, TMR01.TCNT 0008 8208h, TMR23.TCNT 0008 8218h



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 (TMR01.TCNT, TMR23.TCNT) so they can be accessed together in 16-bit units.

The TCCR.CSS[1:0] and CKS[2:0] bits are used to select a count clock.

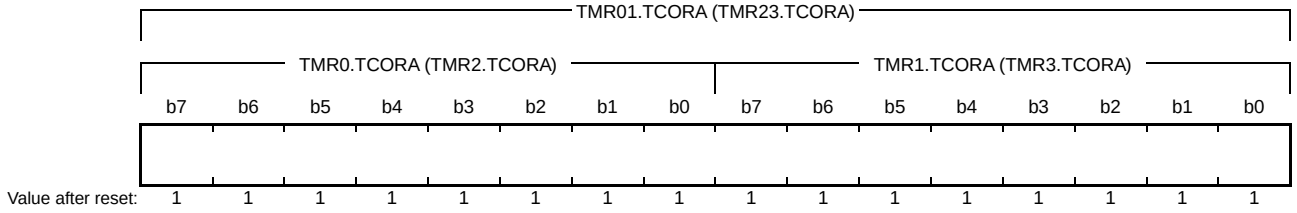
TCNT can be cleared by an external counter reset signal, compare match A, or compare match B. Which compare match to be used for clearing is selected by the TCR.CCLR[1:0] bits.

When TCNT overflows (its value changes from FFh to 00h), an overflow interrupt is output provided the interrupt request is enabled by the TCR.OVIE bit.

For details on the corresponding interrupt vector number, refer to section 15, Interrupt Controller (ICUE), and Table 28.6, TMR Interrupt Sources.

28.2.2 Time Constant Register A (TCORA)

Address(es): TMR0.TCORA 0008 8204h, TMR1.TCORA 0008 8205h, TMR2.TCORA 0008 8214h, TMR3.TCORA 0008 8215h, TMR01.TCORA 0008 8204h, TMR23.TCORA 0008 8214h



TCORA is an 8-bit readable/writable register.

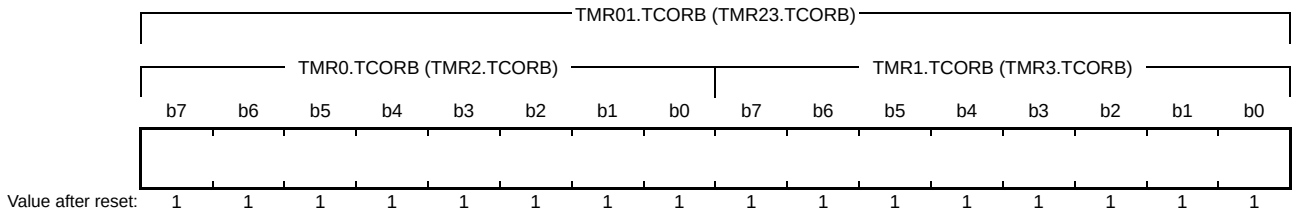
TMR0.TCORA and TMR1.TCORA (TMR2.TCORA and TMR3.TCORA) comprise a single 16-bit register (TMR01.TCORA, TMR23.TCORA) so they can be accessed together in 16-bit units.

The value in TCORA is continually compared with the value in TCNT. When a match is detected, the corresponding compare match A is generated, and a compare match A interrupt is output provided the interrupt request is enabled by the TCR.CMIEA bit.

However, comparison is not performed during writing to TCORA. The timer output from the TMOn pin can be freely controlled by this compare match A and the settings of the TCSR.OSA[1:0] bits.

28.2.3 Time Constant Register B (TCORB)

Address(es): TMR0.TCORB 0008 8206h, TMR1.TCORB 0008 8207h, TMR2.TCORB 0008 8216h, TMR3.TCORB 0008 8217h, TMR01.TCORB 0008 8206h, TMR23.TCORB 0008 8216h



TCORB is an 8-bit readable/writable register.

TMR0.TCORB and TMR1.TCORB (TMR2.TCORB and TMR3.TCORB) comprise a single 16-bit register (TMR01.TCORB, TMR23.TCORB) so they can be accessed together in 16-bit units.

The value in TCORB is continually compared with the value in TCNT. When a match is detected, the corresponding compare match B is generated, and a compare match B interrupt is output provided the interrupt request is enabled by the TCR.CMIEB bit.

However, comparison is not performed during writing to TCORB. The timer output from the TMOn pin can be freely controlled by this compare match B and the settings of the TCSR.OSB[1:0] bits.

28.2.4 Timer Control Register (TCR)

Address(es): TMR0.TCR 0008 8200h, TMR1.TCR 0008 8201h, TMR2.TCR 0008 8210h, TMR3.TCR 0008 8211h

	b7	b6	b5	b4	b3	b2	b1	b0
	CMIEB	CMIEA	OVIE	CCLR[1:0]		—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4, b3	CCLR[1:0]	Counter Clear	b4 b3 0 0: Clearing is disabled 0 1: Cleared by compare match A 1 0: Cleared by compare match B 1 1: Cleared by the external counter reset signal*1 (Select edge or level by the TMRIS bit in TCCR.)	R/W
b5	OVIE	Overflow Interrupt Enable	0: Overflow interrupt requests (OVIn) are disabled 1: Overflow interrupt requests (OVIn) are enabled	R/W
b6	CMIEA	Compare Match A Interrupt Enable	0: Compare match A interrupt requests (CMIAn) are disabled 1: Compare match A interrupt requests (CMIAn) are enabled	R/W
b7	CMIEB	Compare Match B Interrupt Enable	0: Compare match B interrupt requests (CMIBn) are disabled 1: Compare match B interrupt requests (CMIBn) are enabled	R/W

Note 1. To use an external counter reset signal, set the corresponding pin function. For details, refer to section 22, I/O Ports and section 23, Multi-Function Pin Controller (MPC).

CCLR[1:0] Bits (Counter Clear)

Select the condition by which TCNT is cleared.

OVIE Bit (Overflow Interrupt Enable)

Selects whether overflow interrupt requests (OVIn) issued by TCNT are enabled or disabled.

CMIEA Bit (Compare Match A Interrupt Enable)

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 B Interrupt Enable)

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.

28.2.5 Timer Counter Control Register (TCCR)

Address(es): TMR0.TCCR 0008 820Ah, TMR1.TCCR 0008 820Bh, TMR2.TCCR 0008 821Ah, TMR3.TCCR 0008 821Bh, TMR01.TCCR 0008 820Ah, TMR23.TCCR 0008 821Ah

	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 28.5.	R/W
b4, b3	CSS[1:0]	Clock Source Select	See Table 28.5.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	TMRIS	Timer Reset Detection Condition Select	0: Cleared at rising edge of the external counter reset signal 1: Cleared when the external counter reset signal is high	R/W

Note 1. To use an external count clock, set the corresponding pin function. For details, refer to section 22, I/O Ports and section 23, Multi-Function Pin Controller (MPC).

TCCR register is a 8-bit register used to configure the basic operation of the counter. Two TCCR registers can be accessed simultaneously by accessing the address of the even channel TCCR register in 16-bit units.

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 28.5.

TMRIS Bit (Timer Reset Detection Condition Select)

This bit is enabled when the TCR.CCLR[1:0] bits are 11b (cleared by external counter reset signal) and selects the condition for detecting counter reset (level or edge).

Table 28.5 Clock Input to TCNT and Count Condition

Channel	TCCR Register CSS[1:0] b4	CSS[1:0] b3	CKS[2:0] b2	CKS[2:0] b1	CKS[2:0] b0	Description
TMR0 (TMR2)	0	0	—	0	0	Clock input prohibited
					1	Uses external count clock. Counts at rising edge*1.
				1	0	Uses external count clock. Counts at falling edge*1.
					1	Uses external count clock. Counts at both rising and falling edges*1.
	0	1	0	0	0	Uses internal clock. Counts at PCLK.
					1	Uses internal clock. Counts at PCLK/2.
				1	0	Uses internal clock. Counts at PCLK/8.
					1	Uses internal clock. Counts at PCLK/32.
			1	0	0	Uses internal clock. Counts at PCLK/64.
					1	Uses internal clock. Counts at PCLK/1024.
				1	0	Uses internal clock. Counts at PCLK/8192.
					1	Clock input prohibited
	1	0	—	—	—	Setting prohibited
	1	1	—	—	—	Counts at TMR1.TCNT (TMR3.TCNT) overflow signal*2.
TMR1 (TMR3)	0	0	—	0	0	Clock input prohibited
					1	Uses external count clock. Counts at rising edge*1.
				1	0	Uses external count clock. Counts at falling edge*1.
					1	Uses external count clock. Counts at both rising and falling edges*1.
	0	1	0	0	0	Uses internal clock. Counts at PCLK.
					1	Uses internal clock. Counts at PCLK/2.
				1	0	Uses internal clock. Counts at PCLK/8.
					1	Uses internal clock. Counts at PCLK/32.
			1	0	0	Uses internal clock. Counts at PCLK/64.
					1	Uses internal clock. Counts at PCLK/1024.
				1	0	Uses internal clock. Counts at PCLK/8192.
					1	Clock input prohibited
	1	0	—	—	—	Setting prohibited
	1	1	—	—	—	Counts at TMR0.TCNT (TMR2.TCNT) compare match A*2.

Note 1. To use an external count clock, set the corresponding pin function. For details, refer to section 22, I/O Ports and section 23, Multi-Function Pin Controller (MPC).

Note 2. If the clock input of TMR0 (TMR2) is the overflow signal of the TMR1.TCNT (TMR3.TCNT) counter and that of TMR1 (TMR3) is the compare match signal of the TMR0.TCNT (TMR2.TCNT) counter, no TCNT count clock is generated. Do not use this setting.

28.2.6 Timer Control/Status Register (TCSR)

- TMR0.TCSR, TMR2.TCSR

Address(es): TMR0.TCSR 0008 8202h, TMR2.TCSR 0008 8212h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	ADTE	OSB[1:0]		OSA[1:0]	
Value after reset:	x	x	x	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OSA[1:0]	Output Select A*1	b1 b0 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b3, b2	OSB[1:0]	Output Select B*1	b3 b2 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b4	ADTE	A/D Trigger Enable	0: A/D conversion start request in response to compare match A is disabled. 1: A/D conversion start request in response to compare match A is enabled.	R/W
b7 to b5	—	Reserved	These bits are read as an undefined value. The write value should be 1.	R/W

Note 1. When all OSA[1:0] and OSB[1:0] bits are 0, the output enable signal corresponding to the TMOn pin is negated and a request for high-impedance output is issued to the I/O port. Timer output pin is driven low until the first compare-match occurs after a reset when at least one of the OSA[1:0] and OSB[1:0] bits is 1.

OSA[1:0] Bits (Output Select A)

These bits select a method of TMOn pin output when compare match A of TCORA and TCNT occurs.

OSB[1:0] Bits (Output Select B)

These bits select a method of TMOn pin output when compare match B of TCORB and TCNT occurs.

- TMR1.TCSR, TMR3.TCSR

Address(es): TMR1.TCSR 0008 8203h, TMR3.TCSR 0008 8213h

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

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OSA[1:0]	Output Select A*1	b1 b0 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b3, b2	OSB[1:0]	Output Select B*1	b3 b2 0 0: No change 0 1: Low is output 1 0: High is output 1 1: Output is inverted (toggle output)	R/W
b4	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7 to b5	—	Reserved	These bits are read as an undefined value. The write value should be 1.	R/W

Note 1. When all OSA[1:0] and OSB[1:0] bits are 0, the output enable signal corresponding to the TMOn pin is negated and a request for high-impedance output is issued to the I/O port. Timer output pin is driven low until the first compare match occurs after a reset when at least one of the OSA[1:0] and OSB[1:0] bits is 1.

OSA[1:0] Bits (Output Select A)

These bits select a method of TMOn pin output when compare match A of TCORA and TCNT occurs.

OSB[1:0] Bits (Output Select B)

These bits select a method of TMOn pin output when compare match B of TCORB and TCNT occurs.

28.2.7 Timer Counter Start Register (TCSTR)

Address(es): TMR0.TCSTR 0008 820Ch, TMR1.TCSTR 0008 820Dh, TMR2.TCSTR 0008 821Ch, TMR3.TCSTR 0008 821Dh

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

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	TCS	Timer Counter Status	0: Count stopped state in response to ELC. 1: Count start state in response to ELC.	R/W
b7 to b1	—	Reserved	These bits are read as an undefined value. The write value should be 0.	R/W

TCS Bit (Timer Counter Status)

The TCS bit is used to check the state of the timer count in response to ELC.

When this bit is read as 1, it shows the timer start state in response to ELC. When this bit is read as 0, it shows the timer stopped state in response to ELC.

This bit is cleared by writing 0. Do not write 1 to this bit.

The TCS bit is valid only when the count start operation is selected by the ELOPD register of the event controller (ELC). For details, refer to section 28.7, Link Operation by ELC, or section 21, Event Link Controller (ELC).

28.3 Operation

28.3.1 Pulse Output

Figure 28.3 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 is output) and TCSR.OSB[1:0] bits to 01b (low is output), causing the output to change to high at a compare match of TCORA and to low at a compare match of TCORB.

With these settings, the 8-bit timer provides pulses output at a cycle determined by TCORA with a pulse width determined by TCORB. No software intervention is required.
The timer output pin is low after the TCSR.OSA[1:0] or TCSR.OSB[1:0] bits are set until the first compare match occurs after a reset.

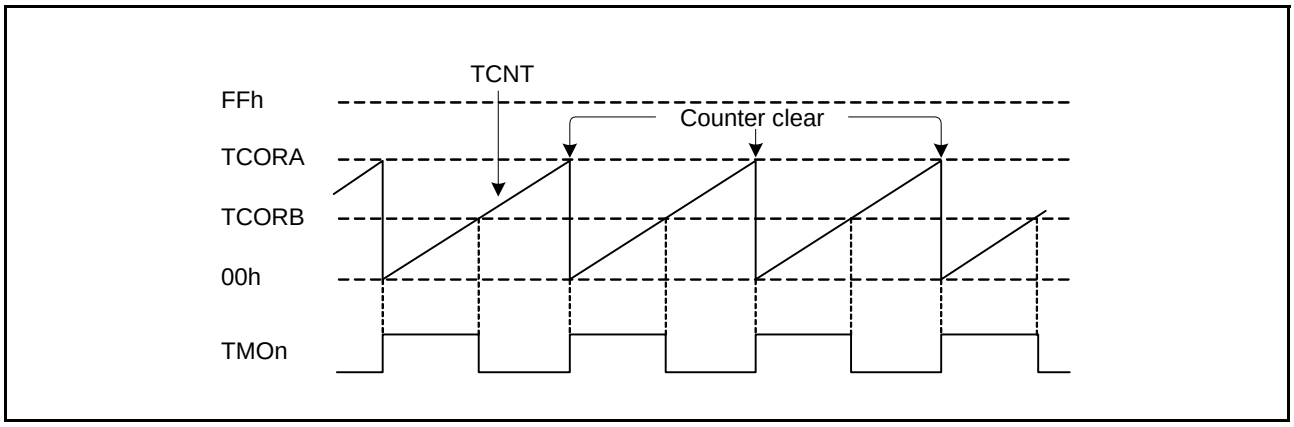


Figure 28.3 Example of Pulse Output (n = 0 to 3)

28.3.2 External Counter Reset Input

Figure 28.4 shows an example of the 8-bit timer being used to generate a pulse which is output after a desired delay time from a TMRIn input.

1. Set the TCR.CCLR[1:0] bits to 11b (cleared by external counter reset signal) and set the TMRIS bit in TCCR to 1 (cleared when the external counter reset signal is high) so that TCNT is cleared at the high level input of the TMRIn signal.
2. Set the TCSR.OSA[1:0] bits to 10b (high output) and the TCSR.OSB[1:0] bits to 01b (low output), causing the output to change to high at a compare match of TCORA and to low at a compare match of TCORB.

With these settings, the 8-bit timer provides pulses output at a desired delay time from a TMRIn input determined by TCORA and with a pulse width determined by TCORB and TCORA.

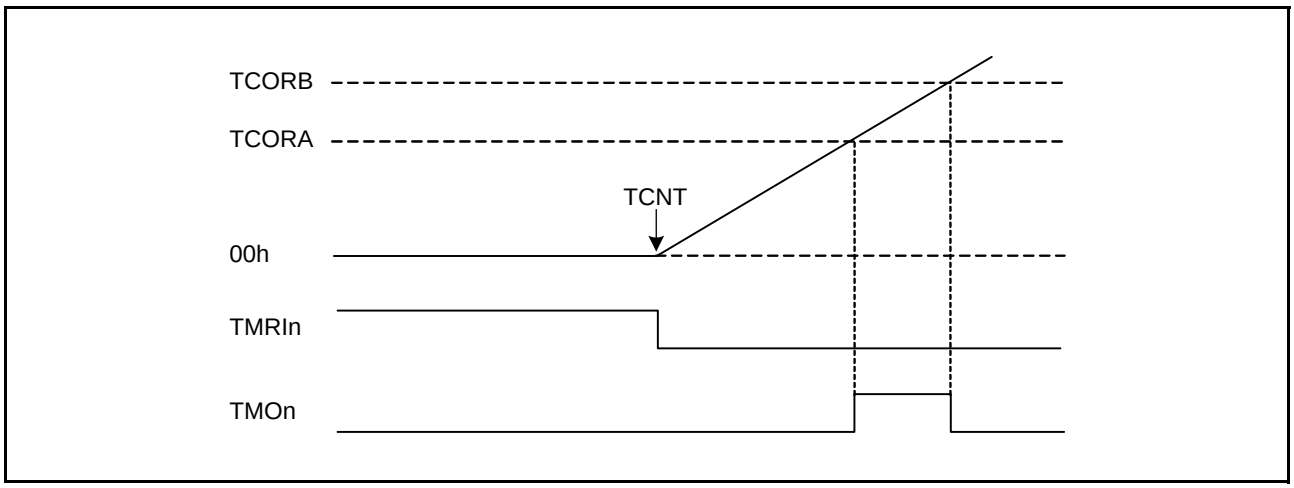


Figure 28.4 Example of External Counter Reset Signal Input (n = 0 to 3)

28.4 Operation Timing

28.4.1 TCNT Count Timing

Figure 28.5 shows the count timing of TCNT for internal clock. Figure 28.6 shows the count timing of TCNT for external clock.

Note that the external clock pulse width must be at least 1.5 PCLK cycles for increment at a single edge, and at least 2.5 PCLK cycles for increment at both edges. The counter will not increment correctly if the pulse width is less than these values.

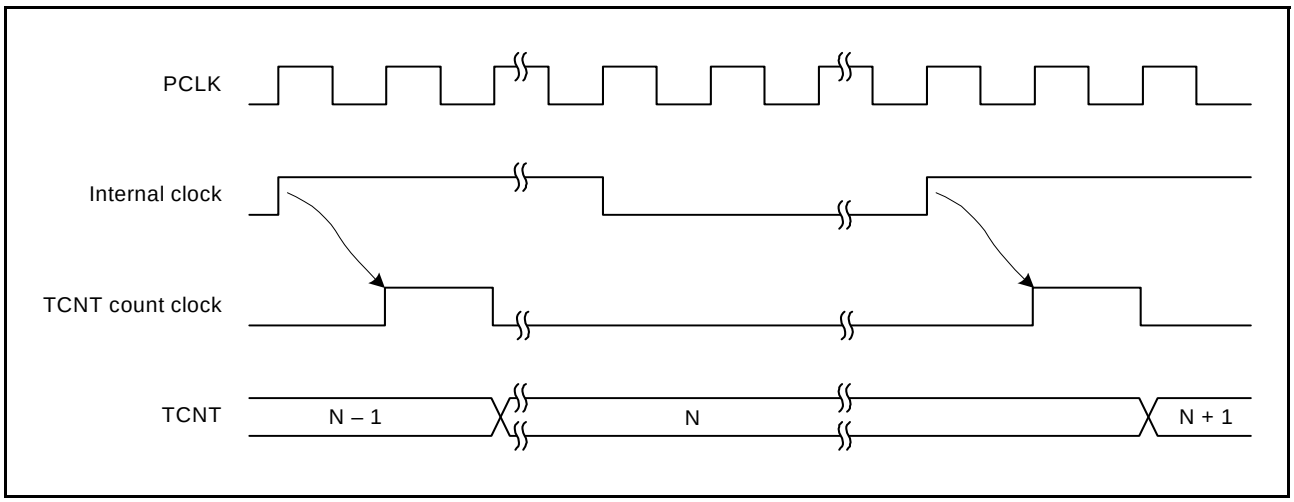


Figure 28.5 Count Timing for Internal Clock

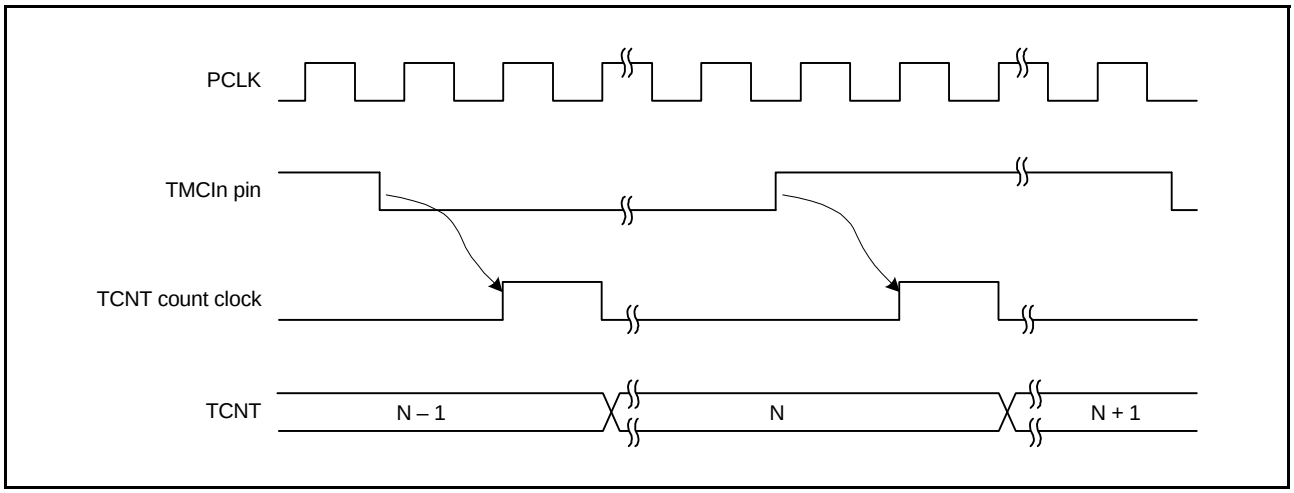


Figure 28.6 Count Timing for External Clock (at Both Edges)

28.4.2 Timing of Interrupt Signal Output on a Compare Match

A compare match refers to a match between the value of the TCORA or TCORB register and the TCNT, and a compare match interrupt signal is output at this time if the interrupt request is enabled. The compare match is generated in the last cycle in which the values match (at the time at which the value counted by TCNT to produce the match is updated). Accordingly, after a match between TCNT and the TCORA or TCORB register is detected, the compare match is not actually generated until the next cycle of the TCNT count clock. Figure 28.7 shows the timing of output of the interrupt signal.
For the corresponding interrupt vector number, refer to section 15, Interrupt Controller (ICUE) and Table 28.6.

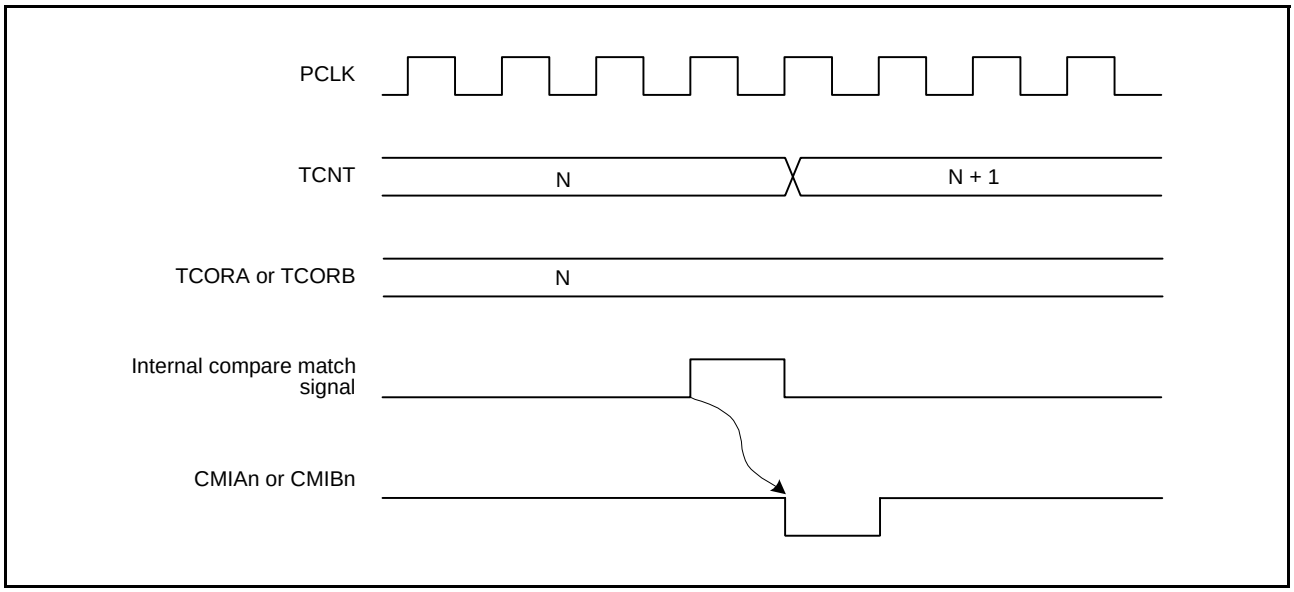


Figure 28.7 Timing of Interrupt Flag Setting to 1 at Compare Match (n = 0 to 3)

28.4.3 Timing of Timer Output Signal at Compare Match

When a compare match signal is generated, the output value specified by the TCSR.OSA[1:0] and OSB[1:0] bits is output on the timer output pin (TMOn).
Figure 28.8 shows the timing when the timer output is toggled by the compare match A signal.

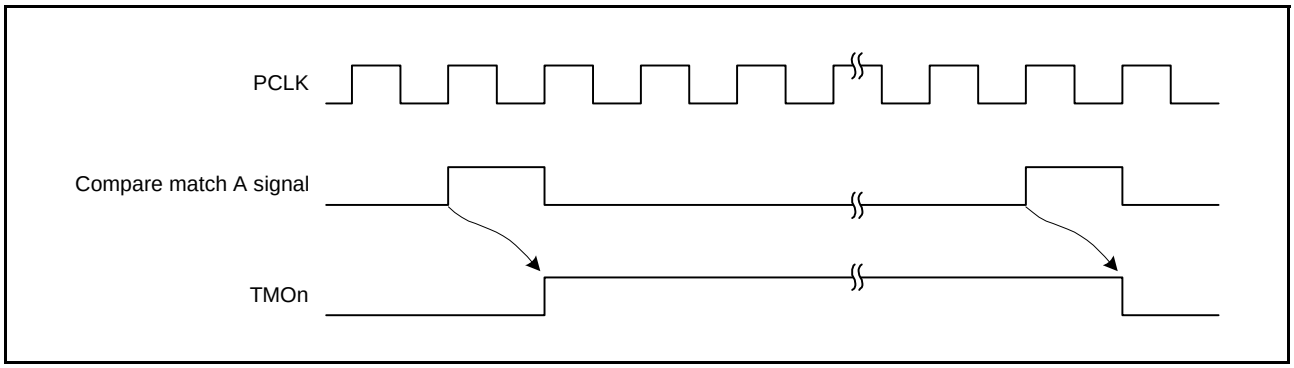


Figure 28.8 Timing of Timer Output Signal at Compare Match A Signal (n = 0 to 3)

28.4.4 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 28.9 shows the timing of this operation.

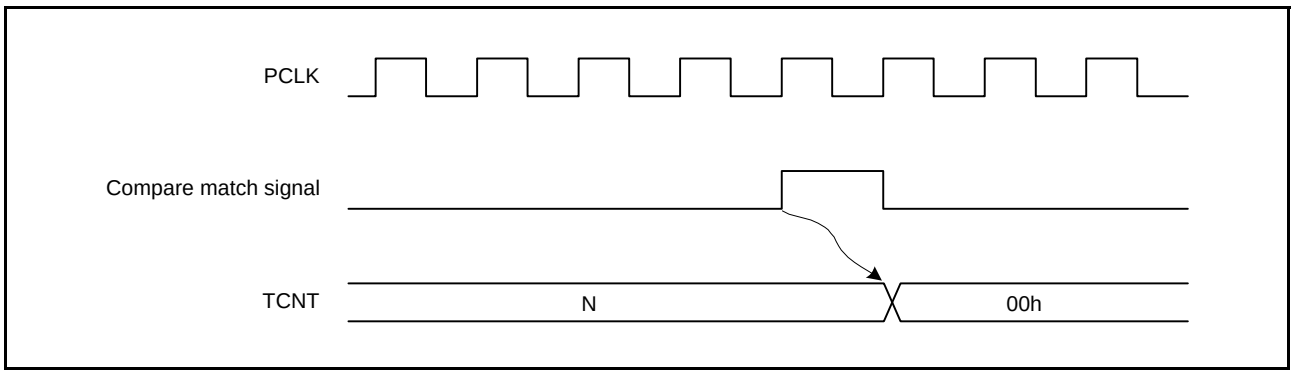


Figure 28.9 Timing of Counter Clear by Compare Match

28.4.5 Timing of the External Reset for TCNT

TCNT is cleared at the rising edge or high level of an external counter reset signal, depending on the settings of the TCR.CCLR[1:0] bits. At least 2 PCLK cycles are required from a reset input to clearing of TCNT. Figure 28.10 and Figure 28.11 show the timing of this operation.

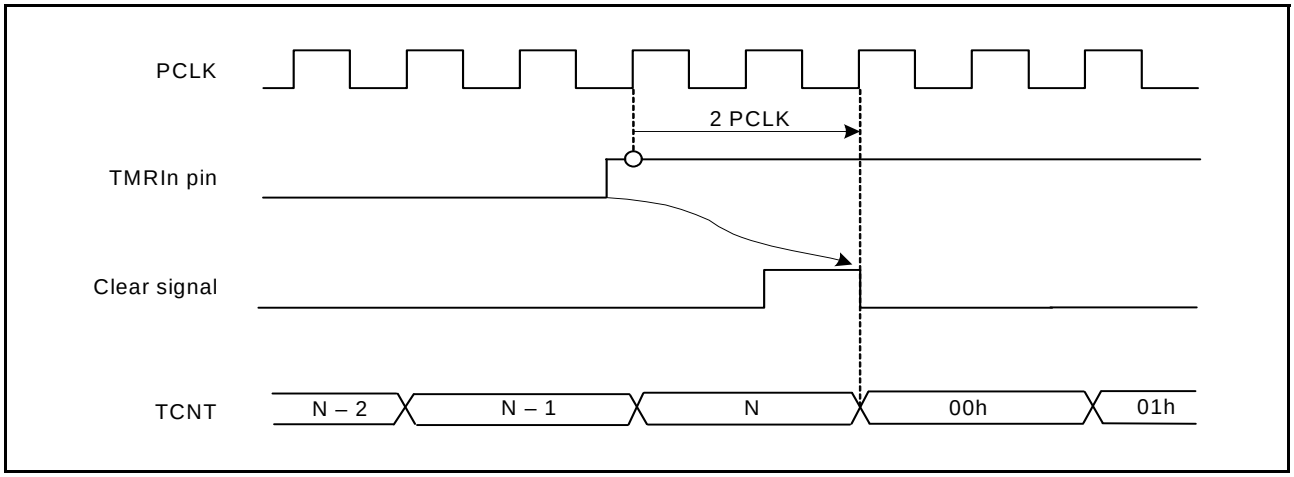


Figure 28.10 Clear Timing by External Counter Reset Signal (Rising Edge)

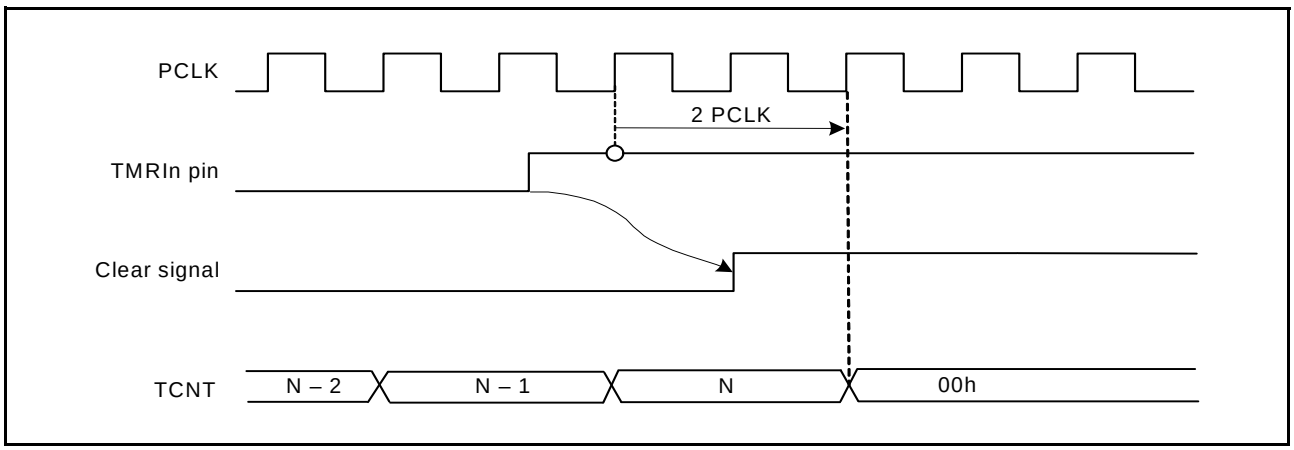


Figure 28.11 Clear Timing by External Counter Reset Signal (High Level)

28.4.6 Timing of Interrupt Signal Output on an Overflow

When TCNT overflows (changes from FFh to 00h), an overflow interrupt signal is output if this interrupt request is enabled.

Figure 28.12 shows the timing of output of the interrupt signal.

For the corresponding interrupt vector number, refer to section 15, Interrupt Controller (ICUE) and Table 28.6.

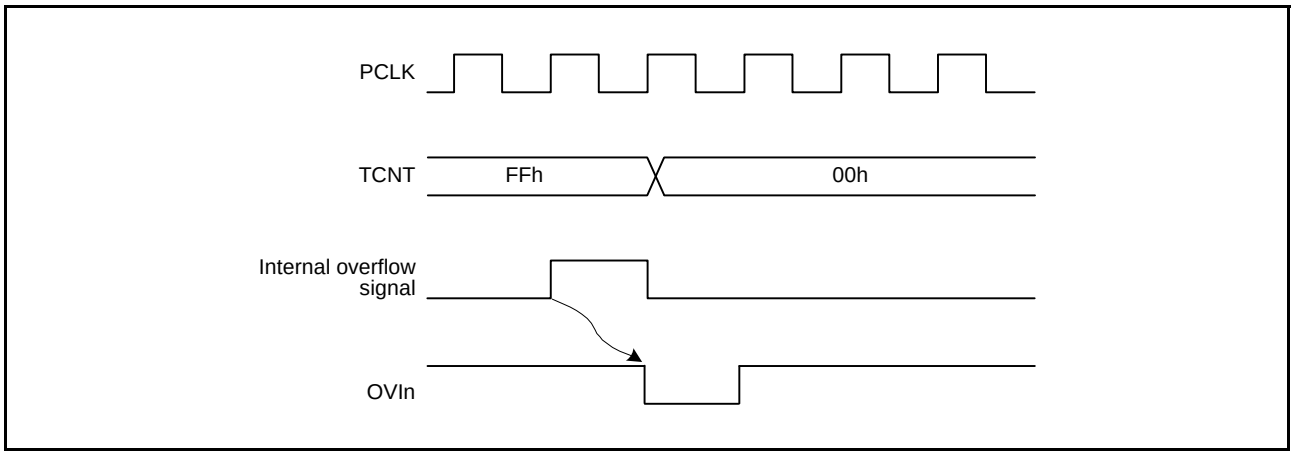


Figure 28.12 Timing of Overflow Interrupt Flag Setting to 1 (n = 0 to 3)

28.5 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).
This section describes unit 0. The operation of unit 1 with cascaded connection is the same as unit 0.

28.5.1 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.

28.5.2 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 pin (n = 0, 1), and counter clear are in accordance with the settings for each channel.

28.6 Interrupt Sources

28.6.1 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 28.6.
It is also possible to activate the DTC by means of CMIAn and CMIBn interrupts.

Table 28.6 TMR Interrupt Sources

Name	Interrupt Sources	DTC Activation
CMIA0	TMR0.TCORA compare match	Possible
CMIB0	TMR0.TCORB compare match	Possible
OVI0	TMR0.TCNT overflow	Not possible
CMIA1	TMR1.TCORA compare match	Possible
CMIB1	TMR1.TCORB compare match	Possible
OVI1	TMR1.TCNT overflow	Not possible
CMIA2	TMR2.TCORA compare match	Possible
CMIB2	TMR2.TCORB compare match	Possible
OVI2	TMR2.TCNT overflow	Not possible
CMIA3	TMR3.TCORA compare match	Possible
CMIB3	TMR3.TCORB compare match	Possible
OVI3	TMR3.TCNT overflow	Not possible

28.6.2 Startup of the A/D Converter

The compare match A of TMR0 and TMR2 allows the A/D converter to be started.
An A/D conversion start request is issued to the A/D converter in response to a generation of compare match A when the TMRn.TCSR.ADTE bit is 1 (i.e., when an A/D conversion request in response to compare match A is enabled). In this case, the conversion trigger for the 8-bit timer should be selected in the A/D converter to start A/D conversion.

Table 28.7 Startup of A/D Converter

A/D Converter	TMR Unit No.	Target	A/D Conversion Start Request
S12AD, S12AD1 (12-bit A/D converter)	0	Compare match between TMR0.TCORA and TMR0.TCNT	TMTRG0AN_0
	1	Compare match between TMR2.TCORA and TMR2.TCNT	TMTRG0AN_1

28.7 Link Operation by ELC

28.7.1 Event Signal Output to ELC

The TMR uses the event link controller (ELC) to perform link operation to the previously specified module using the interrupt request signal as the event signal. The TMR outputs compare match A, compare match B, and overflow signals as event signals. Channels that can be used in this way are TMR0 to TMR3.
The event signal can be output regardless of the setting of the corresponding interrupt request enable bits (TMRn.TCR.OVIE, TMRn.TCR.CMIEA, and TMRn.TCR.CMIEB (n = 0 to 3)). For details, refer to section 21, Event Link Controller (ELC).
The event output function can be used for the cascaded operation.

28.7.2 TMR Operation when Receiving an Event Signal from ELC

The TMR can perform either of the following operations upon the event previously specified by the ELSRn register of the ELC. However, the ELC does not support the cascaded operation.

(1) Count Start

When the TMR count start operation is selected by the ELOPD register of the ELC and the event specified by ELSRn occurs, the TCSTR.TCS bit is set to 1, starting the TMR count operation. After the TMR count start operation is selected by the ELOPD register of the ELC, use the TCCR.CKS[2:0] and CSS[1:0] bits to select the count source.
If the specified event occurs while the TCS bit is 1, the event is ignored.
Write 0 to the TCSTR.TCS bit to stop counting.
When the count start event is input in the count stopped state, the TMR starts counting again according to the CKS[2:0] and CSS[1:0] bits.
The TCS bit is valid only when the ELOPD.TMR0MD[1:0], ELOPD.TMR1MD[1:0], ELOPD.TMR2MD[1:0], and ELOPD.TMR3MD[1:0] bits of the ELC select the count start operation.

(2) Event Count

When the TMR event count operation is selected by the ELOPD register of the ELC and the event specified by ELSRn occurs, the events are counted as the count source regardless of the TCCR.CKS[2:0] and CSS[1:0] bit settings. Reading the counter value returns the number of events that have been actually input.

(3) Count Restart

When the TMR count restart operation is selected by the ELOPD register of the ELC and the event specified by ELSRn occurs, the TCNT counter value is modified to the initial value. If the CKS[2:0] and CSS[1:0] bit settings are not disabling the clock input, the count operation is continued.

28.7.3 Notes on Operating TMR According to an Event Signal from ELC

The following describes the notes on operating the TMR using the event link feature.

(1) Count Start

When the event specified by ELSRn occurs during the write cycle to the TCSTR.TCS bit, the cycle is not completed; setting 1 according to the event occurrence takes priority.

(2) Event Count

When the event specified by ELSRn occurs during the write cycle to the TCNT, the cycle is not completed; event count operation according to the event occurrence takes priority.

(3) Count Restart

When the event specified by ELSRn occurs during the write cycle to the TCNT, the cycle is not completed; count value initialization according to the event occurrence takes priority.

28.8 Usage Notes

28.8.1 Module Stop State Setting

Operation of the TMR can be disabled or enabled by using the module stop control registers. The initial setting is for halting of TMR operation. Register access becomes possible after release from the module stop state. For details, refer to section 11, Low Power Consumption.

28.8.2 Notes on Setting Cycle

If the compare match is selected for counter clear, TCNT is cleared at the last PCLK in the cycle in which the value of TCNT matches with that of TCORA or TCORB. TCNT updates the counter value at this last state. Therefore, the counter frequency is obtained by the following formula (f: Counter frequency, PCLK: Operating frequency, N: TCORA and TCORB register setting value).

$$f = PCLK / (N + 1)$$

28.8.3 Conflict between TCNT Write and Counter Clear

If a counter clear signal is generated concurrently with CPU write to TCNT, the clear takes priority and the write is not performed as shown in Figure 28.13.

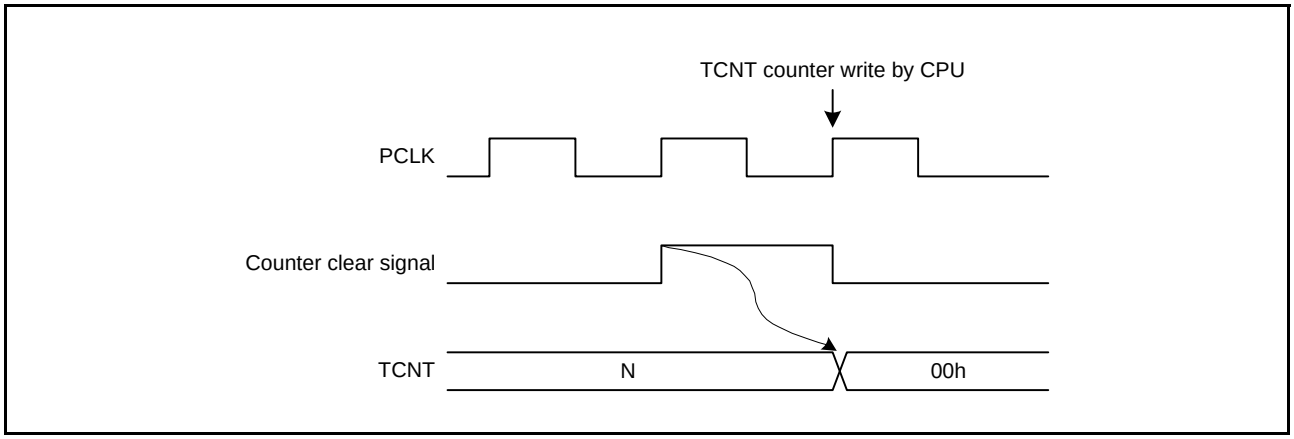


Figure 28.13 Conflict between TCNT Write and Counter Clear

28.8.4 Conflict between TCNT Write and Increment

Even if a counting-up signal is generated concurrently with CPU write to TCNT, the counting-up is not performed and the write takes priority as shown in Figure 28.14.

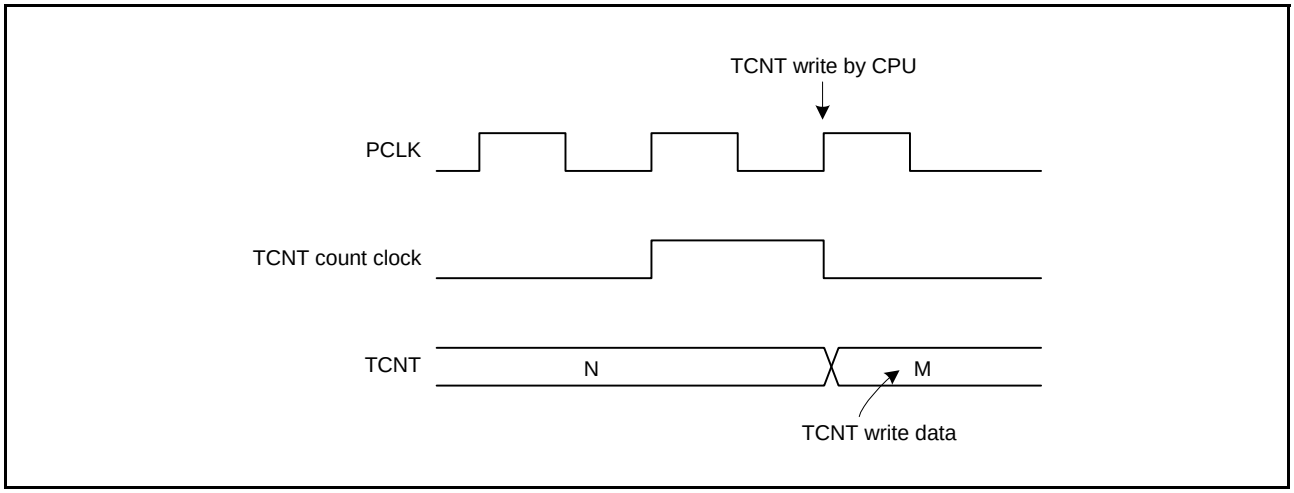


Figure 28.14 Conflict between TCNT Write and Increment

28.8.5 Conflict between TCORA or TCORB Write and Compare Match

Even if a compare match signal is generated simultaneously with CPU write to TCORA or TCORB as shown in Figure 28.15, the write takes priority and the compare match signal does not reach High level.

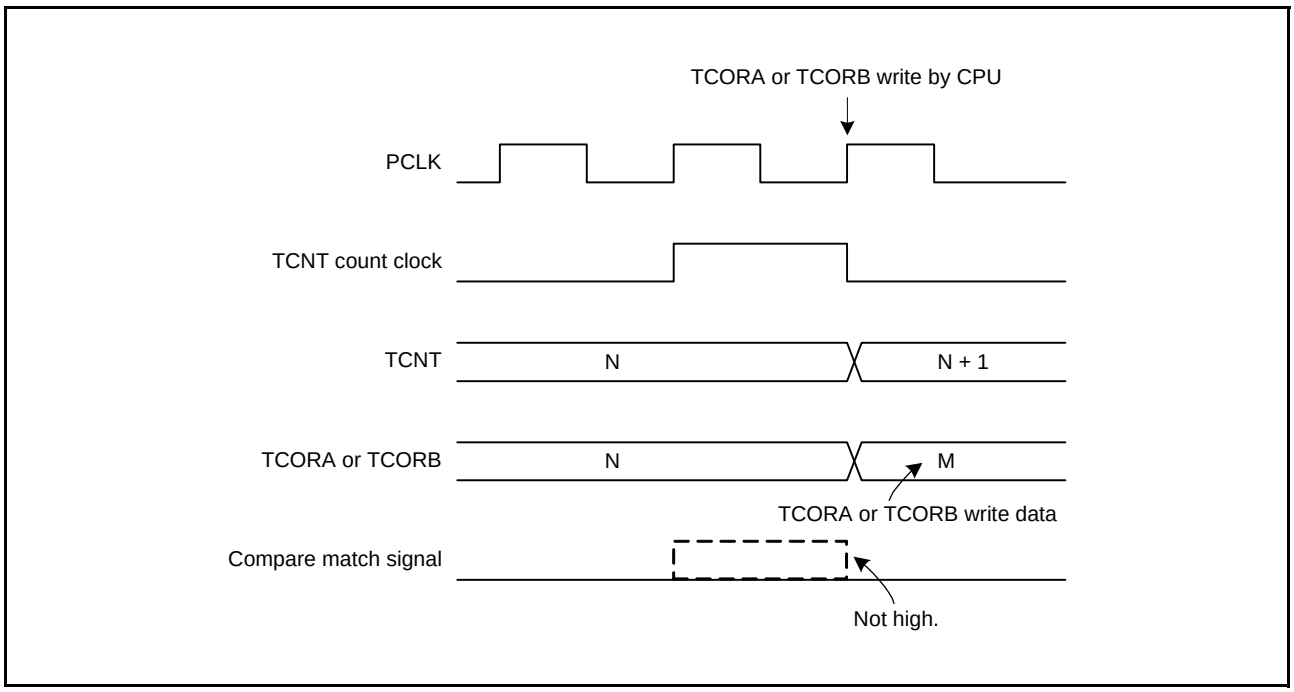


Figure 28.15 Conflict between TCORA or TCORB Write and Compare Match

28.8.6 Conflict between Compare Matches A and B

If compare match events A and B occur at the same time, the 8-bit timer operates in accordance with the priorities for the output methods high for compare match A and compare match B, as listed in Table 28.8.

Table 28.8 Timer Output Priorities

Output Setting	Priority
Toggle output	High
High output	↑
Low output	
No change	Low

28.8.7 Switching of Internal Clocks and TCNT Operation

TCNT may be incremented erroneously depending on when the internal clock is switched. Table 28.9 lists the relationship between the timing at which the internal clock is switched (by writing to the TCCR.CKS[2:0] bits) and the operation of TCNT.

When TCNT count clock is generated from an internal clock, the rising edge of the internal clock pulse are always monitored. If the signal levels of the clocks before and after switching change from low to high as shown in No. 2 in Table 28.9, the change is considered as an edge. Therefore, a TCNT count clock is generated and TCNT is incremented. The erroneous increment of TCNT can also happen when switching between internal and internal clocks.

Table 28.9 Switching of Internal Clocks and TCNT Operation (1/2)

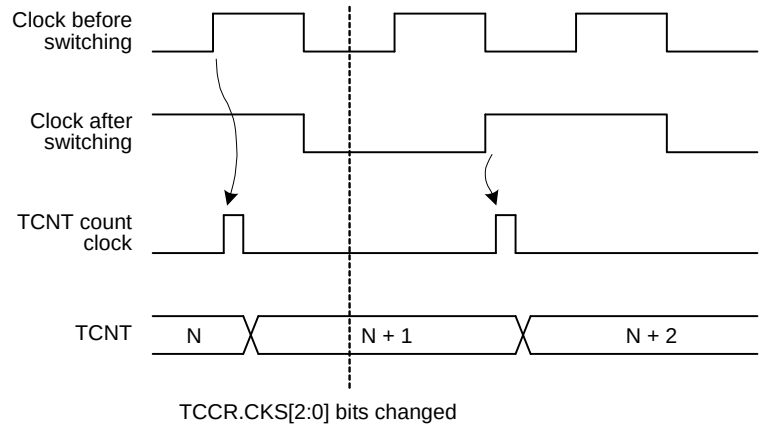
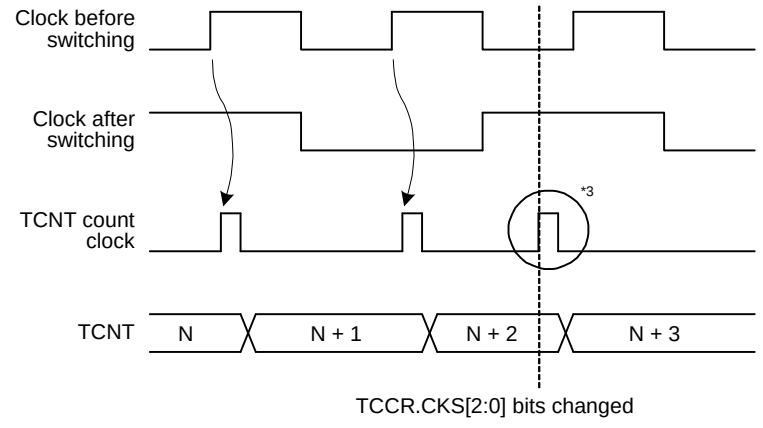
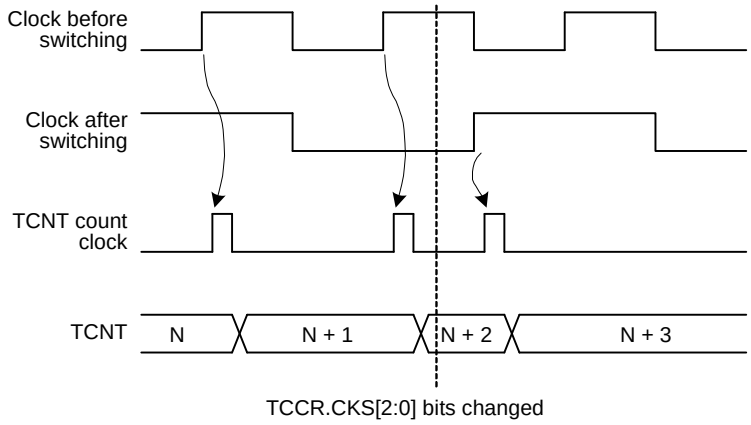
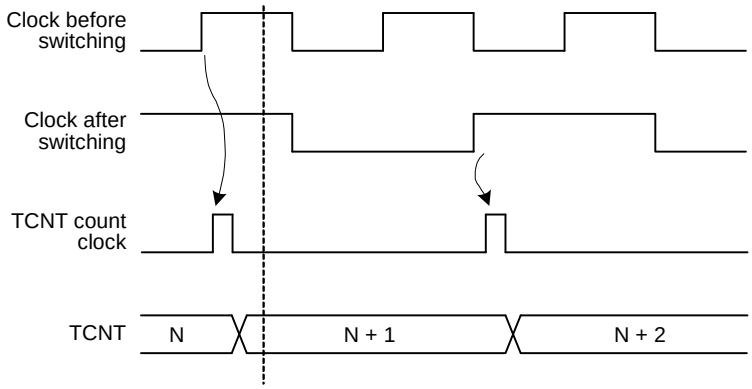
No.	Timing to Change the TCCR.CKS[2:0] Bits	TCNT Counter Operation
1	Switching from low to low*1	
2	Switching from low to high*2	

Table 28.9 Switching of Internal Clocks and TCNT Operation (2/2)

No.	Timing to Change the TCCR.CKS[2:0] Bits	TCNT Counter Operation
3	Switching from high to low*4	
4	Switching from high to high	

Note 1. Includes switching from low to stop, and from stop to low.
Note 2. Includes switching from stop to high.
Note 3. Generated because the change of the signal levels is considered as an edge; TCNT counter is incremented.
Note 4. Includes switching from high to stop.

28.8.8 Clock Source Setting with Cascaded Connection

If 16-bit counter mode and compare match count mode are specified at the same time, count clocks for TMR0.TCNT and TMR1.TCNT (TMR2.TCNT and TMR3.TCNT) are not generated, and the counter stops. Do not specify 16-bit counter mode and compare match count mode simultaneously.

28.8.9 Continuous Output of Compare Match Interrupt Signal

When TCORA or TCORB is set to 00h, PCLK/1 is set as the internal clock, and compare match is set as the counter clear source, the TCNT counter remains 00h and is not updated, and a compare match interrupt signal is output continuously to form a flat signal level.
At this time, the interrupt controller cannot detect the second and subsequent interrupts.
Figure 28.16 shows operation timing when the compare match interrupt signal is continuously output.

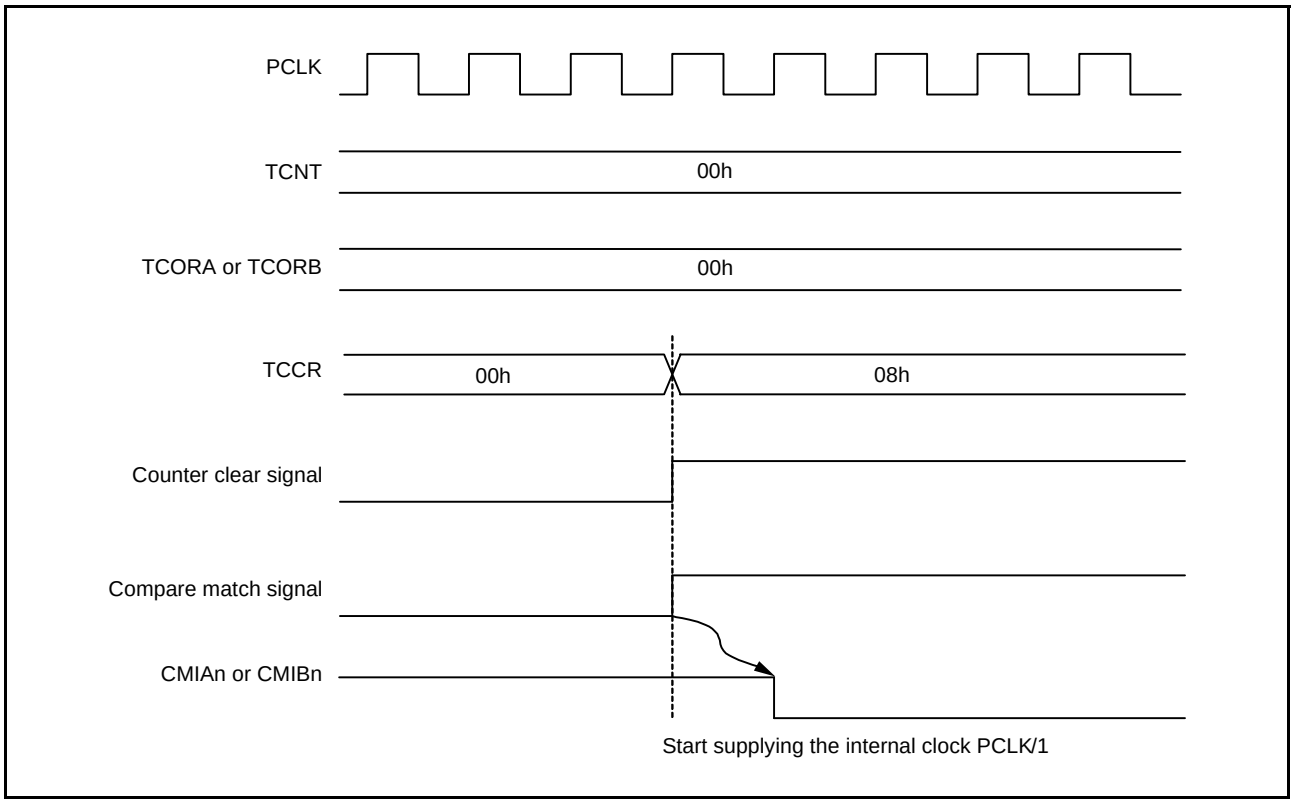


Figure 28.16 Continuous Output of Compare Match Interrupt Signal (n = 0 to 3)

29. Compare Match Timer (CMT)

This MCU has two on-chip compare match timer (CMT) units (unit 0 and unit 1), each consisting of a two-channel 16-bit timer (i.e., a total of four channels). The CMT has a 16-bit counter, and can generate interrupts at set intervals.
In this section, “PCLK” is used to refer to PCLKB.

29.1 Overview

Table 29.1 lists the specifications for the CMT.
Figure 29.1 shows a block diagram of the CMT (unit 0). A two-channel CMT constitutes a unit. Unit 0 and unit 1 are the same in terms of specifications. Compare match timer start register 0 (CMSTR0) and compare match interrupts (CMI0 and CMI1) of unit 0 correspond to compare match timer start register 1 (CMSTR1) and compare match interrupts (CMI2 and CMI3) of unit 1.

Table 29.1 CMT Specifications

Item	Description
Count clocks	• Four frequency dividing clocks One clock from PCLK/8, PCLK/32, PCLK/128, and PCLK/512 can be selected for each channel.
Interrupt	A compare match interrupt can be requested for each channel.
Event link function (output)	An event signal is output upon a CMT1 compare match.
Event link function (input)	Linking to the specified module is possible. CMT1 count start, event counter, or count restart operation is possible.
Low power consumption function	Each unit can be placed in a module stop state.

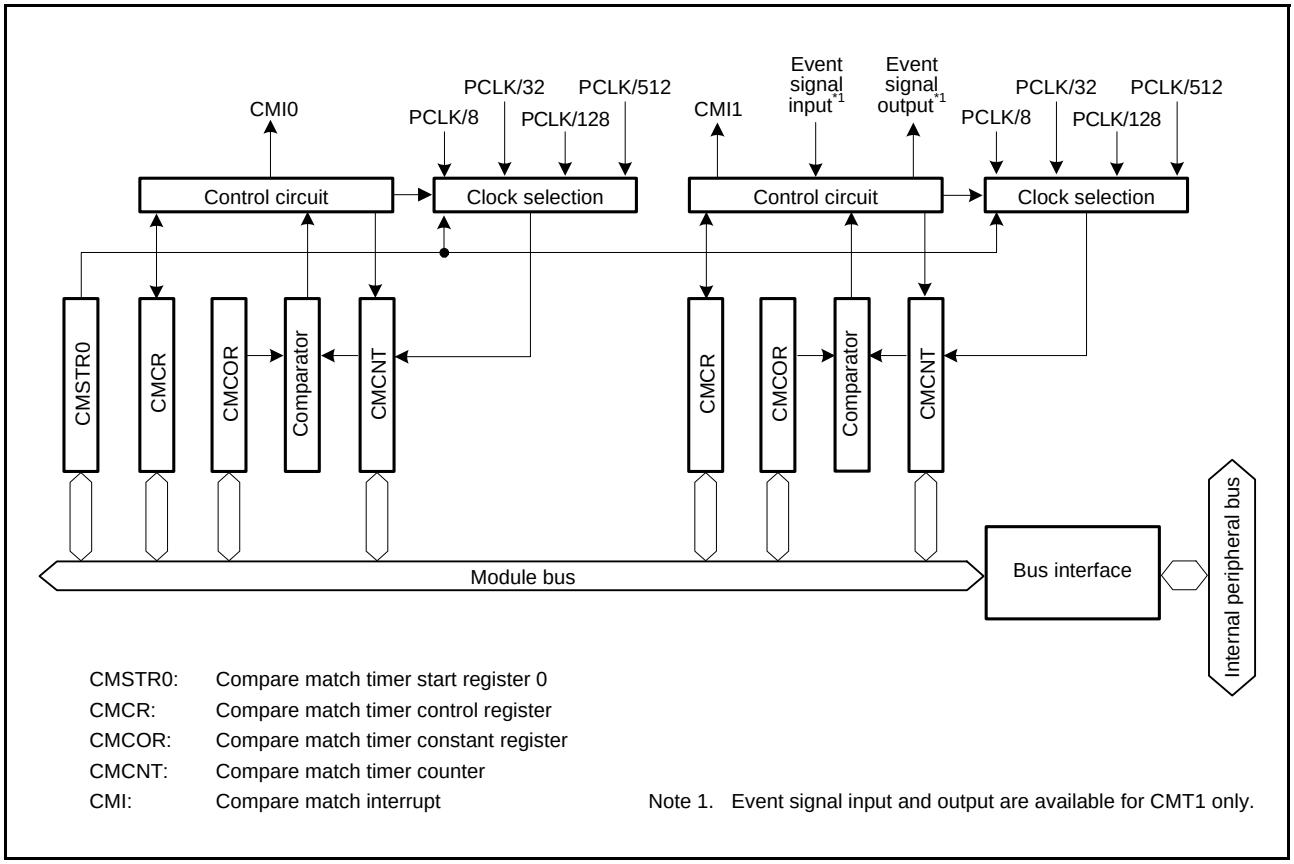


Figure 29.1 CMT (Unit 0) Block Diagram

29.2 Register Descriptions

29.2.1 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

29.2.2 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

29.2.3 Compare Match Timer Control Register (CMCR)

Address(es): CMT0.CMCR 0008 8002h, CMT1.CMCR 0008 8008h, CMT2.CMCR 0008 8012h, CMT3.CMCR 0008 8018h

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

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK/8 0 1: PCLK/32 1 0: PCLK/128 1 1: PCLK/512	R/W
b5 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CMIE	Compare Match Interrupt Enable	0: Compare match interrupt (CMIn) disabled 1: Compare match interrupt (CMIn) enabled	R/W
b7	—	Reserved	This bit is read as undefined. The write value should be 1.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CKS[1:0] Bits (Clock Select)

These bits select the count source from four frequency dividing clocks obtained by dividing the peripheral module clock (PCLK).

When the CMSTRm.STRn (m = 0, 1; n = 0 to 3) bit is set to 1, the CMCNT counter starts counting up on the clock selected with the CKS[1:0] bits.

CMIE Bit (Compare Match Interrupt Enable)

The CMIE bit enables or disables compare match interrupt (CMIn) (n = 0 to 3) generation when the CMCNT counter and the CMCOR register values match.

29.2.4 Compare Match Counter (CMCNT)

Address(es): CMT0.CMCNT 0008 8004h, CMT1.CMCNT 0008 800Ah, CMT2.CMCNT 0008 8014h, CMT3.CMCNT 0008 801Ah

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

The CMCNT counter is a readable/writable up-counter.
When an frequency dividing clock is selected by the CMCR.CKS[1:0] bits and the CMSTRm.STRn (m = 0, 1; n = 0 to 3) bit is set to 1, the CMCNT counter starts counting up using the selected clock.
When the value in the CMCNT counter and the value in the CMCOR register match, the CMCNT counter is set to 0000h. At the same time, a compare match interrupt (CMIn) (n = 0 to 3) is generated.

29.2.5 Compare Match Constant Register (CMCOR)

Address(es): CMT0.CMCOR 0008 8006h, CMT1.CMCOR 0008 800Ch, CMT2.CMCOR 0008 8016h, CMT3.CMCOR 0008 801Ch

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

The CMCOR register is a readable/writable register to set a value for compare match with the CMCNT counter.

29.3 Operation

29.3.1 Periodic Count Operation

When an frequency dividing clock is selected by the CMCR.CKS[1:0] bits and the CMSTRm.STRn (m = 0, 1; n = 0 to 3) bit is set to 1, the CMCNT counter starts counting up using the selected clock.
When the value in the counter and the value in the register match, a compare match interrupt (CMIn) (n = 0 to 3) is generated. The CMCNT counter then starts counting up again from 0000h. Figure 29.2 shows the operation of the CMCNT counter.

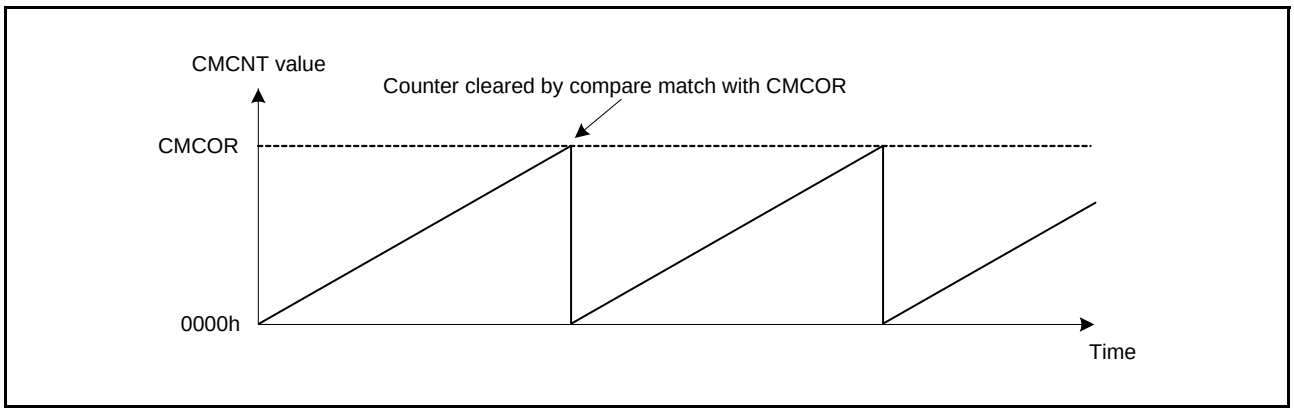


Figure 29.2 CMCNT Counter Operation

29.3.2 CMCNT Count Timing

As the count clock to be input to the CMCNT counter, one of four frequency dividing clocks (PCLK/8, PCLK/32, PCLK/128, and PCLK/512) obtained by dividing the peripheral module clock (PCLK) can be selected with the CMCR.CKS[1:0] bits. Figure 29.3 shows the timing of the CMCNT counter.

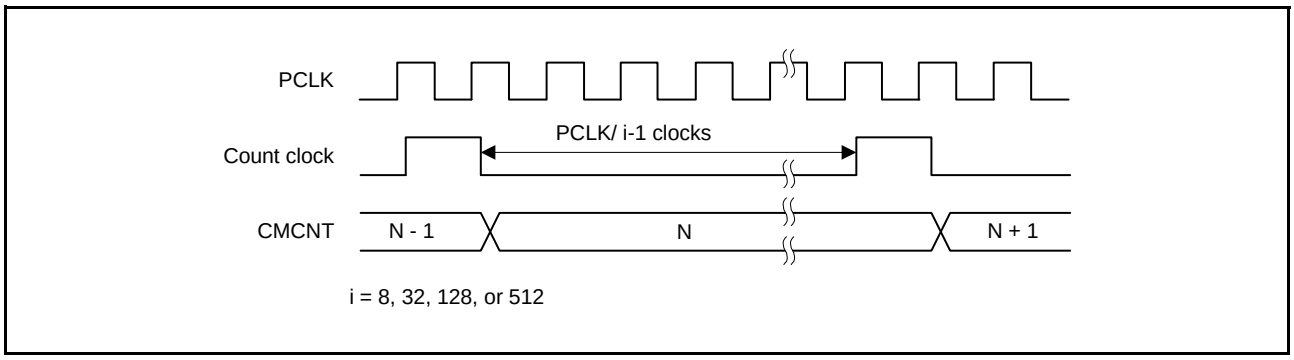


Figure 29.3 CMCNT Count Timing

29.4 Interrupts

29.4.1 Interrupt Sources

The CMT has channels and each of them to which a different vector address is allocated has a compare match interrupt (CMIn) (n = 0 to 3). When a compare match interrupt occurs, the corresponding interrupt request is output.
When the interrupt request is used to generate a CPU interrupt, the priority of channels can be changed by the interrupt controller settings. For details, see section 15, Interrupt Controller (ICUE).

Table 29.2 CMT Interrupt Sources

Name	Interrupt Sources	DTC Activation	DMAC Activation
CMI0	Compare match in CMT0	Possible	Possible
CMI1	Compare match in CMT1	Possible	Possible
CMI2	Compare match in CMT2	Possible	Possible
CMI3	Compare match in CMT3	Possible	Possible

29.4.2 Timing of Compare Match Interrupt Generation

When the CMCNT counter and the CMCOR register match, a compare match interrupt (CMIn) (n = 0 to 3) is generated. A compare match signal is generated at the last state in which the values match (the timing when the CMCNT counter updates the matched count value). That is, after a match between the CMCOR register and the CMCNT counter, the compare match signal is not generated until the next the CMCNT counter input clock.
Figure 29.4 shows the timing of a compare match interrupt.

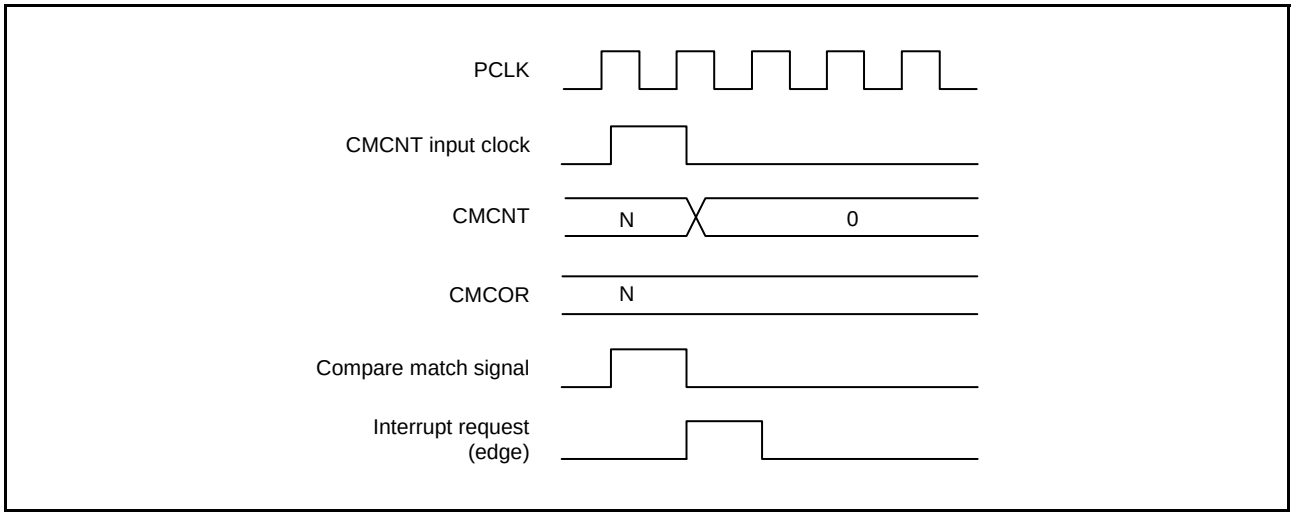


Figure 29.4 Timing of a Compare Match Interrupt

29.5 Link Operations by ELC

29.5.1 Event Signal Output to ELC

The CMT uses the event link controller (ELC) to perform link operation to a preset module using the interrupt request signal as the event signal. The CMT outputs the event signal upon a CMT1 compare match.
The event signal can be output regardless of the setting of the corresponding interrupt request enable bit (CMTn.CMCR.CMIE).

29.5.2 CMT Operation When Receiving an Event Signal from ELC

The CMT can perform either of the following operations upon the event preset by the ELSR7 register of the ELC.

(1) Count Start

When the CMT count start operation is selected by the ELOPC register of the ELC and the event specified by ELSR7 occurs, the CMSTR0.STR1 bit is set to 1, starting the CMT count operation.
However, if the specified event occurs while the CMSTR0.STR1 bit is 1, the event is ignored.

(2) Event Count

When the CMT event count operation is selected by the ELOPC register of the ELC and the event specified by ELSR7 occurs with the CMSTR0.STR1 bit being 1, the events are counted as the count source regardless of the CMT1.CMCR.CKS[1:0] bit setting. Reading the counter value returns the number of events that have been actually input.

(3) Count Restart

When the CMT count restart operation is selected by the ELOPC register of the ELC and the event specified by ELSR7 occurs, the CMT1.CMCNT counter value is modified to the initial value. If the CMSTR0.STR1 bit is 1 here, the count operation can be continued.

29.5.3 Notes on Operating CMT According to an Event Signal from ELC

The following describes the notes on operating the CMT using the event link feature.

(1) Count Start

When the event specified by ELSR7 occurs during the write cycle to the CMSTR0.STR1 bit, the cycle is not completed; setting 1 according to the event occurrence takes priority.

(2) Event Count

When the event specified by ELSR7 occurs during the write cycle to the CMT1.CMCNT, the cycle is not completed; event count operation according to the event occurrence takes priority.

(3) Count Restart

When the event specified by ELSR7 occurs during the write cycle to the CMT1.CMCNT, the cycle is not completed; count value initialization according to the event occurrence takes priority.

29.6 Usage Notes

29.6.1 Setting the Module Stop Function

The CMT can be enabled or disabled using the module stop control register. After a reset, the CMT is in the module stop state. The registers can be accessed by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

29.6.2 Conflict between CMCNT Counter Writing and Compare Match

When the compare match signal is generated while writing to the CMCNT counter, clearing the CMCNT counter has priority over writing to it. In this case, the CMCNT counter is not written to. Figure 29.5 shows the timing to clear the CMCNT counter.

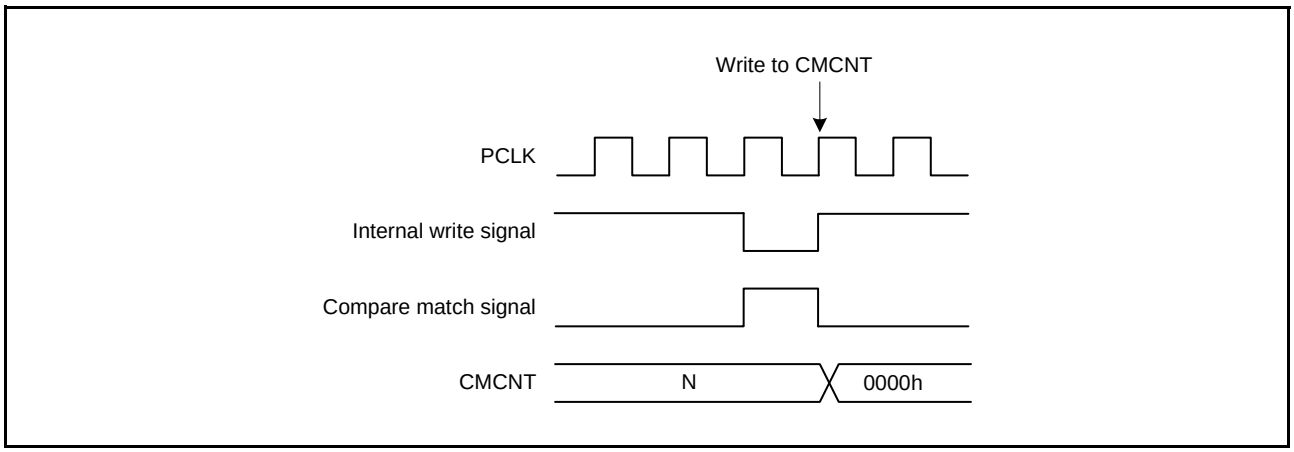


Figure 29.5 Conflict between CMCNT Counter Writing and Compare Match

29.6.3 Conflict between CMCNT Counter Writing and Incrementing

If writing to the counter and the incrementing conflict, the writing has priority over the incrementing. Figure 29.6 shows the timing to write the CMCNT counter.

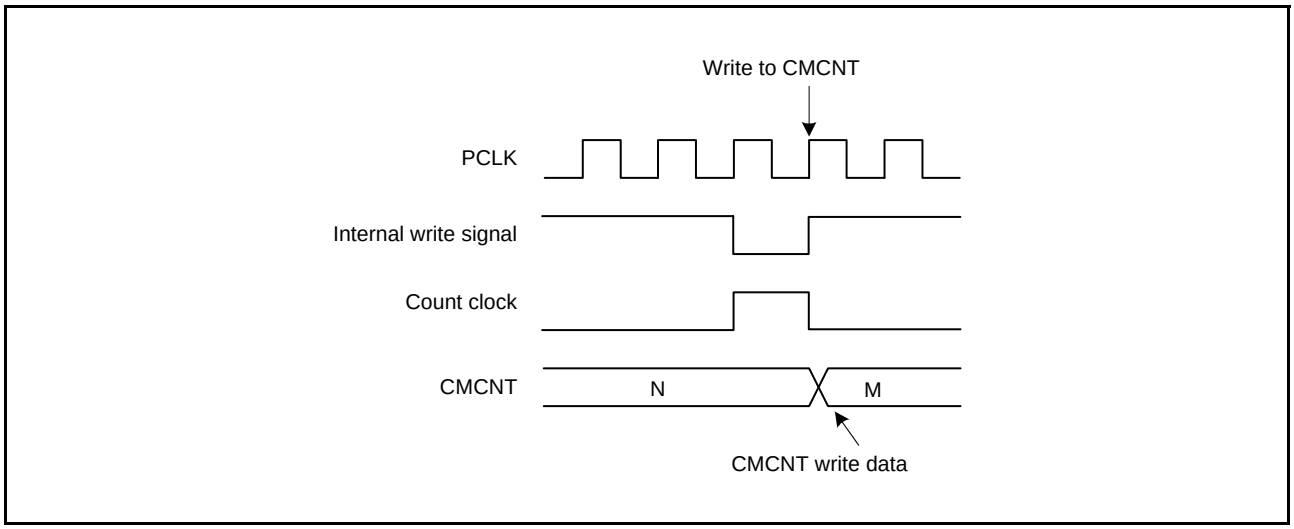


Figure 29.6 Conflict between CMCNT Counter Writing and Incrementing

30. Compare Match Timer W (CMTW)

This MCU includes two units (unit 0 and unit 1) with one channel of 32-bit compare match timer W (CMTW).
CMTW has a 32-bit counter and can generate interrupts each time a set period elapses.
In this section, “PCLK” is used to refer to PCLKB.

30.1 Overview

Table 30.1 shows the specifications of the CMTW.
Figure 30.1 shows a block diagram of the CMTW0 and Figure 30.2 shows a block diagram of the CMTW1.

Table 30.1 CMTW Specifications

Item	Function
Number of channels	Two channels (unit 0, unit 1)
Timer counter	16-bit/32-bit selectable up-counter The counter returns to 0000 0000h after a compare match.
Prescaler	Four dividing clocks are output. Selectable from any of PCLK/8, PCLK/32, PCLK/128, and PCLK/512
Input capture	Up to two input capture input signals available.
Output compare	Up to two output compare output signals available.
Compare match	One compare match available (no output compare output pin used).
Interrupts	Compare match interrupt Input capture 0 and 1 interrupts Output compare 0 and 1 interrupts
Event link function (output) (Unit 0)	Compare match
Event link function (input) (Unit 0)	One of the following operations is enabled after an event is accepted: • Count start operation • Event count operation • Count restart operation
Low power consumption function	Each unit can be placed in the module stop state.

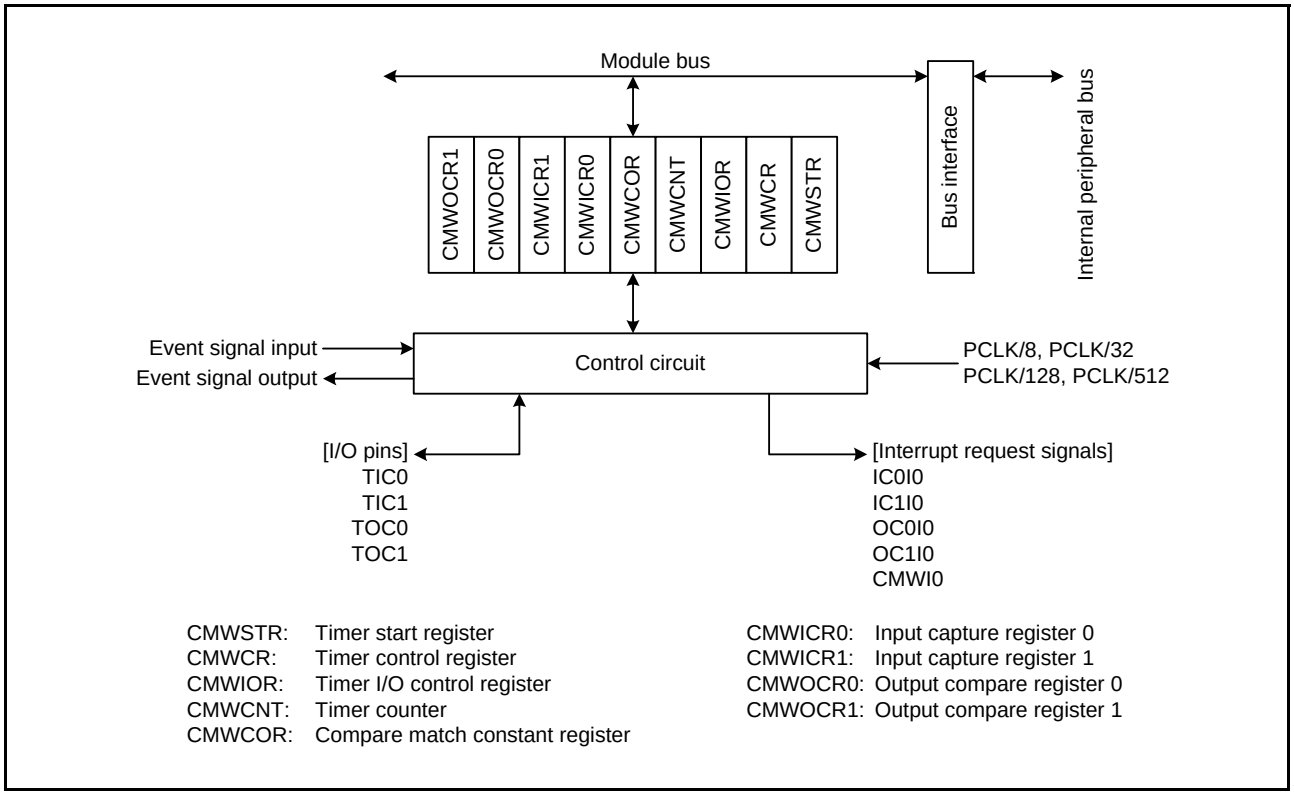


Figure 30.1 CMTW0 Block Diagram

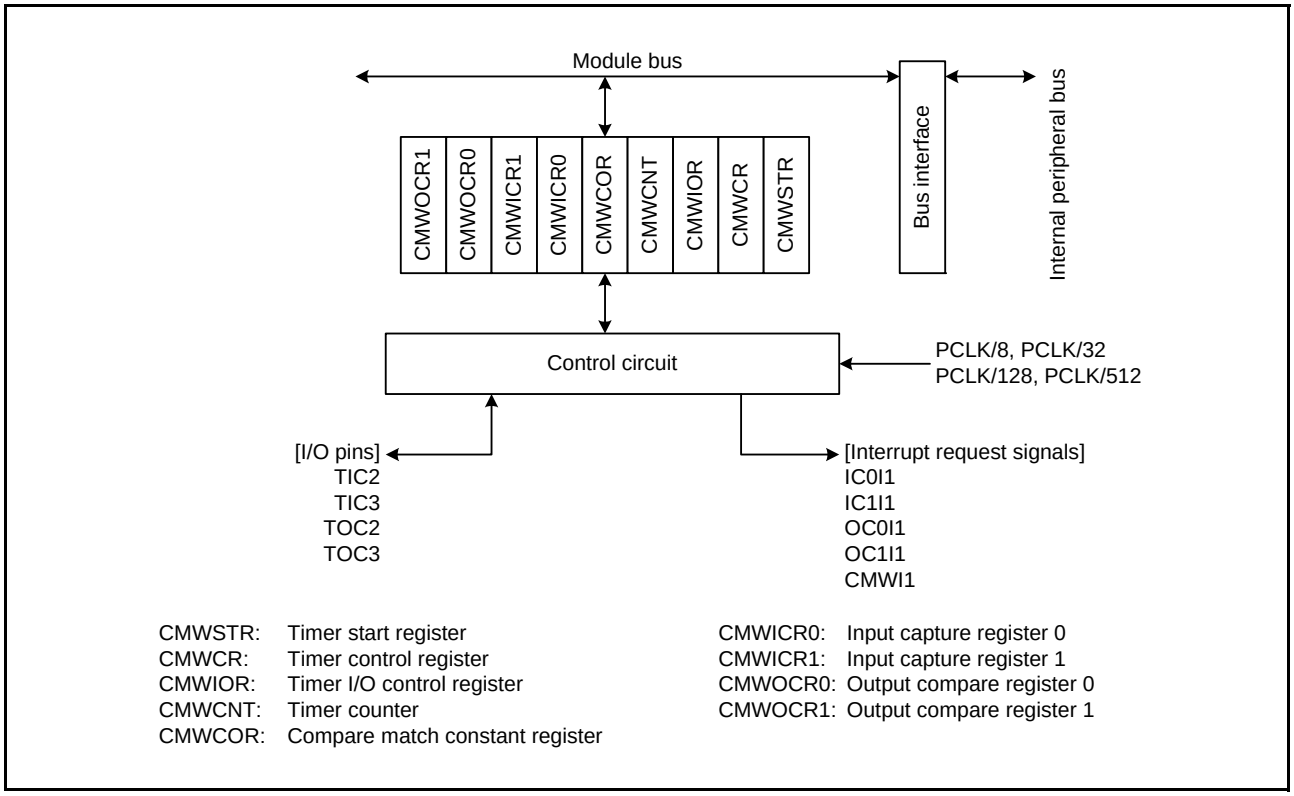


Figure 30.2 CMTW1 Block Diagram

Table 30.2 shows the CMTW pin configuration.

Table 30.2 CMTW Pin Configuration

Unit	Pin Name	I/O	Description
CMTW0	TIC0	Input	Input capture input for the CMTW0.CMWICR0 register
	TIC1	Input	Input capture input for the CMTW0.CMWICR1 register
	TOC0	Output	Output compare output for the CMTW0.CMWOCR0 register
	TOC1	Output	Output compare output for the CMTW0.CMWOCR1 register
CMTW1	TIC2	Input	Input capture input for the CMTW1.CMWICR0 register
	TIC3	Input	Input capture input for the CMTW1.CMWICR1 register
	TOC2	Output	Output compare output for the CMTW1.CMWOCR0 register
	TOC3	Output	Output compare output for the CMTW1.CMWOCR1 register

30.2 Register Descriptions

30.2.1 Timer Start Register (CMWSTR)

Address(es): CMTW0.CMWSTR 0009 4200h, CMTW1.CMWSTR 0009 4280h

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

Bit	Symbol	Bit Name	Description	R/W
b0	STR	Counter Start	0: CMWCNT counter count is stopped. (The value immediately before count operation stops is retained and the count operation is stopped.) 1: CMWCNT counter count is started.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

STR Bit (Counter Start)

Specifies whether the timer counter operates or is stopped. The relevant prescaler operates or is stopped according to the settings of the STR bit.

30.2.2 Timer Control Register (CMWCR)

Address(es): CMTW0.CMWCR 0009 4204h, CMTW1.CMWCR 0009 4284h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CCLR[2:0]			—	—	—	CMS	—	OC1IE	OC0IE	IC1IE	IC0IE	CMWIE	—	CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK/8 0 1: PCLK/32 1 0: PCLK/128 1 1: PCLK/512	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	CMWIE	Compare Match Interrupt Request Enable	0: Interrupt request (CMWI) disabled 1: Interrupt request (CMWI) enabled	R/W
b4	IC0IE	Input Capture 0 Interrupt Request Enable	0: Interrupt request (IC0I) disabled 1: Interrupt request (IC0I) enabled	R/W
b5	IC1IE	Input Capture 1 Interrupt Request Enable	0: Interrupt request (IC1I) disabled 1: Interrupt request (IC1I) enabled	R/W
b6	OC0IE	Output Compare 0 Interrupt Request Enable	0: Interrupt request (OC0I) disabled 1: Interrupt request (OC0I) enabled	R/W
b7	OC1IE	Output Compare 1 Interrupt Request Enable	0: Interrupt request (OC1I) disabled 1: Interrupt request (OC1I) enabled	R/W
b8	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b9	CMS	Timer Counter Size	0: 32 bits 1: 16 bits	R/W
b12 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b13	CCLR[2:0]	Counter Clear	b15 b13 0 0 0: CMWCNT counter cleared by CMWCOR register compare match 0 0 1: Clearing of CMWCNT counter disabled 0 1 0: Clearing of CMWCNT counter disabled 0 1 1: Clearing of CMWCNT counter disabled 1 0 0: CMWCNT counter cleared by CMWICR0 register input capture 1 0 1: CMWCNT counter cleared by CMWICR1 register input capture 1 1 0: CMWCNT counter cleared by CMWOCR0 register compare match 1 1 1: CMWCNT counter cleared by CMWOCR1 register compare match	R/W

The CMWCR register should be set while the timer counter (CMWCNT) operation is stopped.

CKS[1:0] Bits (Clock Select)

Select the clock to be input to the CMWCNT counter among four internal clocks obtained by dividing the peripheral module clock (PCLK). When the CMWSTR.STR bit is set to 1, the CMWCNT counter starts counting based on the clock selected with the CMWCR.CKS[1:0] bits.

CMWIE Bit (Compare Match Interrupt Request Enable)

Enables or disables compare match interrupt request (CMWI) generation when the CMWCNT counter and the CMWCOR register values match.

IC0IE Bit (Input Capture 0 Interrupt Request Enable)

Enables or disables input capture 0 interrupt request (IC0I) generation when input capture is generated in the CMWICR0 register.

IC1IE Bit (Input Capture 1 Interrupt Request Enable)

Enables or disables input capture 1 interrupt request (IC1I) generation when input capture is generated in the CMWICR1 register.

OC0IE Bit (Output Compare 0 Interrupt Request Enable)

Enables or disables output compare 0 interrupt request (OC0I) generation when the CMWCNT counter and the CMWOCR0 register values match.

OC1IE Bit (Output Compare 1 Interrupt Request Enable)

Enables or disables output compare 1 interrupt request (OC1I) generation when the CMWCNT counter and CMWOCR1 register values match.

CMS Bit (Timer Counter Size)

Selects either 16 or 32 bits as the size of the timer counter (CMWCNT). The size selected with the CMS bit is valid in the CMWCOR, CMWICR0, CMWICR1, CMWOCR0, and CMWOCR1 registers.

CCLR[2:0] Bits (Counter Clear)

Select the CMWCNT counter clearing source.

30.2.3 Timer I/O Control Register (CMWIOR)

Address(es): CMTW0.CMWIOR 0009 4208h, CMTW1.CMWIOR 0009 4288h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMWE	—	OC1E	OC0E	OC1[1:0]		OC0[1:0]		—	—	IC1E	IC0E	IC1[1:0]		IC0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	IC0[1:0]	Input Capture 0 Control	b1 b0 0 0: Input capture at the rising edge 0 1: Input capture at the falling edge 1 0: Input capture at both edges 1 1: Setting prohibited	R/W
b3, b2	IC1[1:0]	Input Capture 1 Control	b3 b2 0 0: Input capture at the rising edge 0 1: Input capture at the falling edge 1 0: Input capture at both edges 1 1: Setting prohibited	R/W
b4	IC0E	Input Capture 0 Enable	0: Input capture 0 operation disabled 1: Input capture 0 operation enabled	R/W
b5	IC1E	Input Capture 1 Enable	0: Input capture 1 operation disabled 1: Input capture 1 operation enabled	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	OC0[1:0]	Output Compare 0 Control	b9 b8 0 0: Retains the output value.*1 0 1: Initially outputs low and toggles the output value upon compare match. 1 0: Initially outputs high and toggles the output value upon compare match. 1 1: Setting prohibited	R/W
b11, b10	OC1[1:0]	Output Compare 1 Control	b11 b10 0 0: Retains the output value.*1 0 1: Initially outputs low and toggles the output value upon compare match. 1 0: Initially outputs high and toggles the output value upon compare match. 1 1: Setting prohibited	R/W
b12	OC0E	Output Compare 0 Enable	0: Output compare 0 operation disabled 1: Output compare 0 operation enabled	R/W
b13	OC1E	Output Compare 1 Enable	0: Output compare 1 operation disabled 1: Output compare 1 operation enabled	R/W
b14	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b15	CMWE	Compare Match Enable	0: Compare match operation disabled 1: Compare match operation enabled	R/W

Note 1. After reset, low is output until the CMWIOR register is set.

30.2.4 Timer Counter (CMWCNT)

Address(es): CMTW0.CMWCNT 0009 4210h, CMTW1.CMWCNT 0009 4290h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

The CMWCNT counter is a readable/writable up-counter.
Before starting count operation, the timer control register (CMWCR) should be set. When the 16-bit counter size is selected with the CMS bit in the timer control register (CMWCR), bits 15 to 0 in this register are valid. When writing to the CMWCNT counter, write data in 32-bit units with the upper bits set to 0000h. The CMWCNT counter can only be accessed in longword units.
When the CMWSTR.STR bit is set to 1, the CMWCNT counter starts counting. When the CMWSTR.STR bit is set to 0, the CMWCNT counter retains the value immediately before a stop of counting and stops counting.

30.2.5 Compare Match Constant Register (CMWCOR)

Address(es): CMTW0.CMWCOR 0009 4214h, CMTW1.CMWCOR 0009 4294h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

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

The CMWCOR register is a readable/writable register that specifies the time up to a compare match between the timer counter (CMWCNT) value and CMWCOR value. When the 16-bit counter size is selected with the CMS bit in the timer control register (CMWCR), bits 15 to 0 in this register are valid. When writing to the CMWCOR register, write data in 32-bit units with the upper bits set to 0000h. The CMWCOR register can only be accessed in longword units. To detect an overflow, set the CMWCOR register value to FFFF FFFFh (32-bit count operation) or 0000 FFFFh (16-bit count operation). When the CMWCNT counter is set to 0, a compare match interrupt request (CMWI) can be used as an overflow detection signal.

30.2.6 Input Capture Register n (CMWICRn) (n = 0, 1)

Address(es): CMTW0.CMWICR0 0009 4218h, CMTW0.CMWICR1 0009 421Ch, CMTW1.CMWICR0 0009 4298h, CMTW1.CMWICR1 0009 429Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

The CMWICRn register is a read-only register in which the CMWCNT value is stored when an input capture is generated.
When the 16-bit counter size is selected with the CMWCR.CMS bit, bits 15 to 0 in these registers are valid. Writing to these registers is invalid. The CMWICRn register can only be accessed in longword units.

30.2.7 Output Compare Register n (CMWOCRn) (n = 0, 1)

Address(es): CMTW0.CMWOCR0 0009 4220h, CMTW0.CMWOCR1 0009 4224h, CMTW1.CMWOCR0 0009 42A0h, CMTW1.CMWOCR1 0009 42A4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

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

The CMWOCRn register is a readable/writable register that set the value to be compared when an output compare is generated.
When the 16-bit counter size is selected with the CMWCR.CMS bit, bits 15 to 0 of these registers become valid. When writing to these registers, write data in 32-bit units with the upper bits set to 0000h.
The CMWOCRn register can only be accessed in longword units. The initial value of CMWOCR0 and CMWOCR1 registers is FFFF FFFFh.

30.3 Operation

When the CMWCR register is set and then the STR bit in CMWSTR is set to 1, the CMTW starts count operation. Setting the CMWSTR.STR bit to 0 enables the CMWCNT counter to retain the value immediately before a stop of counting and stop counting. Setting the CMWIOR register enables using the compare match function, input capture input function, and output compare output function.

30.3.1 Period Count Operation

When the counter clock is selected by using the CMWCR.CKS[1:0] bits and the CMWSTR.STR bit is set to 1, the CMWCNT counter starts counting cycles of the selected clock. When clearing of the counter is selected by the CMWCR.CCLR[2:0] bits and the counter clearing source is generated, the CMWCNT counter becomes 0000 0000h and continues counting. When clearing of the counter is not selected, an overflow is generated when FFFF FFFFh changes to 0000 0000h during 32-bit count operation and 0000 FFFFh changes to 0000 0000h during 16-bit count operation, and the CMWCNT counter continues counting.

30.3.2 Compare Match Function

When the values of the CMWCNT counter and CMWCOR register match, the CMWCNT counter becomes 0000 0000h. At this time, a compare match interrupt request (CMWI) is generated. The CMWCNT counter restarts counting from 0000 0000h.

To enable overflow detection, the CMWCOR register value should be set to FFFF FFFFh (when the counter size is 32 bits) or 0000 FFFFh (when the counter size is 16 bits). When the values of the CMWCNT counter and CMWCOR register match, the CMWCNT counter becomes 0000 0000h. In this case, the compare match interrupt request (CMWI) is generated. The CMWCNT counter then restarts counting from 0000 0000h.

Figure 30.3 shows an example of procedure for setting compare match operation.

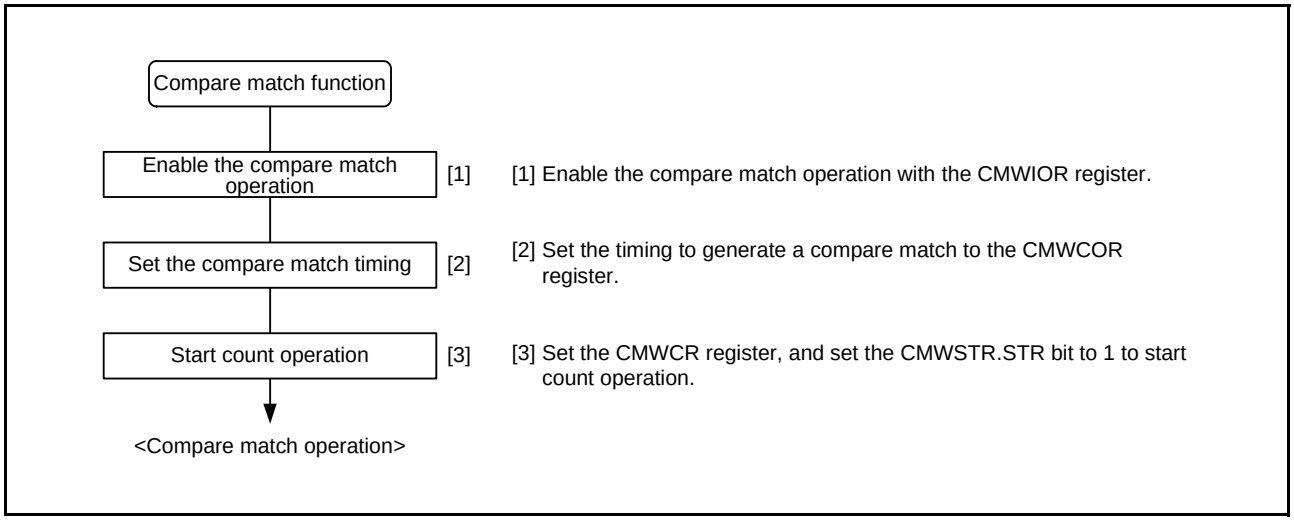


Figure 30.3 Procedure for Setting Compare Match Operation

Figure 30.4 shows an example when compare match with CMWCOR is set as a counter clearing source.

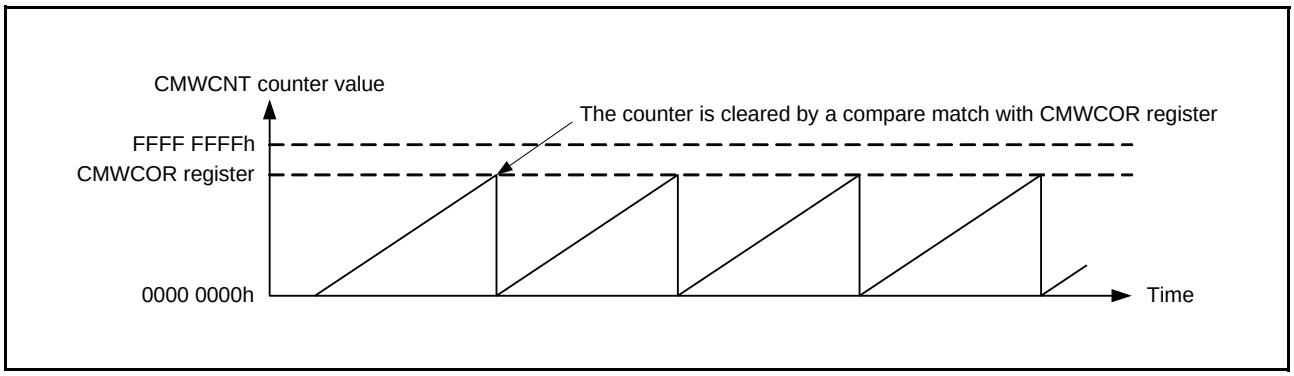


Figure 30.4 Example of Compare Match Operation

Figure 30.5 shows an example when CMWCOR is set to FFFF FFFFh and an overflow is detected.

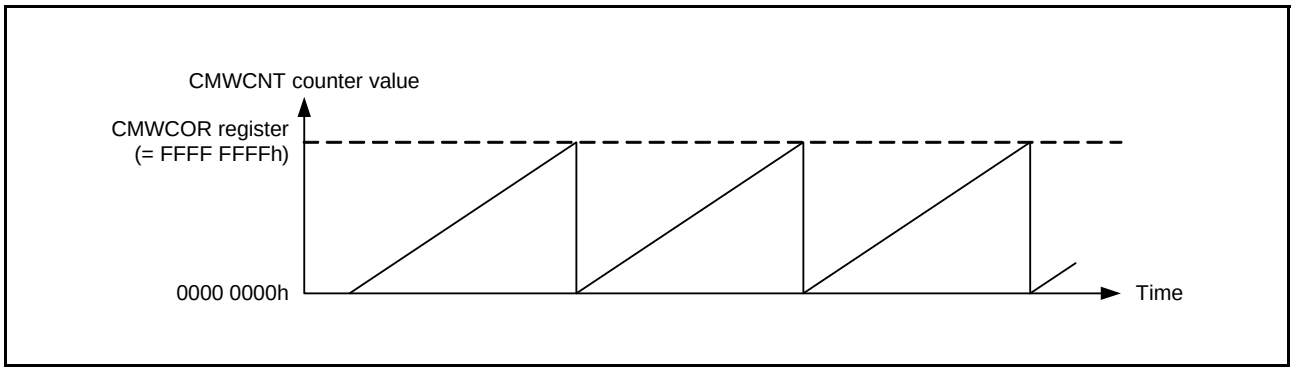


Figure 30.5 Example of Compare Match Operation (Overflow Detected)

30.3.3 Output Compare Function

The output compare function can be used for toggle waveform output. When the CMWCNT counter value matches either of the values of the CMWOCR0 or CMWOCR1 register, the output compare interrupt request (OC0I or OC1I) is generated. Figure 30.6 shows an example of procedure for setting output compare operation.

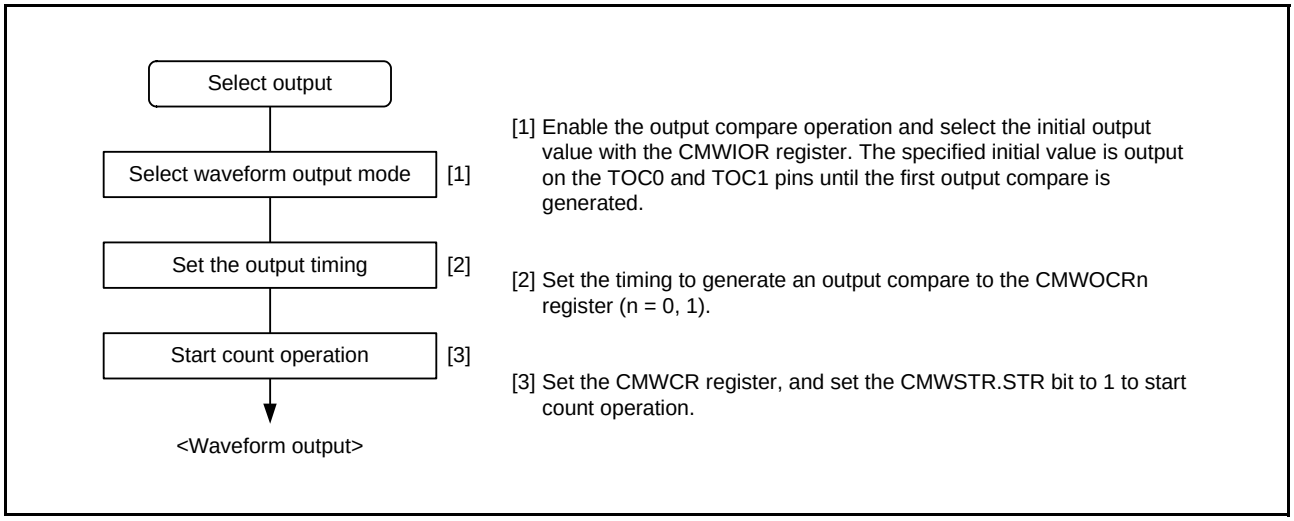


Figure 30.6 Procedure for Setting Output Compare Operation

Figure 30.7 shows an example of toggle waveform output from the TOC0 and TOC1 pins when the counter is set to be cleared by compare match with the CMWOCR1 register.

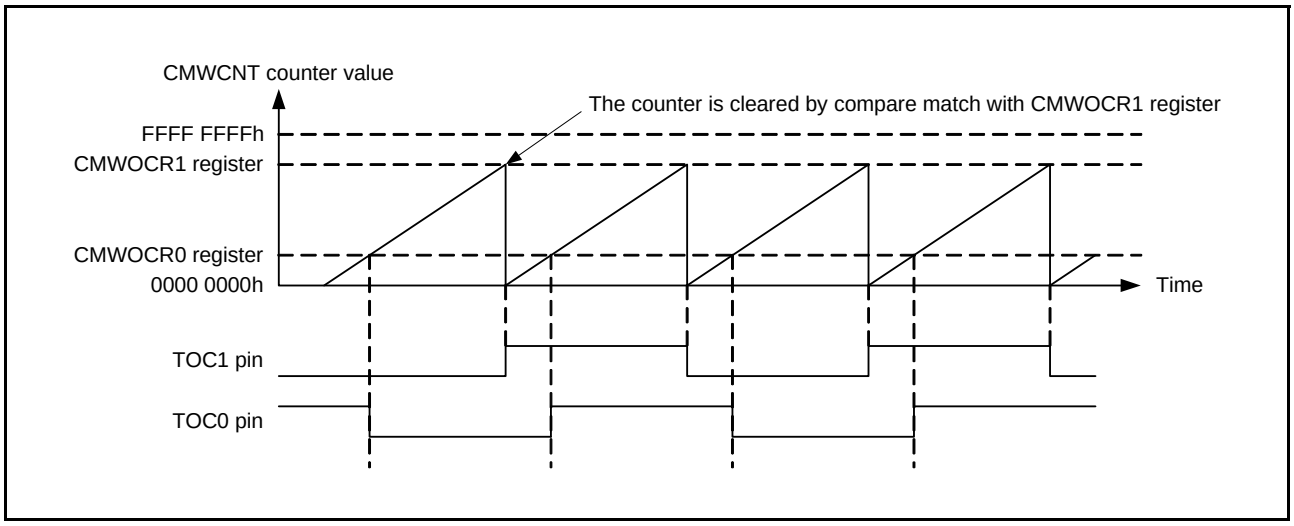


Figure 30.7 Example of Output Compare Operation (Unit 0)

30.3.4 Input Capture Function

Through detecting the edge on the TIC0 and TIC1 pin input, the CMWCNT counter value can be transferred to the CMWICR0 and CMWICR1 registers, respectively. The edges to be detected can be selected from among the rising edge alone, falling edge alone, and both the rising and falling edges. When the CMWCNT counter value is transferred to the CMWICR0 or CMWICR1 register using the input capture operation, an input capture interrupt request (IC0I or IC1I) is generated. Figure 30.8 shows an example of procedure for setting input capture operation.

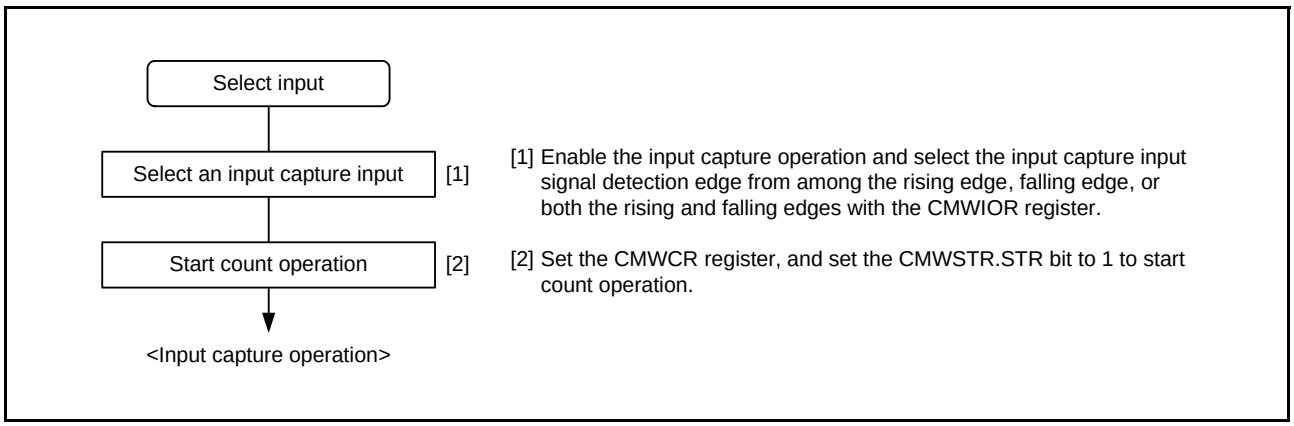


Figure 30.8 Procedure for Setting Input Capture Operation

Figure 30.9 shows an example in which both edges are selected for the TIC0 pin input capture detection edge and the falling edge for the TIC1 pin, and the CMWCNT counter is cleared by a CMWICR1 register input capture.

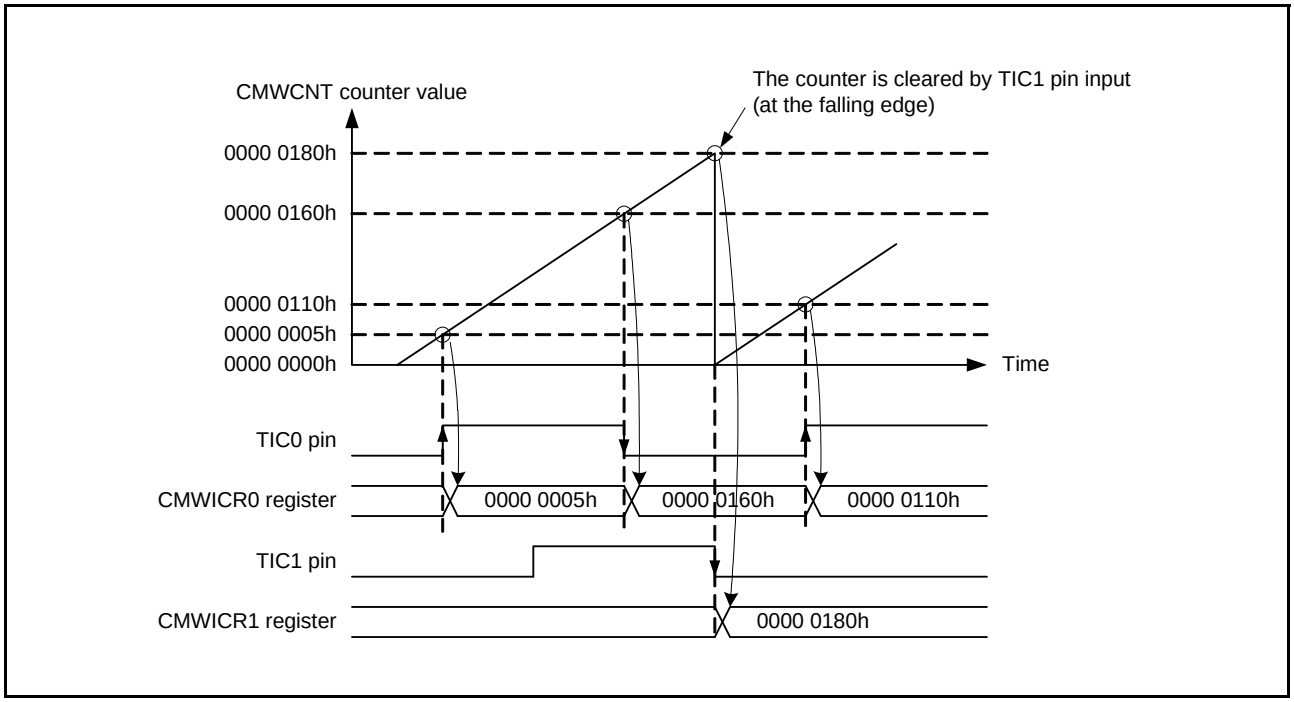


Figure 30.9 Example of Input Capture Operation (Unit 0)

30.3.5 Counter Size

With the CMTW, either 16 or 32 bits can be selected as the counter size by using the CMWCR.CMS bit.
When the counter is used as a 16-bit counter, set the value of the CMWCOR register in 32-bit units with the upper 16 bits set to 0000h. 0000 FFFFh should be set to detect an overflow. Similarly, set the values of the CMWOCR0 and CMWOCR1 registers in 32-bit units with the upper 16 bits set to 0000h. Read the CMWOCR0 and CMWOCR1 registers in 32-bit units. The upper 16 bits can be read as 0000h.

30.3.6 Count Timing of CMWCNT Counter

By setting the CMWCR.CKS[1:0] bits, one of four clocks (PCLK/8, PCLK/32, PCLK/128, and PCLK/512) obtained by dividing the peripheral module clock (PCLK) can be selected as the counter clock to be input to the CMWCNT counter.
Figure 30.10 shows the timing.

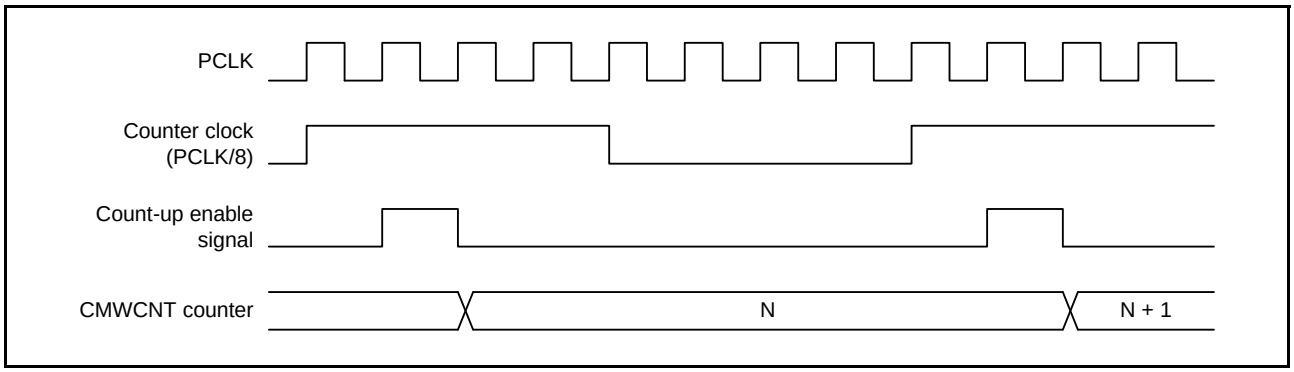


Figure 30.10 Count Timing (PCLK/8)

30.3.7 Output Compare Output Timing

A compare match signal is generated in the last state in which the CMWOCRn register (n = 0, 1) and CMWCNT counter values match (the CMWCNT counter value is updated immediately after the state). The compare match signal is generated if the CMWCNT count-up enable signal is input after a match between the CMWOCRn register and CMWCNT counter values. When a compare match signal is generated, output of the output compare pin (TOC pin) is toggled.
Figure 30.11 shows output compare output timing.

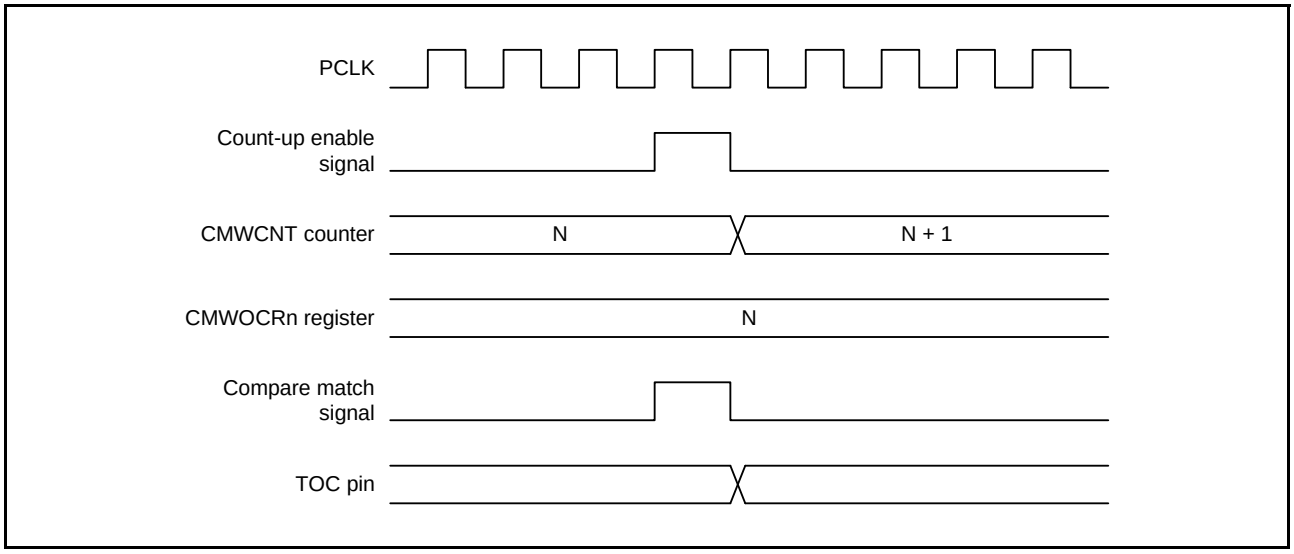


Figure 30.11 Output Compare Output Timing

30.3.8 Input Capture Timing

Figure 30.12 shows the timing of input capture operation at both edges.
When the edge of the TIC0 and TIC1 pins is detected, the CMWCNT counter value is transferred to the CMWICR0 and CMWICR1 registers, respectively.

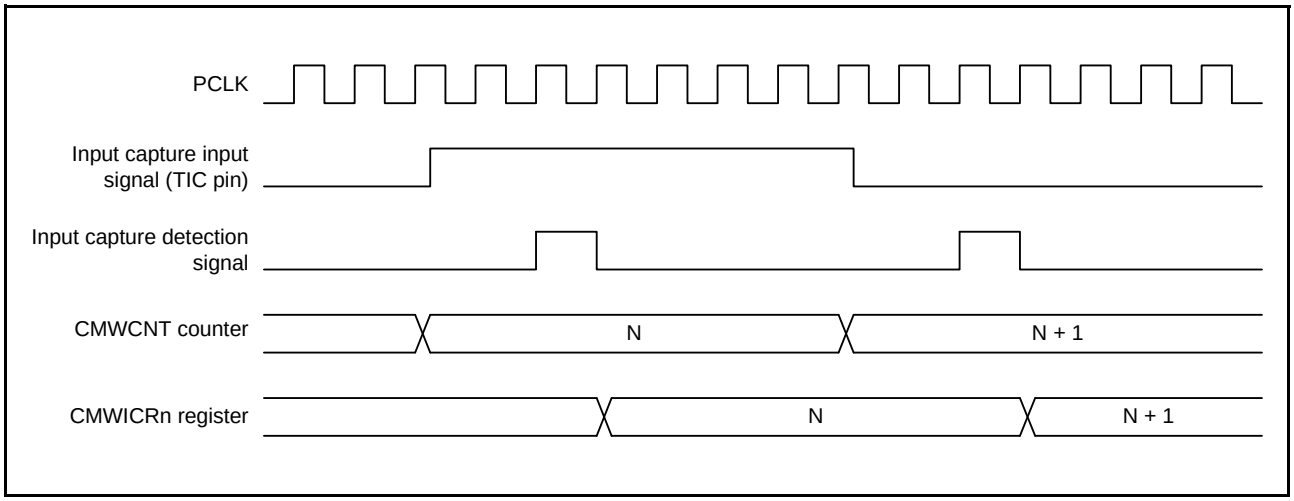


Figure 30.12 Input Capture Timing (Unit 0, Both-Edge Detection) (n = 0, 1)

30.4 Interrupts

30.4.1 CMTW Interrupt Sources and DTC/DMAC Transfer Requests

The CMTW has five interrupt sources: two input capture interrupt requests (IC0I and IC1I), two output compare interrupt requests (OC0I and OC1I), and a compare match interrupt request (CMWI).
Table 30.3 shows the interrupt sources. The interrupt sources can be enabled or disabled using the IC0IE, IC1IE, OC0IE, OC1IE, and CMWIE bits in CMWCR and are separately generated to the interrupt controller.
Each interrupt request can activate the DMAC or DTC. When the DMAC is used for data transfer, an interrupt request is not generated to the CPU. For generating an interrupt request to the CPU during data transfer using the DTC, refer to section 20, Data Transfer Controller (DTCb).

Table 30.3 CMTW Interrupt Sources

Unit	Name	Interrupt Request	Interrupt Request Enable Bit	DMAC/DTC Activation
CMTW0	CMWI0	Compare match of CMTW0.CMWCOR register	CMTW0.CMWCR.CMWIE	Possible
	IC0I0	Input capture of CMTW0.CMWICR0 register	CMTW0.CMWCR.IC0IE	Possible
	IC1I0	Input capture of CMTW0.CMWICR1 register	CMTW0.CMWCR.IC1IE	Possible
	OC0I0	Output compare of CMTW0.CMWOCR0 register	CMTW0.CMWCR.OC0IE	Possible
	OC1I0	Output compare of CMTW0.CMWOCR1 register	CMTW0.CMWCR.OC1IE	Possible
CMTW1	CMWI1	Compare match of CMTW1.CMWCOR register	CMTW1.CMWCR.CMWIE	Possible
	IC0I1	Input capture of CMTW1.CMWICR0 register	CMTW1.CMWCR.IC0IE	Possible
	IC1I1	Input capture of CMTW1.CMWICR1 register	CMTW1.CMWCR.IC1IE	Possible
	OC0I1	Output compare of CMTW1.CMWOCR0 register	CMTW1.CMWCR.OC0IE	Possible
	OC1I1	Output compare of CMTW1.CMWOCR1 register	CMTW1.CMWCR.OC1IE	Possible

30.4.2 Timing of Compare Match Interrupt Generation

When the values of the CMWCNT counter and CMWCOR register match, a compare match interrupt request (CMWI) is generated. The compare match signal is generated at the end of the cycle where the values matched (i.e. when the CMWCNT counter is updated from the matching counter value).The compare match signal, therefore, is not generated until a count-up enable signal is generated after the values of the CMWCNT counter and CMWCOR register have matched. Figure 30.13 shows the timing of compare match interrupt generation.

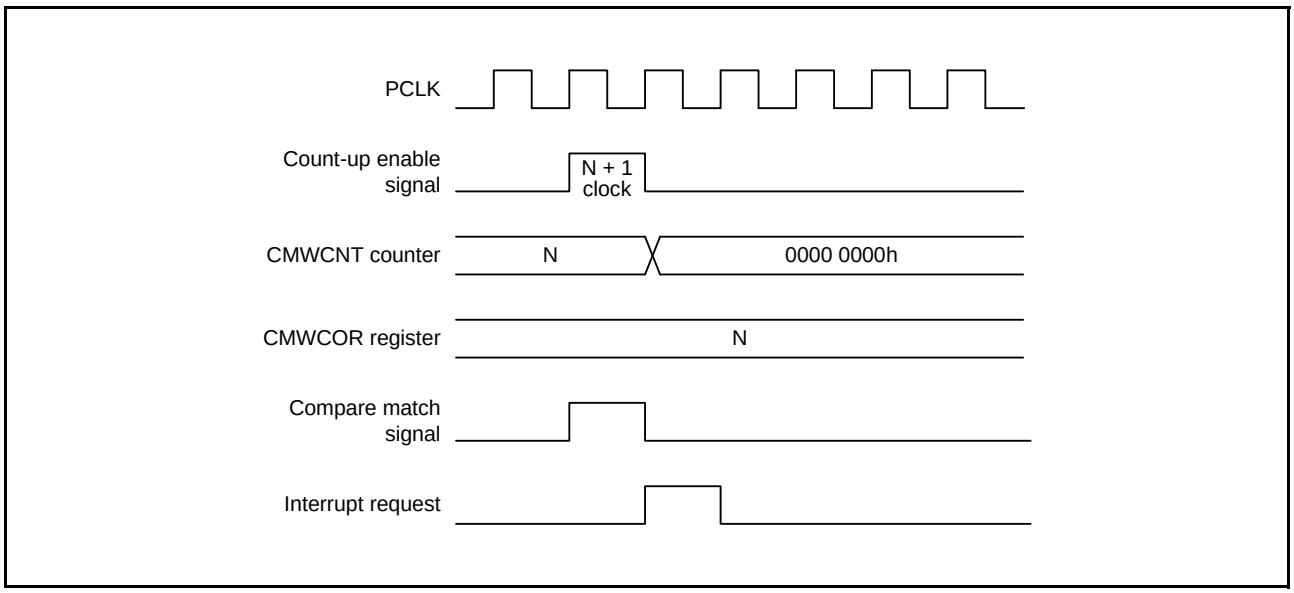


Figure 30.13 Timing of Compare Match Interrupt Generation

(a) Timing of Output Compare Interrupt Generation

Figure 30.14 shows the timing of output compare interrupt generation.

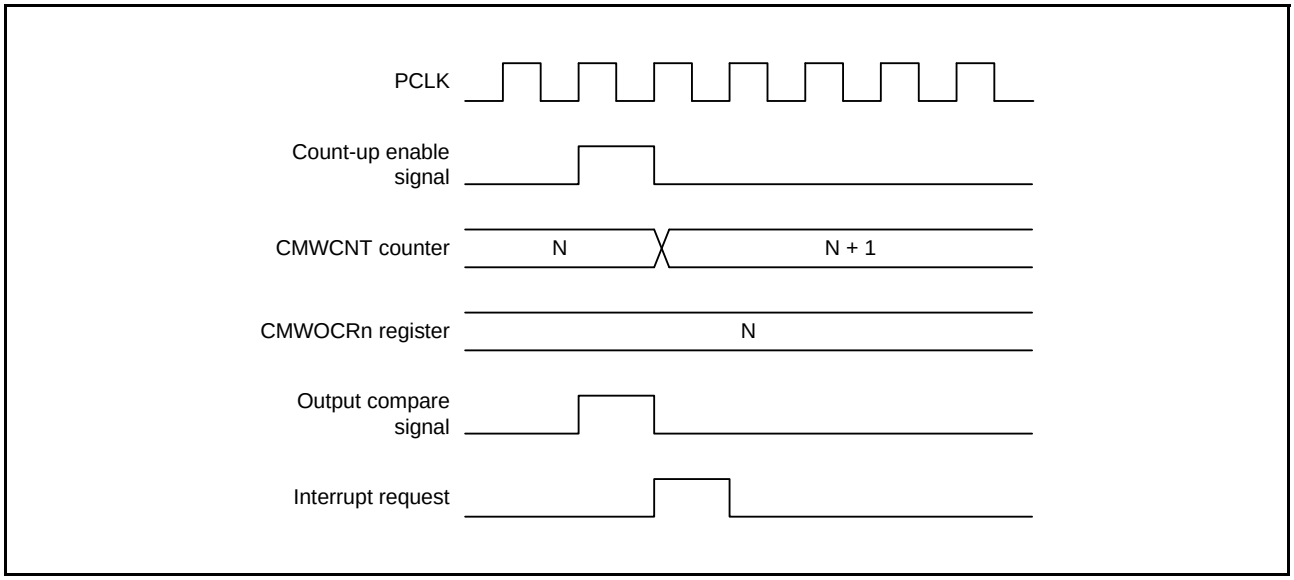


Figure 30.14 Timing of Output Compare Interrupt Generation (n = 0, 1)

(b) Timing of Input Capture Interrupt Generation

Figure 30.15 shows the timing of input capture interrupt generation.

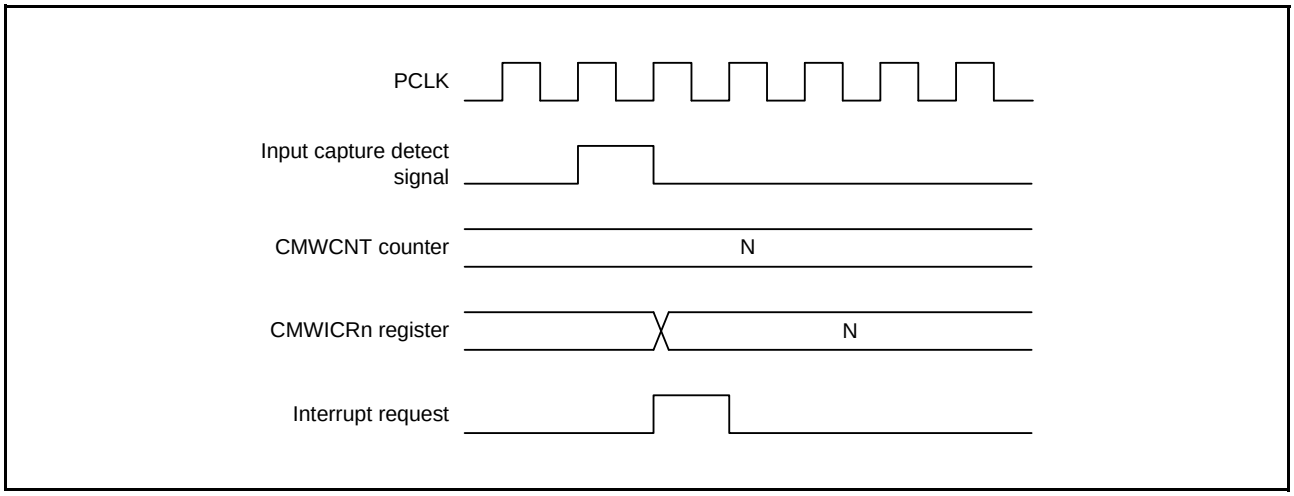


Figure 30.15 Timing of Input Capture Interrupt Generation (n = 0, 1)

30.5 Link Operations by ELC

30.5.1 Event Signal Output to ELC

The CMTW uses the event link controller (ELC) to perform link operation to a preset module using the interrupt request signal as the event signal.
The CMTW outputs the event signal upon a compare match. The corresponding channel is channel 0. The event signal can be output regardless of the corresponding interrupt request enable bit (CMWCR.CMWIE).
For details, refer to section 21, Event Link Controller (ELC).

(1) Compare Match Event

In response to a compare match, the CMTW simultaneously issues an interrupt request and a compare match event signal to the ELC. The event signal is issued regardless of the settings of the corresponding interrupt request enable bit (CMWCR.CMWIE bit).

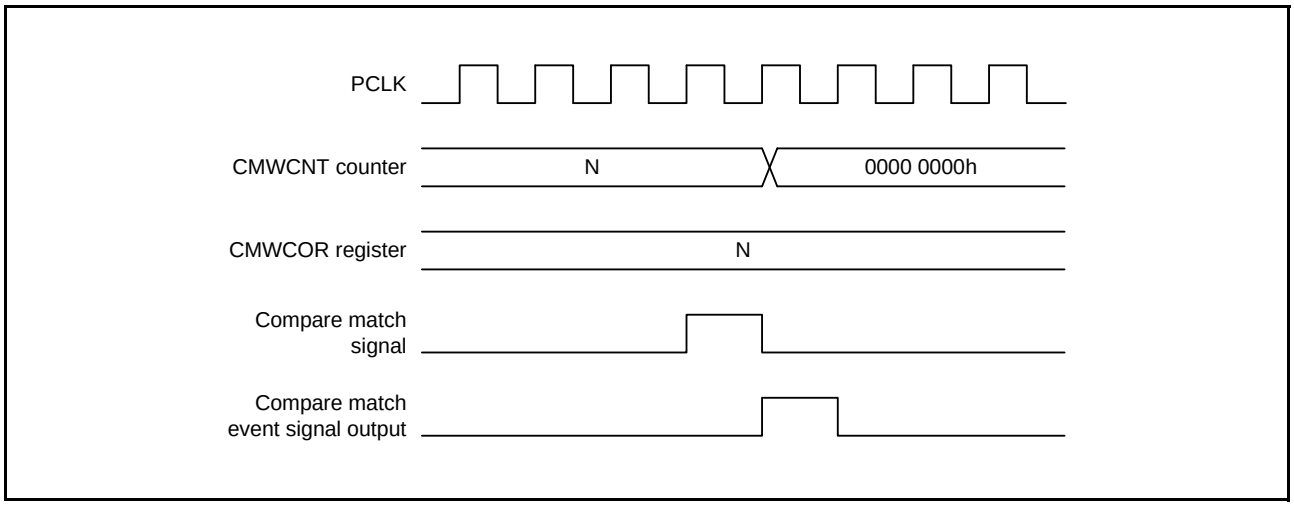


Figure 30.16 Timing of Issuing a Compare Match Event Signal

30.5.2 CMTW Operation When Receiving an Event Signal from ELC

The CMTW can perform any of the following operations according to the event preset by the ELSRn register of the ELC.

(1) Count Start

The CMTW count start operation is selected by the ELOPH register of the ELC. If the event specified by the ELSRn register occurs, the CMWSTR.STR bit is set to 1, starting the CMTW count operation.
However, if the specified event occurs while the CMWSTR.STR bit is 1, the event is ignored.

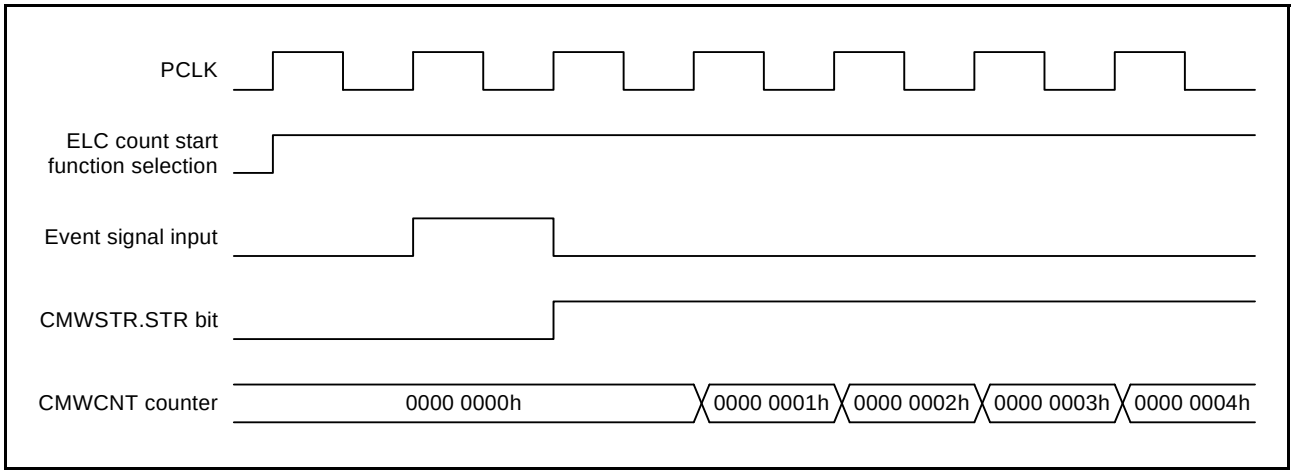


Figure 30.17 Count Start Operation on Acceptance of the Event Signal

(2) Event Count

The CMTW event count operation is selected by the ELOPH register of the ELC. If the event specified by the ELSRn register occurs when the CMWCR.STR bit is 1, the event is counted as the count source regardless of the CMWCR.CKS[1:0] bit setting.
Reading the counter value returns the number of events that have been actually input.

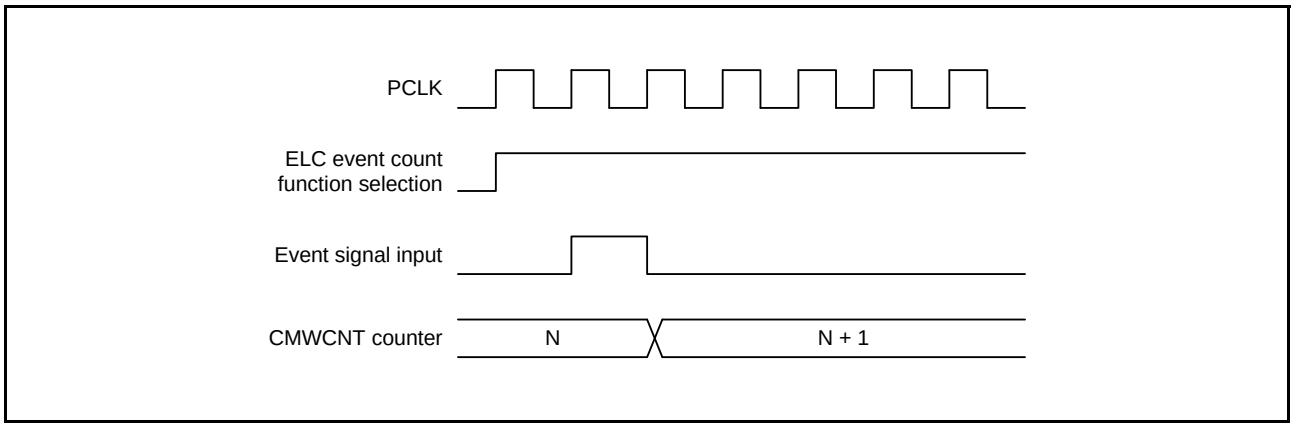


Figure 30.18 Event Count Operation on Acceptance of the Event Signal

(3) Count Restart

The CMTW count restart operation is selected by the ELOPH register of the ELC. If the event specified by the ELSRn register occurs, the CMWCNT counter value becomes 0000 0000h. If the CMWSTR.STR bit is 1, the count operation can be continued.

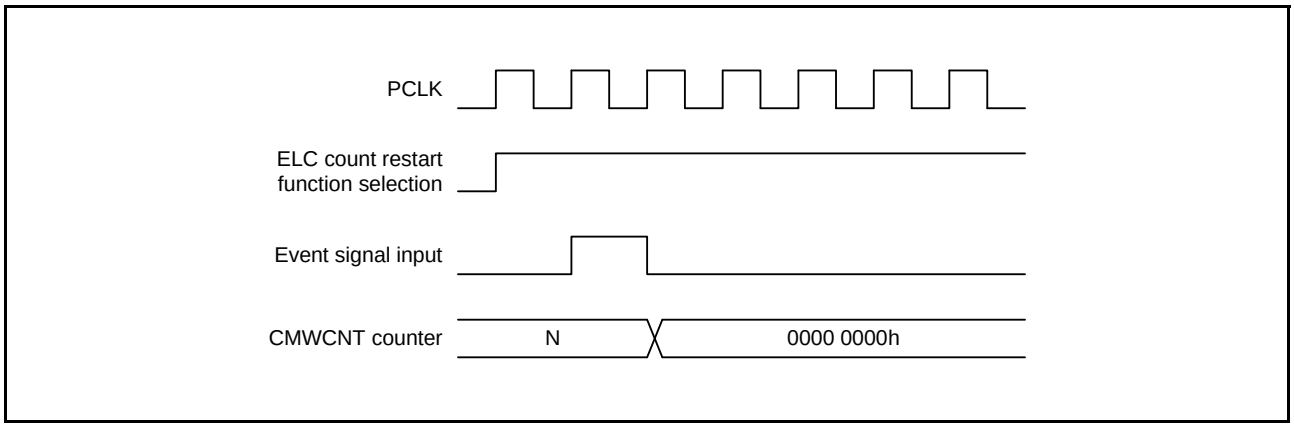


Figure 30.19 Count Restart Operation on Acceptance of the Event Signal

30.5.3 Conflict between Event Link Operation and Register Access

The followings are the notes on using CMTW for event link operations.
Table 30.4 lists count operations when event link operation and register access conflict.

Table 30.4 Count Operations When Event Link Operation and Register Access Conflict (n = 0, 1)

Event Link Operation	Register Access	CMWCNT Counter Status	Operation to be Performed
Count start	Writing to the CMWSTR.STR bit	Stopped state	Count start
		Compare match	Count start
		Counting up	Count start
Event count	Writing to CMWCNT counter	—	Event count
	Writing to CMWCOR register	Compare match	Compare match
Count restart	Writing to CMWCNT counter	Other than compare match	Count restart
	Writing to CMWCNT counter	Compare match	Compare match
	(No access to registers)	Compare match	Compare match
(No events)	Writing to CMWCNT counter	Compare match	Output of compare match interrupt request Writing to CMWCNT counter
		Counting up	Writing to CMWCNT counter
	Writing to CMWCOR register	Compare match	Compare match
	Writing to CMWOCRn register	Compare match	Compare match
	Reading from CMWCNT counter	Counting up	Counting up and reading of the previous value

(1) Count Start

When writing to the STR bit in the CMWSTR register and acceptance of the event signal are in contention, writing to the STR bit does not proceed since setting of the STR bit to 1 in response to the event takes priority.

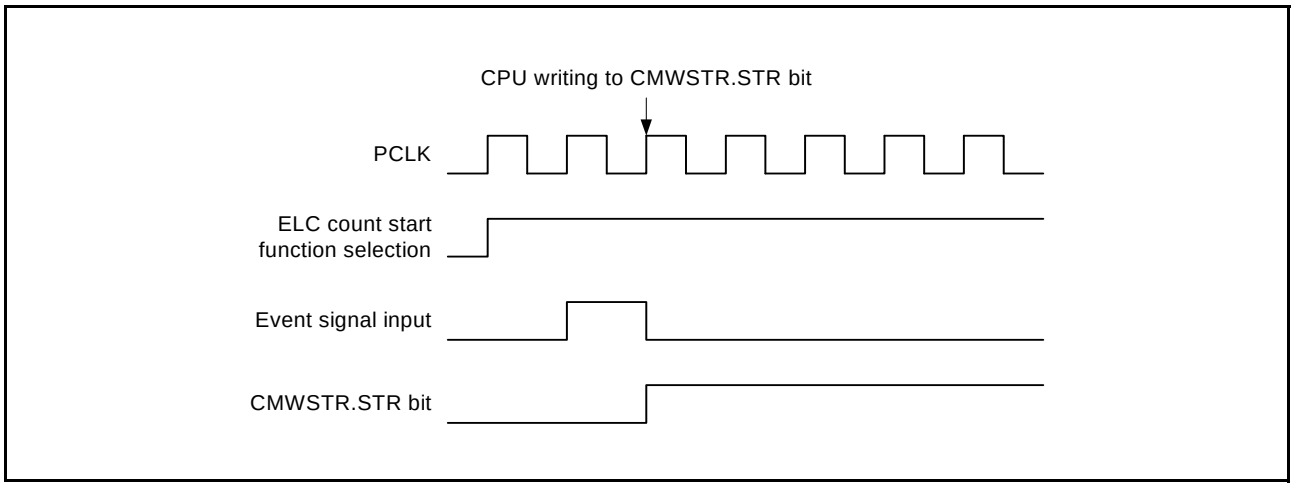


Figure 30.20 Conflict between Event Acceptance and Register Access in Count Start Operation

(2) Event Count

When writing to the CMWCNT counter and acceptance of the event signal are in contention, writing to the CMWCNT counter does not proceed since the count operation in response to the event takes priority.

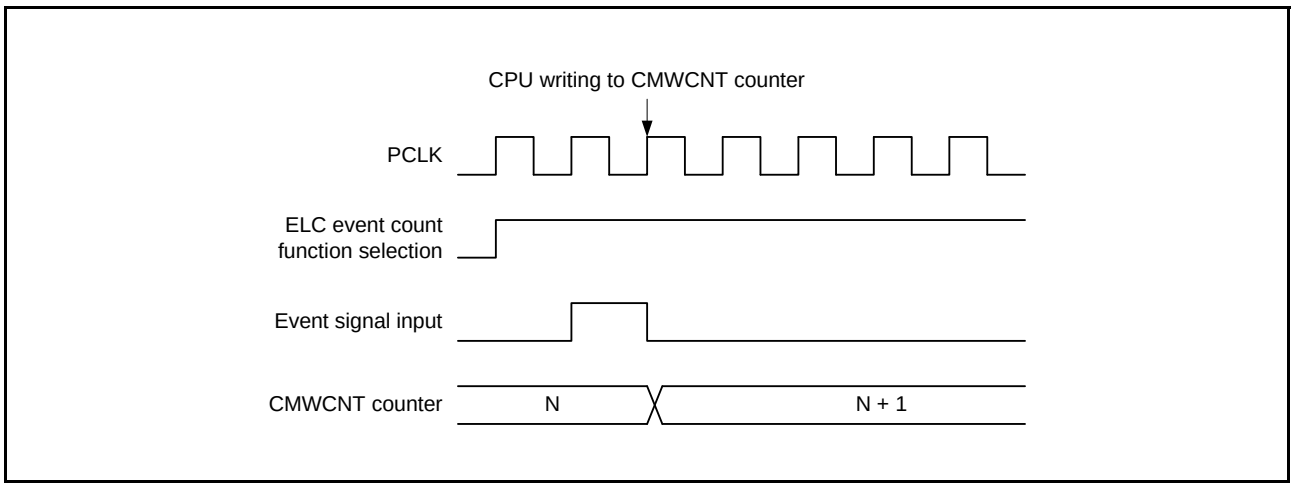


Figure 30.21 Conflict between Event Acceptance and Register Access in Event Counting Operation

(3) Count Restart

When writing to the CMWCNT counter and acceptance of the event signal are in contention, writing to the CMWCNT counter does not proceed since the counter value initialization in response to the event occurrence takes priority.

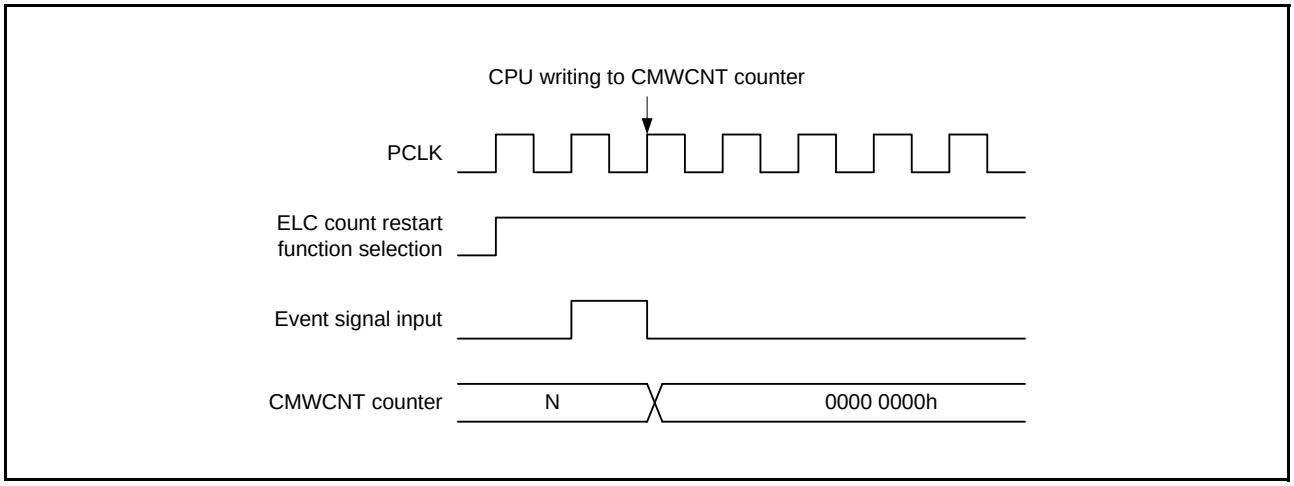


Figure 30.22 Conflict between Event Acceptance and Register Access in Count Restart Operation

30.6 Usage Notes

30.6.1 Setting the Module Stop Function

The CMTW operation can be enabled or disabled using the MSTPCRA register. The CMTW0 and CMTW1 are initially disabled after a reset. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

30.6.2 Conflict between CMWCNT Counter Writing and Compare Match

If the compare match signal is generated during writing to the CMWCNT counter, the compare match request is output but the counter is not cleared since writing to the counter takes priority.
Figure 30.23 shows the timing of conflict between CMWCNT counter writing and compare match.

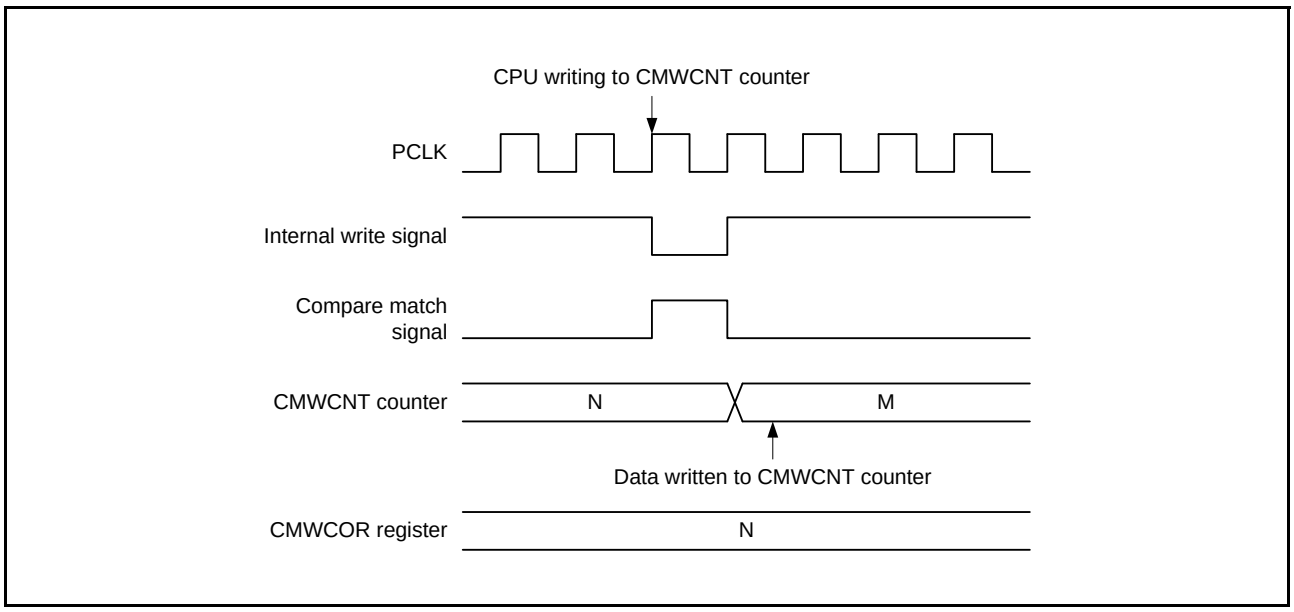


Figure 30.23 Conflict between CMWCNT Counter Writing and Compare Match

30.6.3 Conflict between CMWCNT Counter Writing and Incrementing or Clearing

In case of conflict between incrementation or clearing of the CMWCNT counter and writing to the CMWCNT counter, the counter is not actually incremented or cleared since writing to the CMWCNT counter takes priority.
Figure 30.24 shows the timing in the case of contention between writing to the CMWCNT counter and incrementation or clearing.

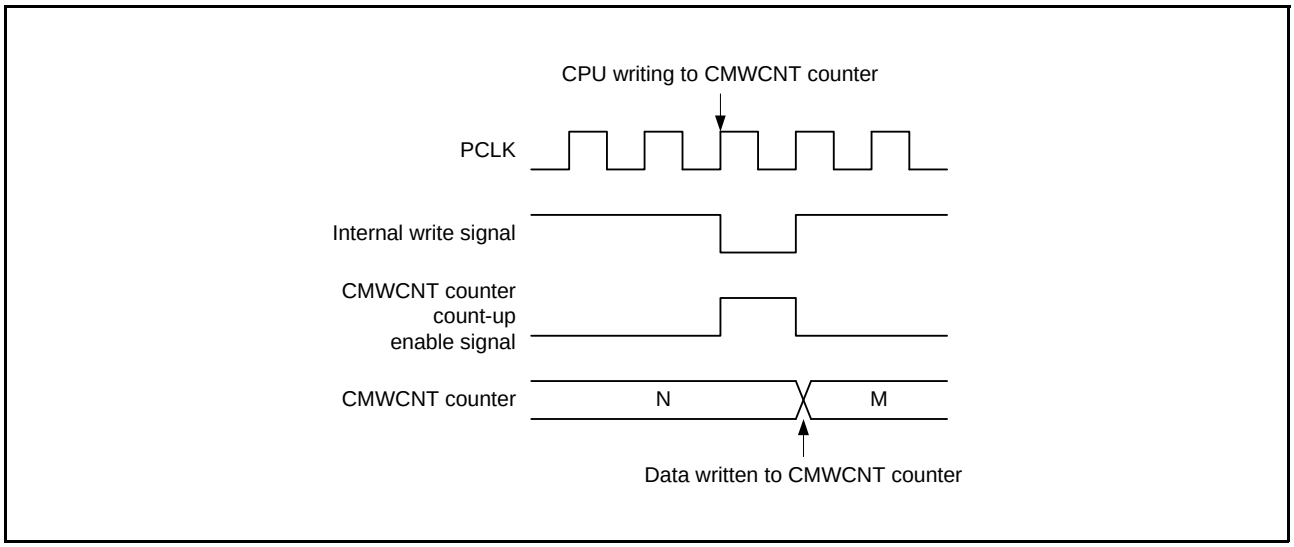


Figure 30.24 Conflict between CMWCNT Counter Writing and Incrementing

30.6.4 Conflict between CMWCOR Register Writing and Compare Match

If the compare match is generated during the CMWCOR register write cycle, the writing to the CMWCOR register takes priority and also the compare match signal is output.
Figure 30.25 shows the timing of conflict between CMWCOR register writing and compare match.

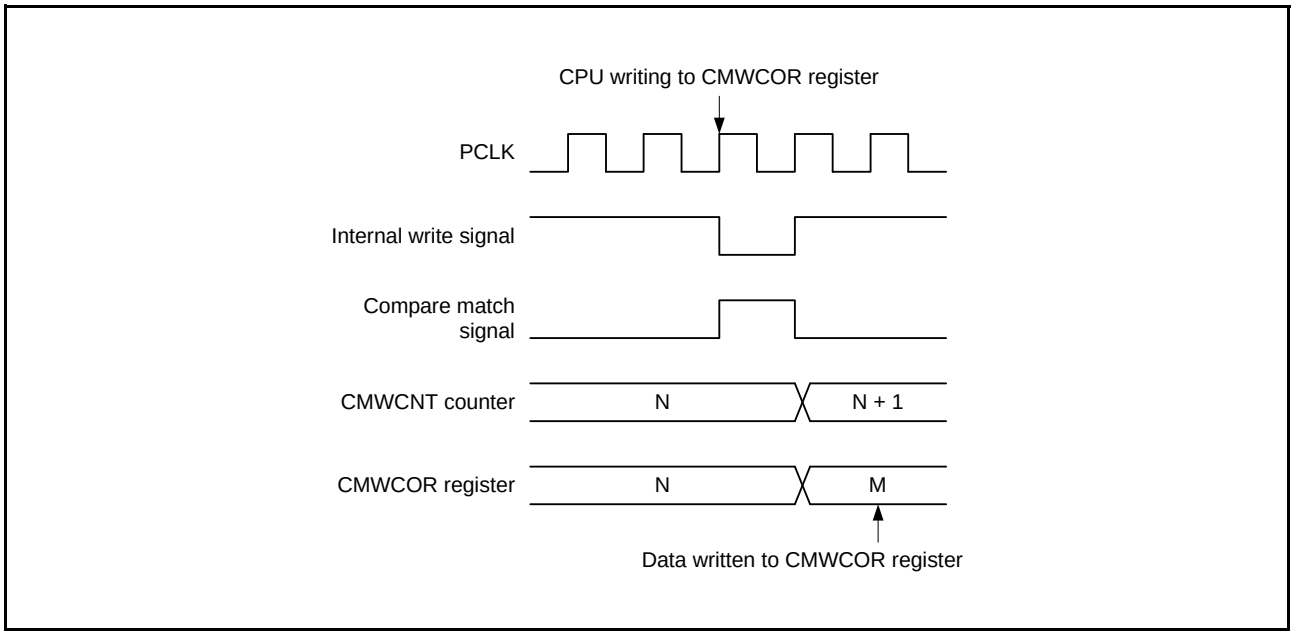


Figure 30.25 Conflict between CMWCOR Register Writing and Compare Match

30.6.5 Conflict between CMWOCRn Register Writing and Compare Match (n = 0, 1)

If the compare match is generated during the CMWOCRn register write cycle, the writing to the CMWOCRn register takes priority and also the compare match signal is output.
Figure 30.26 shows the timing of conflict between CMWOCRn register writing and compare match.

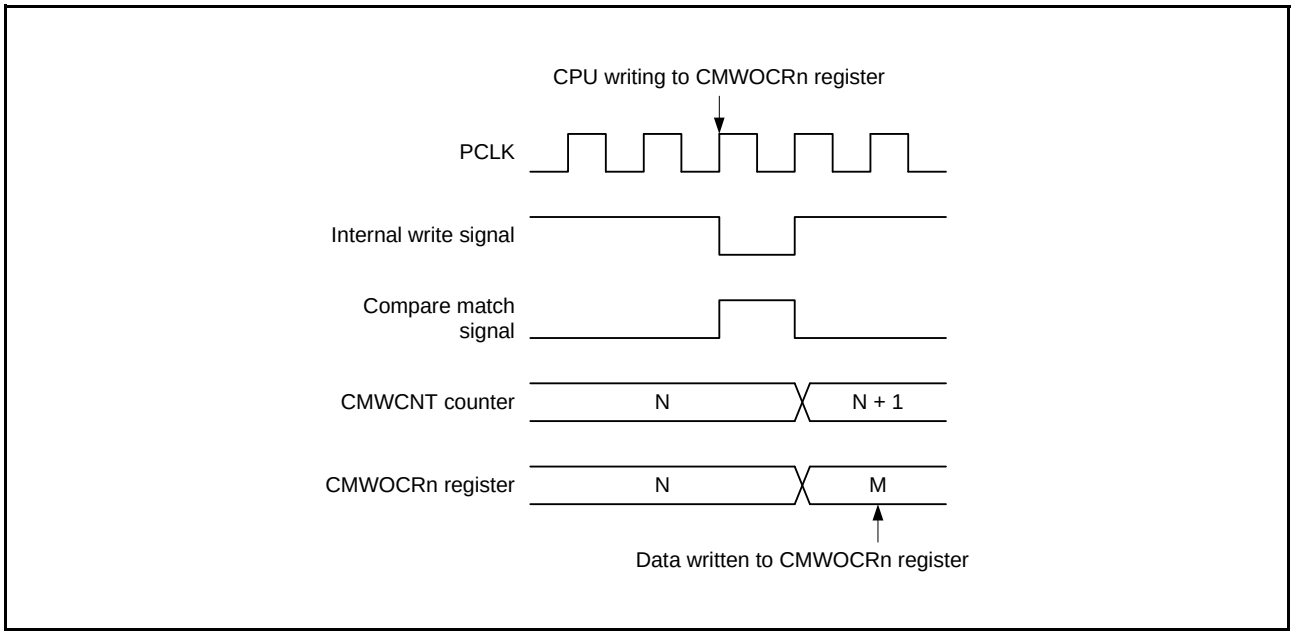


Figure 30.26 Conflict between CMWOCRn Register Writing and Compare Match

30.6.6 Conflict between CMWCNT Counter Reading and Incrementing or Clearing

If the CMWCNT counter incrementing or clearing process occurs at the same time that the data of the CMWCNT counter is read, the value having been in the CMWCNT counter before incremented or cleared is read.
Figure 30.27 shows the timing of conflict between CMWCNT counter reading and incrementing.

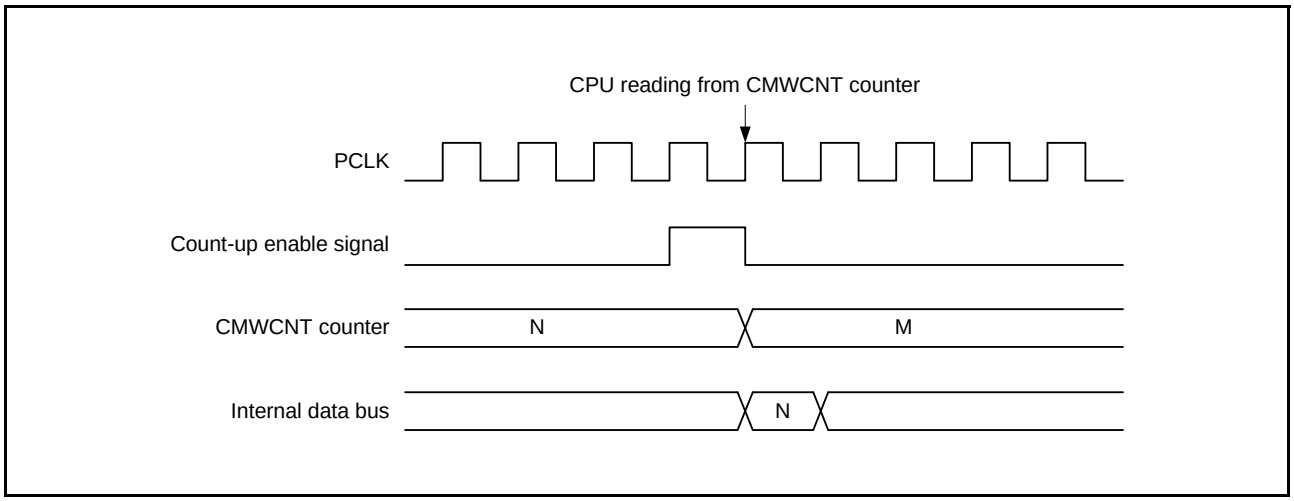


Figure 30.27 Conflict between CMWCNT Counter Reading and Incrementing

30.6.7 Conflict between CMWICRn Register Reading and Input Capture (n = 0, 1)

If the input capture detection signal is generated at the same time that the data of the CMWICRn register is read, the value having been in the CMWICRn register before updated by input capture transfer is read.
Figure 30.28 shows the timing of conflict between CMWICRn register reading and input capture.

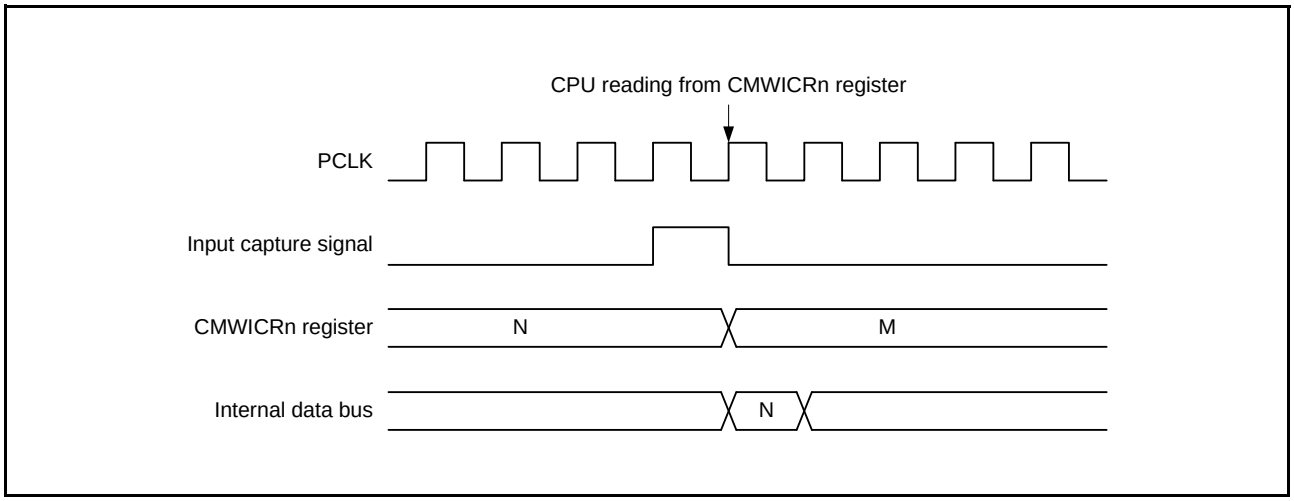


Figure 30.28 Conflict between CMWICRn Register Reading and Input Capture

31. Realtime Clock (RTCd)

In this section, “PCLK” is used to refer to PCLKB.

31.1 Overview

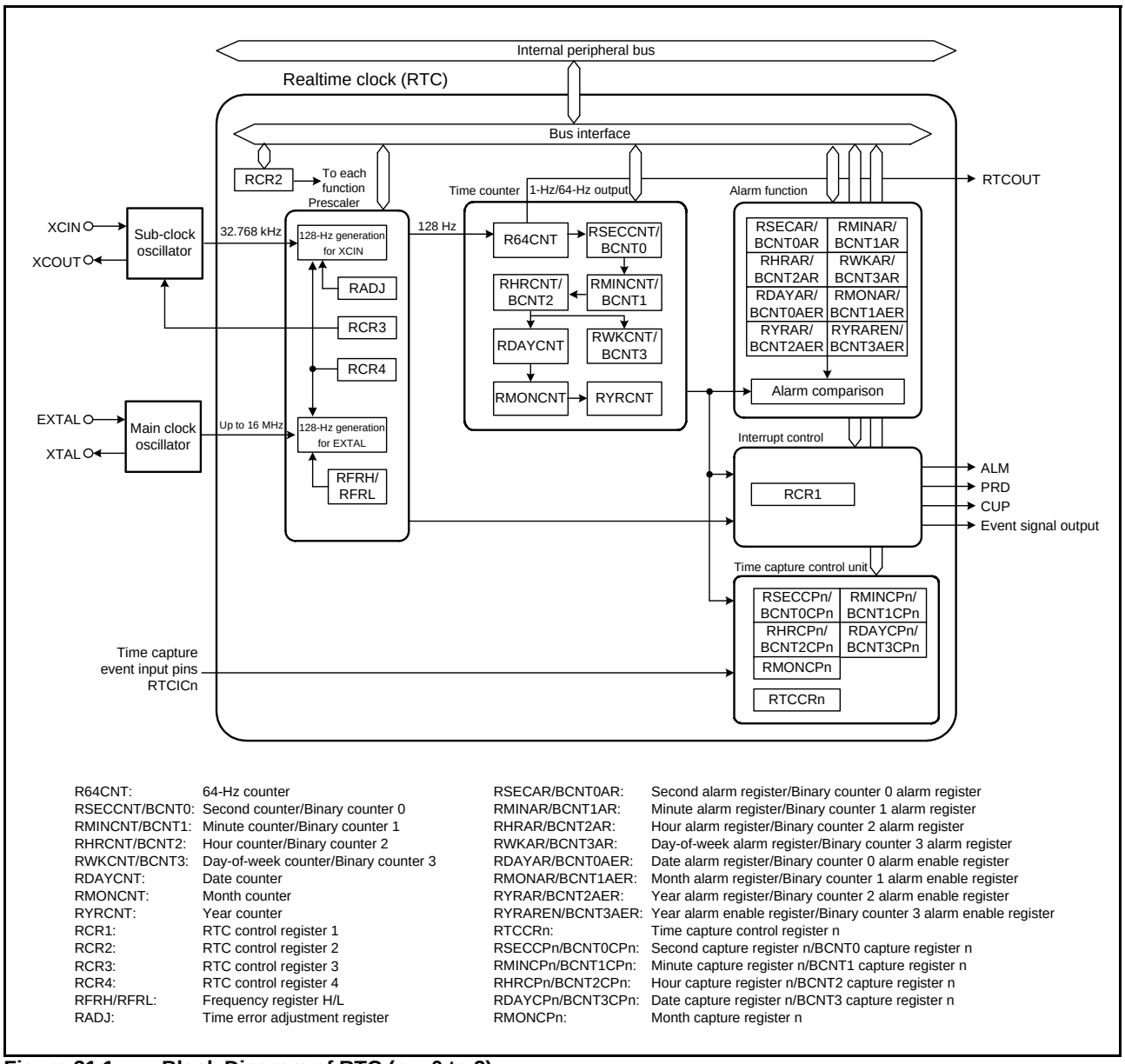
The RTC has two types of counting modes: calendar count mode and binary count mode. They are used by switching the register settings.
For calendar count mode, the RTC has a 100 year calendar from 2000 to 2099 and automatically adjusts dates for leap years.
For binary count mode, the RTC does not count in terms of years, months, dates, day-of-week, hours, or minutes; it counts seconds, and retains the information as a serial value. This mode can be used for calendars other than the Gregorian calendar.
The count source of the time counters is selectable as the sub-clock or main clock.
The RTC uses the 128-Hz clock which is acquired by the count source divided by the prescaler as the reference clock.
Year, month, date, day-of-week, a.m./p.m. (in 12-hour mode), hour, minute, second, or 32-bit binary is counted in 1/128 second units.

Table 31.1 lists the specifications of the RTC, Figure 31.1 shows a block diagram of the RTC, and Table 31.2 shows the pin configuration of the RTC.

Table 31.1 RTC Specifications

Item	Description
Count mode	Calendar count mode/binary count mode
Count source*1	Sub-clock (XCIN) or main clock (EXTAL)
Clock and calendar functions	• Calendar count mode Year, month, date, day-of-week, hour, minute, second are counted, BCD display 12 hours/24 hours mode switching function 30 seconds adjustment function (a number less than 30 is rounded down to 00 seconds, and 30 seconds or more are rounded up to one minute) Automatic adjustment function for leap years • Binary count mode Count seconds in 32 bits, binary display • Common to both modes Start/stop function The sub-second digit is displayed in binary units (1 Hz, 2 Hz, 4 Hz, 8 Hz, 16 Hz, 32 Hz, or 64 Hz). Clock error correction function Clock (1 Hz/64 Hz) output
Interrupts	• Alarm interrupt (ALM) As an alarm interrupt condition, selectable which of the below is compared with: - Calendar count mode: Year, month, date, day-of-week, hour, minute, or second can be selected - Binary count mode: Each bit of the 32-bit binary counter • Periodic interrupt (PRD) 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, 1/128 second, or 1/256 second can be selected as an interrupt period. • Carry interrupt (CUP) An interrupt is generated at either of the following timings: - When a carry from the 64-Hz counter to the second counter is generated. - When the 64-Hz counter is changed and the R64CNT register is read at the same time. • Recovery from software standby mode or deep software standby mode can be performed by an alarm interrupt or periodic interrupt
Time capture function	• Times can be captured when the edge of the time capture event input pin is detected. For every event input, month, date, hour, minute, and second are captured or 32-bit binary counter value is captured.
Event link function	Periodic event output

Note 1. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ the frequency of the count source.



R64CNT:	64-Hz counter	RSECAR/BCNT0AR:	Second alarm register/Binary counter 0 alarm register
RSECCNT/BCNT0:	Second counter/Binary counter 0	RMINAR/BCNT1AR:	Minute alarm register/Binary counter 1 alarm register
RMINCNT/BCNT1:	Minute counter/Binary counter 1	RHRAR/BCNT2AR:	Hour alarm register/Binary counter 2 alarm register
RHRCNT/BCNT2:	Hour counter/Binary counter 2	RWKAR/BCNT3AR:	Day-of-week alarm register/Binary counter 3 alarm register
RWKCNT/BCNT3:	Day-of-week counter/Binary counter 3	RDAYAR/BCNT0AER:	Date alarm register/Binary counter 0 alarm enable register
RDAYCNT:	Date counter	RMONAR/BCNT1AER:	Month alarm register/Binary counter 1 alarm enable register
RMONCNT:	Month counter	RYRAR/BCNT2AER:	Year alarm register/Binary counter 2 alarm enable register
RYRCNT:	Year counter	RYRAREN/BCNT3AER:	Year alarm enable register/Binary counter 3 alarm enable register
RCR1:	RTC control register 1	RTCCRn:	Time capture control register n
RCR2:	RTC control register 2	RSECCPn/BCNT0CPn:	Second capture register n/BCNT0 capture register n
RCR3:	RTC control register 3	RMINCPn/BCNT1CPn:	Minute capture register n/BCNT1 capture register n
RCR4:	RTC control register 4	RHRCPn/BCNT2CPn:	Hour capture register n/BCNT2 capture register n
RFRH/RFRL:	Frequency register H/L	RDAYCPn/BCNT3CPn:	Date capture register n/BCNT3 capture register n
RADJ:	Time error adjustment register	RMONCPn:	Month capture register n

Figure 31.1 Block Diagram of RTC (n = 0 to 2)

Table 31.2 Pin Configuration of RTC

Pin Name	I/O	Function
XTAL	Output	These pins are used to connect a crystal. The EXTAL pin can also be used to input an external clock. For details, refer to section 9.3.2, External Clock Input.
EXTAL	Input	
XCIN	Input	Connect a 32.768-kHz crystal to these pins.
XCOUT	Output	
RTCOUT	Output	This pin is used to output a 1-Hz/64-Hz waveform, but not in deep software standby mode.
RTCIC0	Input	Time capture event input pins
RTCIC1	Input	
RTCIC2	Input	

31.2 Register Descriptions

When writing to or reading from RTC registers, do so in accordance with section 31.6.5, Notes on Writing to and Reading from Registers.

If the value in an RTC register after a reset is given as x (undefined bits) in the list, it is not initialized by a reset. When RTC enters the reset state or a low power consumption state during counting operations (i.e. while the RCR2.START bit is 1), the year, month, day of the week, date, hours, minutes, seconds, and 64-Hz counters continue to operate. However, when the count source is the main clock, the main clock is not stopped by a reset other than a deep software standby reset. For details, refer to Table 6.2, Targets to be Initialized by Each Reset Source. Note that a reset generated during writing to or updating of a register might destroy the register value. In addition, do not allow the chip to enter software standby mode or deep software standby mode immediately after setting any of these registers. For details, refer to section 31.6.4, Transitions to Low Power Consumption Modes after Setting Registers.

31.2.1 64-Hz Counter (R64CNT)

Address(es): RTC.R64CNT 0008 C400h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	F1HZ	F2HZ	F4HZ	F8HZ	F16HZ	F32HZ	F64HZ
Value after reset:	0	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	F64HZ	64 Hz	Indicate the state between 1 Hz and 64 Hz of the sub-second digit.	R
b1	F32HZ	32 Hz		R
b2	F16HZ	16 Hz		R
b3	F8HZ	8 Hz		R
b4	F4HZ	4 Hz		R
b5	F2HZ	2 Hz		R
b6	F1HZ	1 Hz		R
b7	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R

The R64CNT counter is used in both calendar count mode and in binary count mode.

The 64-Hz counter (R64CNT) generates a period of one second by counting the 128-Hz reference clock.

The state in the sub-second range can be confirmed by reading this counter.

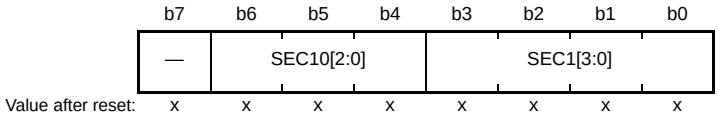
This counter is set to 00h by an RTC software reset or executing 30-second adjustment.

To read this counter, follow the procedure in section 31.3.5, Reading 64-Hz Counter and Time.

31.2.2 Second Counter (RSECCNT)/Binary Counter 0 (BCNT0)

(1) In calendar count mode:

Address(es): RTC.RSECCNT 0008 C402h



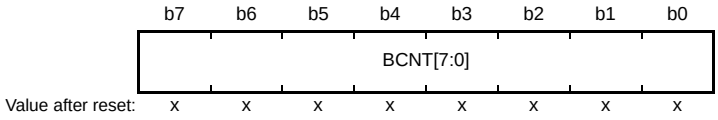
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SEC1[3:0]	1-Second Count	Counts from 0 to 9 every second. When a carry is generated, 1 is added to the tens place.	R/W
b6 to b4	SEC10[2:0]	10-Second Count	Counts from 0 to 5 for 60-second counting.	R/W
b7	—	Reserved	Set this bit to 0. It is read as the set value.	R/W

The RSECCNT counter is used for setting and counting the BCD-coded second value. It counts carries generated once per second in the 64-Hz counter.
The setting range is decimal 00 to 59. The RTC will not operate normally if any other value is set. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
After writing to the RSECCNT register, confirm that its value has actually changed before proceeding with further processing. Refer to section 31.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

(2) In binary count mode:

Address(es): RTC.BCNT0 0008 C402h



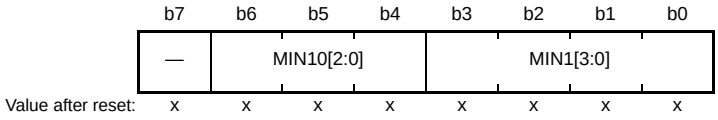
x: Undefined

The BCNT0 counter is a readable/writable 32-bit binary counter b7 to b0.
The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 31.3.5, Reading 64-Hz Counter and Time.

31.2.3 Minute Counter (RMINCNT)/Binary Counter 1 (BCNT1)

(1) In calendar count mode:

Address(es): RTC.RMINCNT 0008 C404h



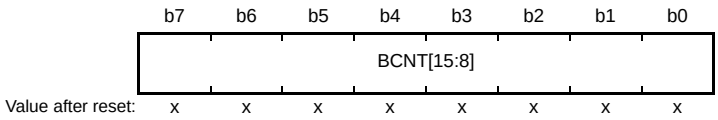
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MIN1[3:0]	1-Minute Count	Counts from 0 to 9 every minute. When a carry is generated, 1 is added to the tens place.	R/W
b6 to b4	MIN10[2:0]	10-Minute Count	Counts from 0 to 5 for 60-minute counting.	R/W
b7	—	Reserved	Set this bit to 0. It is read as the set value.	R/W

The RMINCNT counter is used for setting and counting the BCD-coded minute value. It counts carries generated once per minute in the second counter.
A value from 00 through 59 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
After writing to the RMINCNT counter, confirm that its value has actually changed before proceeding with further processing. Refer to section 31.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

(2) In binary count mode:

Address(es): RTC.BCNT1 0008 C404h



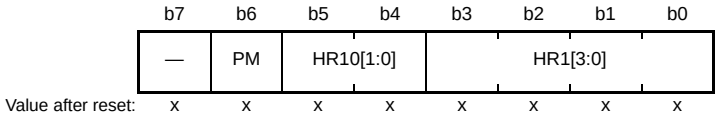
x: Undefined

The BCNT1 counter is a readable/writable 32-bit binary counter b15 to b8.
The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 31.3.5, Reading 64-Hz Counter and Time.

31.2.4 Hour Counter (RHRCNT)/Binary Counter 2 (BCNT2)

(1) In calendar count mode:

Address(es): RTC.RHRCNT 0008 C406h



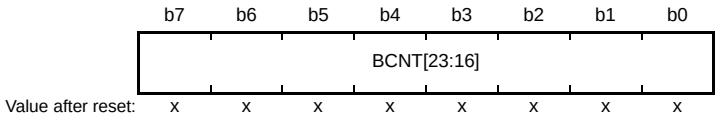
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	HR1[3:0]	1-Hour Count	Counts from 0 to 9 once per hour. When a carry is generated, 1 is added to the tens place.	R/W
b5, b4	HR10[1:0]	10-Hour Count	Counts from 0 to 2 once per carry from the ones place.	R/W
b6	PM	PM	Time Counter Setting for a.m./p.m. 0: a.m. 1: p.m.	R/W
b7	—	Reserved	Set this bit to 0. It is read as the set value.	R/W

The RHRCNT counter is used for setting and counting the BCD-coded hour value. It counts carries generated once per hour in the minute counter.
The specifiable time differs according to the setting in the hours mode bit (RCR2.HR24).
When the RCR2.HR24 bit is 0: From 00 to 11 (in BCD)
When the RCR2.HR24 bit is 1: From 00 to 23 (in BCD)
If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
The PM bit is only enabled when the RCR2.HR24 bit is 0. Otherwise, the setting in the PM bit has no effect.
After writing to the RHRCNT counter, confirm that its value has actually changed before proceeding with further processing. Refer to section 31.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

(2) In binary count mode:

Address(es): RTC.BCNT2 0008 C406h



x: Undefined

The BCNT2 counter is a readable/writable 32-bit binary counter b23 to b16.
The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 31.3.5, Reading 64-Hz Counter and Time.

31.2.5 Day-of-Week Counter (RWKCNT)/Binary Counter 3 (BCNT3)

(1) In calendar count mode:

Address(es): RTC.RWKCNT 0008 C408h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	DAYW[2:0]		
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	DAYW[2:0]	Day-of-Week Counting	b2 b0 0 0 0: Sunday 0 0 1: Monday 0 1 0: Tuesday 0 1 1: Wednesday 1 0 0: Thursday 1 0 1: Friday 1 1 0: Saturday 1 1 1: Setting Prohibited	R/W
b7 to b3	—	Reserved	Set these bits to 0. They are read as the set value.	R/W

The RWKCNT counter is used for setting and counting in the coded day-of-week value. It counts carries generated once per day in the hour counter.
A value from 0 through 6 can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
Refer to section 31.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

(2) In binary count mode:

Address(es): RTC.BCNT3 0008 C408h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNT[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT3 counter is a readable/writable 32-bit binary counter b31 to b24.
The 32-bit binary counter performs count operation by a carry generating for each second of the 64-Hz counter.
Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
To read this counter, follow the procedure in section 31.3.5, Reading 64-Hz Counter and Time.

31.2.6 Date Counter (RDAYCNT)

Address(es): RTC.RDAYCNT 0008 C40Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	DATE10[1:0]		DATE1[3:0]			
Value after reset:	0	0	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DATE1[3:0]	1-Day Count	Counts from 0 to 9 once per day. When a carry is generated, 1 is added to the tens place.	R/W
b5, b4	DATE10[1:0]	10-Day Count	Counts from 0 to 3 once per carry from the ones place.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RDAYCNT counter is used in calendar count mode.
The RDAYCNT counter is used for setting and counting the BCD-coded date value. It counts carries generated once per day in the hour counter. The count operation depends on the month and whether the year is a leap year.
Leap years are determined according to whether the year counter (RYRCNT) value is divisible by 400, 100, and 4.
A value from 01 through 31 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. (When specifying a value, note that the range of specifiable days depends on the month and whether the year is a leap year.) Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
After writing to the RHRCNT counter, confirm that its value has actually changed before proceeding with further processing. Refer to section 31.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

31.2.7 Month Counter (RMONCNT)

Address(es): RTC.RMONCNT 0008 C40Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	MON10	MON1[3:0]			
Value after reset:	0	0	0	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MON1[3:0]	1-Month Count	Counts from 0 to 9 once per month. When a carry is generated, 1 is added to the tens place.	R/W
b4	MON10	10-Month Count	Counts from 0 to 1 once per carry from the ones place.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RMONCNT counter is used in calendar count mode.
The RMONCNT counter is used for setting and counting the BCD-coded month value. It counts carries generated once per month in the date counter.
A value from 01 through 12 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.
After writing to the RMONCNT counter, confirm that its value has actually changed before proceeding with further processing. Refer to section 31.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

31.2.8 Year Counter (RYRCNT)

Address(es): RTC.RYRCNT 0008 C40Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	YR10[3:0]				YR1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	YR1[3:0]	1-Year Count	Counts from 0 to 9 once per year. When a carry is generated, 1 is added to the tens place.	R/W
b7 to b4	YR10[3:0]	10-Year Count	Counts from 0 to 9 once per carry from ones place. When a carry is generated in the tens place, 1 is added to the hundreds place.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RYRCNT counter is used in calendar count mode.

The RYRCNT counter is used for setting and counting the BCD-coded year value. It counts carries generated once per year in the month counter.

A value from 00 through 99 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

After writing to the RYRCNT counter, confirm that its value has actually changed before proceeding with further processing. Refer to section 31.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

31.2.9 Second Alarm Register (RSECAR)/Binary Counter 0 Alarm Register (BCNT0AR)

(1) In calendar count mode:

Address(es): RTC.RSECAR 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	0: The register value is not compared with the RSECCNT counter value. 1: The register value is compared with the RSECCNT counter value.	R/W

The RSECAR register is an alarm register corresponding to the BCD-coded second counter RSECCNT. When the ENB bit is set to 1, the RSECAR value is compared with the RSECCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.
RSECAR values from 00 through 59 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
After writing to the RSECAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 31.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.
This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT0AR 0008 C410h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTAR[7:0]							
Value after reset:	x	x	x	x	x	x	x	x

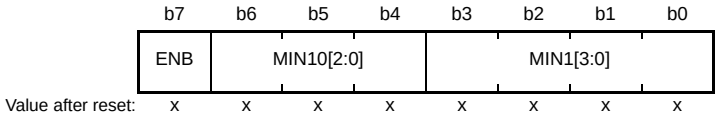
x: Undefined

The BCNT0AR counter is a readable/writable alarm register corresponding to 32-bit binary counter b7 to b0.
This register is set to 00h by an RTC software reset.

31.2.10 Minute Alarm Register (RMINAR)/Binary Counter 1 Alarm Register (BCNT1AR)

(1) In calendar count mode:

Address(es): RTC.RMINAR 0008 C412h



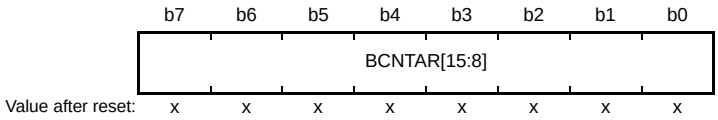
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MIN1[3:0]	1 Minute	Value for the ones place of minutes	R/W
b6 to b4	MIN10[2:0]	10 Minutes	Value for the tens place of minutes	R/W
b7	ENB	ENB	0: The register value is not compared with the RMINCNT counter value. 1: The register value is compared with the RMINCNT counter value.	R/W

The RMINAR register is an alarm register corresponding to the BCD-coded minute counter RMINCNT. When the ENB bit is set to 1, the RMINAR value is compared with the RMINCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.
RMINAR values from 00 through 59 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
After writing to the RMINAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 31.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.
This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT1AR 0008 C412h



x: Undefined

The BCNT1AR counter is a readable/writable alarm register corresponding to 32-bit binary counter b15 to b8.
This register is set to 00h by an RTC software reset.

31.2.11 Hour Alarm Register (RHRAR)/Binary Counter 2 Alarm Register (BCNT2AR)

(1) In calendar count mode:

Address(es): RTC.RHRAR 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	PM	Time Alarm Setting for a.m./p.m. 0: a.m. 1: p.m.	R/W
b7	ENB	ENB	0: The register value is not compared with the RHRCNT counter value. 1: The register value is compared with the RHRCNT counter value.	R/W

The RHRAR register is an alarm register corresponding to the BCD-coded hour counter RHRCNT. When the ENB bit is set to 1, the RHRAR value is compared with the RHRCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.
The specifiable time differs according to the setting in the hours mode bit (RCR2.HR24).
When the RCR2.HR24 bit is 0: From 00 to 11 (in BCD)
When the RCR2.HR24 bit is 1: From 00 to 23 (in BCD)
If a value outside of this range is specified, the RTC does not operate correctly.
When the RCR2.HR24 bit is 0, be sure to set the PM bit.
When the RCR2.HR24 bit is 1, the setting in the PM bit has no effect.
After writing to the RHRAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 31.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.
This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT2AR 0008 C414h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTAR[23:16]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT2AR counter is a readable/writable alarm register corresponding to 32-bit binary counter b23 to b16.
This register is set to 00h by an RTC software reset.

31.2.12 Day-of-Week Alarm Register (RWKAR)/Binary Counter 3 Alarm Register (BCNT3AR)

(1) In calendar count mode:

Address(es): RTC.RWKAR 0008 C416h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	—	—	—	DAYW[2:0]		
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	DAYW[2:0]	Day-of-Week Setting	b2 b0 0 0 0: Sunday 0 0 1: Monday 0 1 0: Tuesday 0 1 1: Wednesday 1 0 0: Thursday 1 0 1: Friday 1 1 0: Saturday 1 1 1: Setting Prohibited	R/W
b6 to b3	—	Reserved	Set these bits to 0. They are read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RWKCNT counter value. 1: The register value is compared with the RWKCNT counter value.	R/W

The RWKAR register is an alarm register corresponding to the coded day-of-week counter RWKCNT. When the ENB bit is set to 1, the RWKAR value is compared with the RWKCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.

RWKAR values from 0 through 6 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

After writing to the RWKAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 31.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT3AR 0008 C416h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTAR[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT3AR counter is a readable/writable alarm register corresponding to 32-bit binary counter b31 to b24.

This register is set to 00h by an RTC software reset.

31.2.13 Date Alarm Register (RDAYAR)/Binary Counter 0 Alarm Enable Register (BCNT0AER)

(1) In calendar count mode:

Address(es): RTC.RDAYAR 0008 C418h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	DATE10[1:0]		DATE1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DATE1[3:0]	1 Day	Value for the ones place of days	R/W
b5, b4	DATE10[1:0]	10 Days	Value for the tens place of days	R/W
b6	—	Reserved	Set this bit to 0. It is read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RDAYCNT counter value. 1: The register value is compared with the RDAYCNT counter value.	R/W

The RDAYAR register is an alarm register corresponding to the BCD-coded date counter RDAYCNT. When the ENB bit is set to 1, the RDAYAR value is compared with the RDAYCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.

RDAYAR values from 01 through 31 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

After writing to the RDAYAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 31.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.

This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT0AER 0008 C418h

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB[7:0]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT0AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b7 to b0. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the IR flag corresponding to the ALM interrupt becomes 1.

This register is set to 00h by an RTC software reset.

31.2.14 Month Alarm Register (RMONAR)/Binary Counter 1 Alarm Enable Register (BCNT1AER)

(1) In calendar count mode:

Address(es): RTC.RMONAR 0008 C41Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB	—	—	MON10	MON1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MON1[3:0]	1 Month	Value for the ones place of months	R/W
b4	MON10	10 Months	Value for the tens place of months	R/W
b6, b5	—	Reserved	Set these bits to 0. They are read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RMONCNT counter value. 1: The register value is compared with the RMONCNT counter value.	R/W

The RMONAR register is an alarm register corresponding to the BCD-coded month counter RMONCNT. When the ENB bit is set to 1, the RMONAR value is compared with the RMONCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.
RMONAR values from 01 through 12 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
After writing to the RMONAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 31.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.
This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT1AER 0008 C41Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB[15:8]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT1AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b15 to b8. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the IR flag corresponding to the ALM interrupt becomes 1.
This register is set to 00h by an RTC software reset.

31.2.15 Year Alarm Register (RYRAR)/Binary Counter 2 Alarm Enable Register (BCNT2AER)

(1) In calendar count mode:

Address(es): RTC.RYRAR 0008 C41Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	YR10[3:0]				YR1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	YR1[3:0]	1 Year	Value for the ones place of years	R/W
b7 to b4	YR10[3:0]	10 Years	Value for the tens place of years	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RYRAR register is an alarm register corresponding to the BCD-coded year counter RYRCNT.
RYRAR values from 00 through 99 (in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.
After writing to the RYRAR register, confirm that its value has actually changed before proceeding with further processing. Refer to section 31.6.5, Notes on Writing to and Reading from Registers for notes on accessing registers.
This register is set to 0000h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT2AER 0008 C41Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ENB[23:16]							
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

x: Undefined

The BCNT2AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b23 to b16. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the IR flag corresponding to the ALM interrupt becomes 1.
This register is set to 0000h by an RTC software reset.

31.2.16 Year Alarm Enable Register (RYRAREN)/Binary Counter 3 Alarm Enable Register (BCNT3AER)

(1) In calendar count mode:

Address(es): RTC.RYRAREN 0008 C41Eh

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

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	Set these bits to 0. They are read as the set value.	R/W
b7	ENB	ENB	0: The register value is not compared with the RYRCNT counter value. 1: The register value is compared with the RYRCNT counter value.	R/W

When the ENB bit in the RYRAREN register is set to 1, the RYRAR value is compared with the RYRCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the IR flag corresponding to the ALM interrupt is set to 1.
This register is set to 00h by an RTC software reset.

(2) In binary count mode:

Address(es): RTC.BCNT3AER 0008 C41Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	ENB[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

The BCNT3AER register is a readable/writable register for setting the alarm enable corresponding to 32-bit binary counter b31 to b24. Among the ENB[31:0] bits, the binary counter (BCNT[31:0]) corresponding to the bits which are set to 1 and the binary alarm register (BCNTAR[31:0]) are compared, and when all match, the IR flag corresponding to the ALM interrupt becomes 1.
This register is set to 00h by an RTC software reset.

31.2.17 RTC Control Register 1 (RCR1)

Address(es): RTC.RCR1 0008 C422h

	b7	b6	b5	b4	b3	b2	b1	b0
	PES[3:0]				RTCOS	PIE	CIE	AIE
Value after reset:	x	x	x	x	0	x	0	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	AIE	Alarm Interrupt Enable	0: An alarm interrupt request is disabled. 1: An alarm interrupt request is enabled.	R/W
b1	CIE	Carry Interrupt Enable	0: A carry interrupt request is disabled. 1: A carry interrupt request is enabled.	R/W
b2	PIE	Periodic Interrupt Enable	0: A periodic interrupt request is disabled. 1: A periodic interrupt request is enabled.	R/W
b3	RTCOS	RTCOUT Output Select	0: RTCOUT outputs 1 Hz. 1: RTCOUT outputs 64 Hz.	R/W
b7 to b4	PES[3:0]	Periodic Interrupt Select	b7 b4 0 1 1 0: A periodic interrupt is generated every 1/256 second.*1 0 1 1 1: A periodic interrupt is generated every 1/128 second. 1 0 0 0: A periodic interrupt is generated every 1/64 second. 1 0 0 1: A periodic interrupt is generated every 1/32 second. 1 0 1 0: A periodic interrupt is generated every 1/16 second. 1 0 1 1: A periodic interrupt is generated every 1/8 second. 1 1 0 0: A periodic interrupt is generated every 1/4 second. 1 1 0 1: A periodic interrupt is generated every 1/2 second. 1 1 1 0: A periodic interrupt is generated every 1 second. 1 1 1 1: A periodic interrupt is generated every 2 seconds. Other than above: No periodic interrupts are generated.	R/W

Note 1. When the main clock is selected (RCR4.RCKSEL bit = 1) while the PES[3:0] bits are 0110b, a periodic interrupt is generated every 1/128 second.

The RCR1 register is used in both calendar count mode and in binary count mode.
Bits AIE, PIE, and PES[3:0] are updated synchronously with the count source. When the RCR1 register is modified, check that all the bits have been updated before proceeding to the next processing.

AIE Bit (Alarm Interrupt Enable)

This bit enables or disables alarm interrupt requests.
If the times indicated by the counters and alarm settings match in deep software standby mode, the MCU returns from the mode regardless of the value of the AIE bit.

CIE Bit (Carry Interrupt Enable)

This bit enables and disables interrupt requests when a carry to the RSECCNT/BCNT0 register occurs, or when a carry to the 64-Hz counter (R64CNT) occurs while reading the 64-Hz counter.

PIE Bit (Periodic Interrupt Enable)

This bit enables or disabled a periodic interrupt.
If the periods indicated by the counters and PES[3:0] settings match in deep software standby mode, the MCU returns from the mode regardless of the value of the PIE bit.

RTCOS Bit (RTCOUT Output Select)

This bit selects the RTCOUT output period. The RTCOS bit must be rewritten while count operation is stopped (the RCR2.START bit is 0) and RTCOUT output is disabled (the RCR2.RTCOE bit is 0). When the RTCOUT is output to an external pin, the RCR2.RTCOE bit must be enabled. For details on controlling I/O ports, refer to section 23.4.1, Procedure for Specifying Input/Output Pin Function.

PES[3:0] Bits (Periodic Interrupt Select)

These bits specify the period for the periodic interrupt. A periodic interrupt is generated with the period specified by these bits.

31.2.18 RTC Control Register 2 (RCR2)

Address(es): RTC.RCR2 0008 C424h

	b7	b6	b5	b4	b3	b2	b1	b0
	CNTMD	HR24	AADJP	AADJE	RTCOE	ADJ30	RESET	START
Value after reset:	x	x	x	x	0	0	0	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	START	Start*4	0: Prescaler and counter are stopped. 1: Prescaler and counter operate normally.	R/W
b1	RESET	RTC Software Reset	• In writing 0: Writing is invalid. 1: The prescaler and the target registers for RTC software reset*1 are initialized. • In reading 0: In normal time operation, or an RTC software reset has completed. 1: During an RTC software reset	R/W
b2	ADJ30	30-Second Adjustment*2	• In writing 0: Writing is invalid. 1: 30-second adjustment is executed. • In reading 0: In normal time operation, or 30-second adjustment has completed. 1: During 30-second adjustment	R/W
b3	RTCOE	RTCOUT Output Enable	0: RTCOUT output disabled. 1: RTCOUT output enabled.	R/W
b4	AADJE	Automatic Adjustment Enable*3, *4	0: Automatic adjustment is disabled. 1: Automatic adjustment is enabled.	R/W
b5	AADJP	Automatic Adjustment Period Select*3, *4	0: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every minute (every 32 seconds in binary counter mode). 1: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every 10 seconds (every 8 seconds in binary counter mode).	R/W
b6	HR24	Hours Mode*2, *4	0: The RTC operates in 12-hour mode. 1: The RTC operates in 24-hour mode.	R/W
b7	CNTMD	Count Mode Select*4	0: The calendar count mode. 1: The binary count mode.	R/W

Note 1. R64CNT, RSECAR/BCNT0AR, RMINAR/BCNT1AR, RHRAR/BCNT2AR, RWKAR/BCNT3AR, RDAYAR/BCNT0AER, RMONAR/BCNT1AER, RYRAR/BCNT2AER, RYRAREN/BCNT3AER, RADJ, RTCCRn, RSECCPn/BCNT0CPn, RMINCPn/BCNT1CPn, RHRCPn/BCNT2CPn, RDAYCPn/BCNT3CPn, RMONCPn, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP

Note 2. This bit is reserved in binary counter mode. The write value should be 0.

Note 3. When the main clock is selected, the setting of this bit is disabled.

Note 4. After writing to this bit, confirm that its value has actually changed before proceeding with further processing. Refer to section 31.6.5, Notes on Writing to and Reading from Registers regarding changes to the values of the AADJE, AADJP, and HR24 bits.

The RCR2 register is related to hours mode, automatic adjustment function, enabling the RTCOUT output, 30-second adjustment, RTC software reset, and controlling count operation.

START Bit (Start)

This bit stops or restarts the prescaler or counter (clock) operation.

The START bit is updated in synchronization with the count source. When the START bit is modified, check that the bit is updated before proceeding to the next processing.

RESET Bit (RTC Software Reset)

This bit initializes the prescaler and registers to be reset by RTC software reset.
When 1 is written to the RESET bit, the initialization starts in synchronization with the count source. When the initialization is completed, the RESET bit is automatically set to 0.
When 1 is written to the RESET bit, check that the bit is set to 0, and then make next settings.

ADJ30 Bit (30-Second Adjustment)

This bit is for 30-second adjustment.
When 1 is written to the ADJ30 bit, the RSECCNT value of 30 seconds or less is rounded down to 00 second and the value of 30 seconds or more is rounded up to 1 minute.
The 30-second adjustment is performed in synchronization with the count source. When 1 is written to this bit, the ADJ30 bit is automatically set to 0 after the 30-second adjustment is completed. In case when 1 is written to the ADJ30 bit, check that the bit is set to 0, and then make next settings.
When the 30-second adjustment is performed, the prescaler and R64CNT are also reset.
The ADJ30 bit is set to 0 by an RTC software reset.
This bit is reserved in binary counter mode. The write value should be 0.

RTCOE Bit (RTCOUT Output Enable)

This bit enables output of a 1-Hz/64-Hz clock signal from the RTCOUT pin.
Use the START bit to stop counting by the counters before changing the value of the RTCOE bit. Do not stop counting (write 0 to the START bit) and change the value of the RTCOE bit at the same time.
When RTCOUT signal is to be output from an external pin, set the RTCOE bit to 1 and set up the port control for the pin.

AADJE Bit (Automatic Adjustment Enable)

This bit controls (enables or disables) automatic adjustment.
Set the plus–minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJE bit.
The AADJE bit is set to 0 by an RTC software reset.

AADJP Bit (Automatic Adjustment Period Select)

This bit selects the automatic-adjustment period.
Set the plus–minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJP bit.
The AADJP bit is set to 0 by an RTC software reset.

HR24 Bit (Hours Mode)

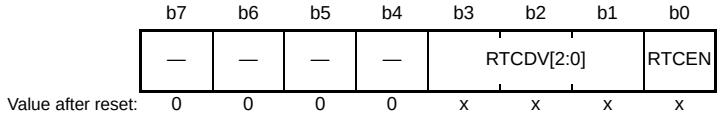
This bit specifies whether the RTC will operate in 12- or 24-hour mode.
Use the START bit to stop counting by the counters before changing the value of the HR24 bit. Do not stop counting (write 0 to the START bit) and change the value of the HR24 bit at the same time.
This bit is reserved in binary counter mode. The write value should be 0.

CNTMD Bit (Count Mode Select)

This bit specifies whether the RTC count mode is operated in calendar count mode or in binary count mode.
After setting the count mode, execute an RTC software reset and start again from the initial settings.
The CNTMD bit is updated in synchronization with the count source, so when the value of the CNTMD bit has been changed, check that the value of the bit has actually been updated before applying the RTC software reset. The count mode changes to that which was specified beforehand in the CNTMD bit after the RTC software reset is applied.
For details on initial settings, refer to section 31.3.1, Outline of Initial Settings of Registers after Power On.

31.2.19 RTC Control Register 3 (RCR3)

Address(es): RTC.RCR3 0008 C426h



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	RTCEN	Sub-Clock Oscillator Control	0: Sub-clock oscillator is stopped. 1: Sub-clock oscillator is operating.	R/W
b3 to b1	RTCDV[2:0]	Sub-Clock Oscillator Drive Capacity Control	b3 b2 b1 0 0 0: Setting prohibited 0 0 1: Drive capacity for low CL 0 1 0: Setting prohibited 0 1 1: Setting prohibited 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: Drive capacity for standard CL 1 1 1: Setting prohibited	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RCR3 register is used for controlling the sub-clock oscillator in the clock generation circuit. For details on controlling the sub-clock oscillator, refer to section 9, Clock Generation Circuit.
This register is a function common to calendar count mode and binary count mode.
When this register is modified, check that all the bits have been updated before proceeding to the next processing.

RTCEN Bit (Sub-Clock Oscillator Control)

The RTCEN bit and a clock generation circuit register (the SOSCCR.SOSTP bit) control whether to operate or stop the sub-clock oscillator. If one of the bits is set so as to enable the operation, the sub-clock oscillator runs.
When using the sub-clock as the count source to the RTC, set the sub-clock oscillator using the RTCEN bit.
When the main clock is selected (RCR4.RCKSEL bit = 1), making the sub-clock oscillator run or stop is controlled only by the SOSCCR.SOSTP bit regardless of the value set in the RTCEN bit.

RTCDV[2:0] Bits (Sub-Clock Oscillator Drive Capacity Control)

These bits control the drive capacity of the sub-clock oscillator.
Before modifying the RTCDV[2:0] bits, be sure to stop the sub-clock oscillator. When the main clock is selected (RCR4.RCKSEL bit = 1), making the sub-clock oscillator run or stop is controlled only by the SOSCCR.SOSTP bit regardless of the value set in the RTCEN bit.

31.2.19.1 Notes on using a low CL crystal unit

When the RCR3.RTCDV[2:0] bits are 001b (drive capacity for low CL), the oscillator is susceptible to noise. Especially when the signal level of any pin near the XCIN or XCOUT pin is changed, the oscillation accuracy of the sub-clock oscillator may be affected. The accuracy is affected differently depending on the board traces and how the signal level of any pin near the XCIN or XCOUT pin is changed. When designing a board using a low CL crystal unit, refer to the application note “Design Guide for Low CL Sub-clock Circuits” (R01AN1187EJ) to reduce the influence from noise. The following are examples that may significantly affect oscillation accuracy:

(1) When connecting an on-chip debugging emulator to the FINED pin

Since the FINED pin (FINE interface pin) is near the XCIN and XCOUT pins, the oscillation accuracy of the sub-clock oscillator is affected when using the FINED pin in debugging. When using the FINED pin in debugging, set the RCR3.RTCDV[2:0] bits to 001b (drive capacity for low CL) and debug at the room temperature.

(2) When supplying an external clock to the main clock oscillator

When inputting an external clock to the EXTAL pin, the oscillation accuracy of the sub-clock oscillator may be affected. When inputting an inverted external clock to the XTAL pin, the oscillation accuracy will be affected more significantly.

31.2.20 RTC Control Register 4 (RCR4)

Address(es): RTC.RCR4 0008 C428h

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

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	RCKSEL	Count Source Select	0: Sub-clock oscillator is selected. 1: Main clock oscillator is selected.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The RCR4 register is used for selecting the count source. This function is used in both calendar count mode and binary count mode.
When the RCKSEL bit is set to 0, the time is counted with the sub-clock. When the bit is set to 1, the time is counted with the main clock.

RCKSEL Bit (Count Source Select)

This bit selects the count source from the sub-clock and the main clock.
The count source should be selected only once before the initial settings of the RTC registers at power on.

31.2.21 Frequency Register H/L (RFRH/RFRL)

Address(es): RTC.RFRH 0008 C42Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFC[16]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

x: Undefined

Address(es): RTC.RFRL 0008 C42Ch

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

x: Undefined

- RFRH register

Bit	Symbol	Bit Name	Description	R/W
b0	RFC[16]	Frequency Divide Ratio Setting	Set the divide ratio of the main clock to generate the reference clock of 128 Hz. If the setting value in the RFC[16:0] bits is n, the prescaler divides the main clock by n + 1.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- RFRL register

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	RFC[15:0]	Frequency Divide Ratio Setting	Set the divide ratio of the main clock to generate the reference clock of 128 Hz. If the setting value in the RFC[16:0] bits is n, the prescaler divides the main clock by n + 1.	R/W

RFRH/RFRL is a register for controlling the prescaler when the main clock is selected.
The RTC time counter operates on a 128-Hz clock signal as the reference clock. Therefore, when the main clock is selected, the main clock is divided by the prescaler to generate a 128-Hz clock signal. Set the frequency divide ratio in the RFC[16:0] bits to generate a 128-Hz clock from the main clock frequency. As for the calculation method, refer to the formula below.

A value from 0 0007h through 1 FFFFh can be specified in the RFC[16:0] bits; if a value outside of this range is specified, the RTC does not operate correctly. Rewrite this register while the RCR2.START bit is 0 (counter is stopped). In addition, the range of the main clock frequency that can be used to generate a 128-Hz clock is 1.024 kHz to 16.778 MHz.
The frequency of the peripheral module clock and the main clock should be in the relationship that the peripheral module clock ≥ the main clock.

- Calculation method of frequency comparison value

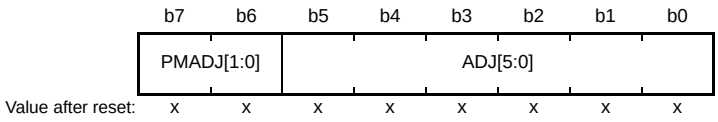
RFC[16:0] = (Main clock frequency) ÷ 128 – 1

Table 31.3 RFRH/RFRL Register Settings by the Main Clock Frequency

Main Clock Frequency	RFRH/RFRL Register Setting Value
4 MHz	0000 7A11h
8 MHz	0000 F423h
10 MHz	0001 312Ch
12 MHz	0001 6E35h
16 MHz	0001 E847h

31.2.22 Time Error Adjustment Register (RADJ)

Address(es): RTC.RADJ 0008 C42Eh



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	ADJ[5:0]	Adjustment Value	These bits specify the adjustment value from the prescaler.	R/W
b7, b6	PMADJ[1:0]	Plus–Minus	b7 b6 0 0: Adjustment is not performed. 0 1: Adjustment is performed by the addition to the prescaler. 1 0: Adjustment is performed by the subtraction from the prescaler. 1 1: Setting prohibited	R/W

Adjustment is performed by the addition to or subtraction from the prescaler.
In case when the automatic adjustment enable (RCR2.AADJE) bit is 0, adjustment is performed when writing to the RADJ.
In case when the RCR2.AADJE bit is 1, adjustment is performed in the interval specified by the automatic adjustment period select (RCR2.AADJP) bit.
The current adjustment by software (disabling automatic adjustment) may be invalid if the following adjustment value is specified within 320 cycles of the count source after the register setting. To perform adjustment consecutively, wait for 320 cycles or more of the count source after the register setting and then specify the next adjustment value.
RADJ is updated in synchronization with the count source. When RADJ is modified, check that all the bits have been updated before continuing with further processing.
This register is set to 00h by an RTC software reset.
The setting of this register is enabled only when the sub-clock is selected.
When the main clock is selected, adjustment is not performed.

ADJ[5:0] Bits (Adjustment Value)

These bits specify the adjustment value (the number of sub-clock cycles) from the prescaler.

PMADJ[1:0] Bits (Plus–Minus)

These bits select whether the clock is set ahead or back depending on the error-adjustment value set in the ADJ[5:0] bits.

31.2.23 Time Capture Control Register n (RTCCRn) (n = 0 to 2)

Address(es): RTC.RTCCR0 0008 C440h, RTC.RTCCR1 0008 C442h, RTC.RTCCR2 0008 C444h

	b7	b6	b5	b4	b3	b2	b1	b0
	TCEN	—	TCNF[1:0]		—	TCST	TCCT[1:0]	
Value after reset:	x	x	x	x	x	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	0: No event is detected. 1: An event is detected.*1	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	TCNF[1:0]	Time Capture Noise Filter Control	b5 b4 0 0: The noise filter is off. 0 1: Setting prohibited 1 0: The noise filter is on (count source). 1 1: The noise filter is on (count source by divided by 32).	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	TCEN	Time Capture Event Input Pin Enable	0: The RTCICn pin is disabled as the time capture event input. 1: The RTCICn pin is enabled as the time capture event input.	R/W

Note 1. Indicates that an event has been detected. Writing 1 to this bit has no effect. Writing 0 sets this bit to 0.

The RTCCRn register is used both in calendar count mode and in binary count mode.
RTCCR0, RTCCR1, and RTCCR2 control the RTCIC0, RTCIC1, and RTCIC2 pins, respectively.
RTCCRn is updated in synchronization with the count source. When RTCCRn is modified, check that all the bits except for the TCST bit have been updated before continuing with further processing.
This register is set to 00h by an RTC software reset.

TCCT[1:0] Bits (Time Capture Control)

These bits control the edge detection of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2). The detection edge is selectable. The TCCT[1:0] bits should be set while the TCEN bit is 1.

TCST Bit (Time Capture Status)

This bit indicates that an event of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2) has been detected.
When the TCST bit is 0, no event is detected.
When the TCST bit is 1, this bit indicates that an event of the corresponding pin has been detected and the capture register is valid. When multiple events have been detected, the capture time for the first event is retained.
If an event is detected while the count operation is stopped (the RCR2.START bit is 0), the captured value is not guaranteed. In this case, set the TCST bit to 0 for deleting the captured value.
Writing 0 sets the TCST bit to 0. In addition, writing any other value except 0 has no effect.
Set the TCST bit while the TCCT[1:0] bits are 00b (no event is detected).
The TCST bit is set to 0 in synchronization with the count source. When the TCST bit is set to 0, check that the bit has been updated before continuing with further processing.

TCNF[1:0] Bits (Time Capture Noise Filter Control)

These bits control the noise filter of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2).
When the noise filter is on, the count source divided by 1 or divided by 32 is selectable. In this case, when the input level on the time capture event input pin matches three consecutive times at the set sampling period, the input level is determined.
Set the TCNF[1:0] bits while the TCCT[1:0] bits are 00b (no event is detected). When the noise filter is used, set the TCNF[1:0] bits, wait for three cycles of the specified sampling period, and then set the TCCT[1:0] bits. Set the TCNF[1:0] bits when the TCEN bit is 1.

TCEN Bit (Time Capture Event Input Pin Enable)

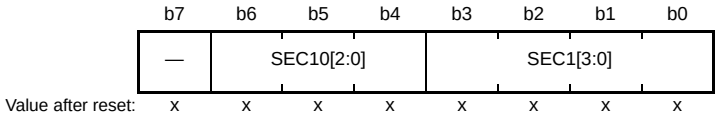
This bit enables or disables the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2).
When the sub-clock is selected (RCR4.RCKSEL bit = 0) and the sub-clock oscillator is stopped (RCR3.RTCEN bit = 0), the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2) are disabled regardless of the value of the TCEN bit.
When the time capture event input pin functions (RTCIC0, RTCIC1, and RTCIC2) are allocated to the same pins as other multiplexed functions, make the general input port pin settings for the corresponding pins, set the corresponding TAMPICR1.CHnEN bit*1 (n = 0 to 2) to 1, and then set this bit to 1. If the TCEN bit is set to 0, set also the TCCT[1:0] bits to 00b.

Note 1. For details, refer to section 12, Battery Backup Function (VBATTB).

31.2.24 Second Capture Register n (RSECCPn) (n = 0 to 2)/BCNT0 Capture Register n (BCNT0CPn) (n = 0 to 2)

(1) In calendar count mode:

Address(es): RTC.RSECCP0 0008 C452h, RTC.RSECCP1 0008 C462h, RTC.RSECCP2 0008 C472h



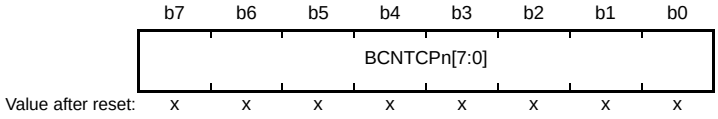
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SEC1[3:0]	1-Second Capture	Capture value for the ones place of seconds	R
b6 to b4	SEC10[2:0]	10-Second Capture	Capture value for the tens place of seconds	R
b7	—	Reserved	This bit is read as 0 after an RTC software reset.	R

RSECCPn is a read-only register that captures the RSECCNT value when a time capture event is detected.
The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RSECCP0, RSECCP1, and RSECCP2 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

(2) In binary count mode:

Address(es): RTC.BCNT0CP0 0008 C452h, RTC.BCNT0CP1 0008 C462h, RTC.BCNT0CP2 0008 C472h



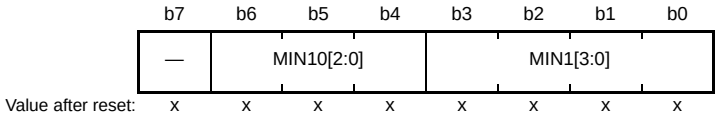
x: Undefined

BCNT0CPn is a read-only register that captures the BCNT0 value when a time capture event is detected.
The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNT0CP0, BCNT0CP1, and BCNT0CP2 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

31.2.25 Minute Capture Register n (RMINCPn) (n = 0 to 2)/BCNT1 Capture Register n (BCNT1CPn) (n = 0 to 2)

(1) In calendar count mode:

Address(es): RTC.RMINCP0 0008 C454h, RTC.RMINCP1 0008 C464h, RTC.RMINCP2 0008 C474h



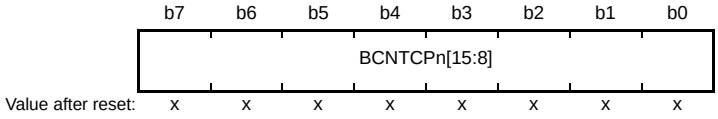
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MIN1[3:0]	1-Minute Capture	Capture value for the ones place of minutes	R
b6 to b4	MIN10[2:0]	10-Minute Capture	Capture value for the tens place of minutes	R
b7	—	Reserved	This bit is read as 0 after an RTC software reset.	R

RMINCPn is a read-only register that captures the RMINCNT value when a time capture event is detected.
The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RMINCP0, RMINCP1, and RMINCP2 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

(2) In binary count mode:

Address(es): RTC.BCNT1CP0 0008 C454h, RTC.BCNT1CP1 0008 C464h, RTC.BCNT1CP2 0008 C474h



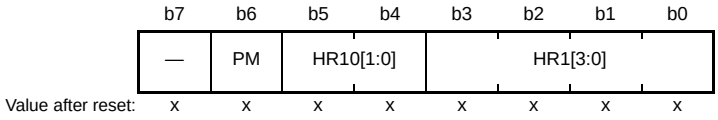
x: Undefined

BCNT1CPn is a read-only register that captures the BCNT1 value when a time capture event is detected.
The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNT1CP0, BCNT1CP1, and BCNT1CP2 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

31.2.26 Hour Capture Register n (RHRCPn) (n = 0 to 2)/BCNT2 Capture Register n (BCNT2CPn) (n = 0 to 2)

(1) In calendar count mode:

Address(es): RTC.RHRCP0 0008 C456h, RTC.RHRCP1 0008 C466h, RTC.RHRCP2 0008 C476h



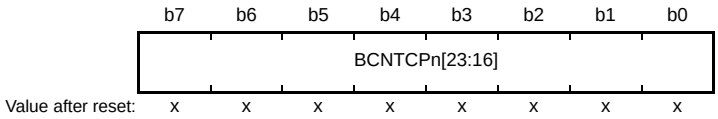
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	HR1[3:0]	1-Hour Capture	Capture value for the ones place of hours	R
b5, b4	HR10[1:0]	10-Hour Capture	Capture value for the tens place of hours	R
b6	PM	PM	0: a.m. 1: p.m.	R
b7	—	Reserved	This bit is read as 0 after an RTC software reset.	R

RHRCPn is a read-only register that captures the RHRCNT value when a time capture event is detected.
The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RHRCP0, RHRCP1, and RHRCP2 registers, respectively.
The PM bit is only enabled when the RCR2.HR24 bit is 0 (in 12-hour mode).
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

(2) In binary count mode:

Address(es): RTC.BCNT2CP0 0008 C456h, RTC.BCNT2CP1 0008 C466h, RTC.BCNT2CP2 0008 C476h



x: Undefined

BCNT2CPn is a read-only register that captures the BCNT2 value when a time capture event is detected.
The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNT2CP0, BCNT2CP1, and BCNT2CP2 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

31.2.27 Date Capture Register n (RDAYCPn) (n = 0 to 2)/BCNT3 Capture Register n (BCNT3CPn) (n = 0 to 2)

(1) In calendar count mode:

Address(es): RTC.RDAYCP0 0008 C45Ah, RTC.RDAYCP1 0008 C46Ah, RTC.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]	1-Day Capture	Capture value for the ones place of days	R
b5, b4	DATE10[1:0]	10-Day Capture	Capture value for the tens place of days	R
b7, b6	—	Reserved	These bits are read as 0 after an RTC software reset.	R

RDAYCPn is a read-only register that captures the RDAYCNT value when a time capture event is detected.
The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RDAYCP0, RDAYCP1, and RDAYCP2 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

(2) In binary count mode:

Address(es): RTC.BCNT3CP0 0008 C45Ah, RTC.BCNT3CP1 0008 C46Ah, RTC.BCNT3CP2 0008 C47Ah

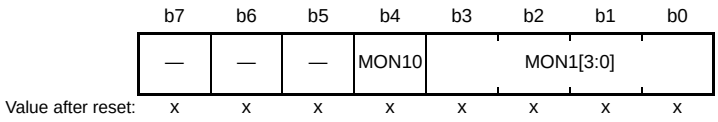
	b7	b6	b5	b4	b3	b2	b1	b0
	BCNTCPn[31:24]							
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

BCNT3CPn is a read-only register that captures the BCNT3 value when a time capture event is detected.
The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNT3CP0, BCNT3CP1, and BCNT3CP2 registers, respectively.
This register is set to 00h by an RTC software reset.
Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

31.2.28 Month Capture Register n (RMONCPn) (n = 0 to 2)

Address(es): RTC.RMONCP0 0008 C45Ch, RTC.RMONCP1 0008 C46Ch, RTC.RMONCP2 0008 C47Ch



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MON1[3:0]	1-Month Capture	Capture value for the ones place of months	R
b4	MON10	10-Month Capture	Capture value for the tens place of months	R
b7 to b5	—	Reserved	These bits are read as 0	R

RMONCPn is a read-only register that captures the RMONCNT value when a time capture event is detected. The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RMONCP0, RMONCP1, and RMONCP2 registers, respectively.

This register is set to 00h by an RTC software reset.

Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRn.TCCT[1:0] bits.

31.3 Operation

31.3.1 Outline of Initial Settings of Registers after Power On

After the power is turned on, the initial settings for the clock setting, count mode setting, time error adjustment, time setting, alarm, interrupt, and time capture control register should be performed.

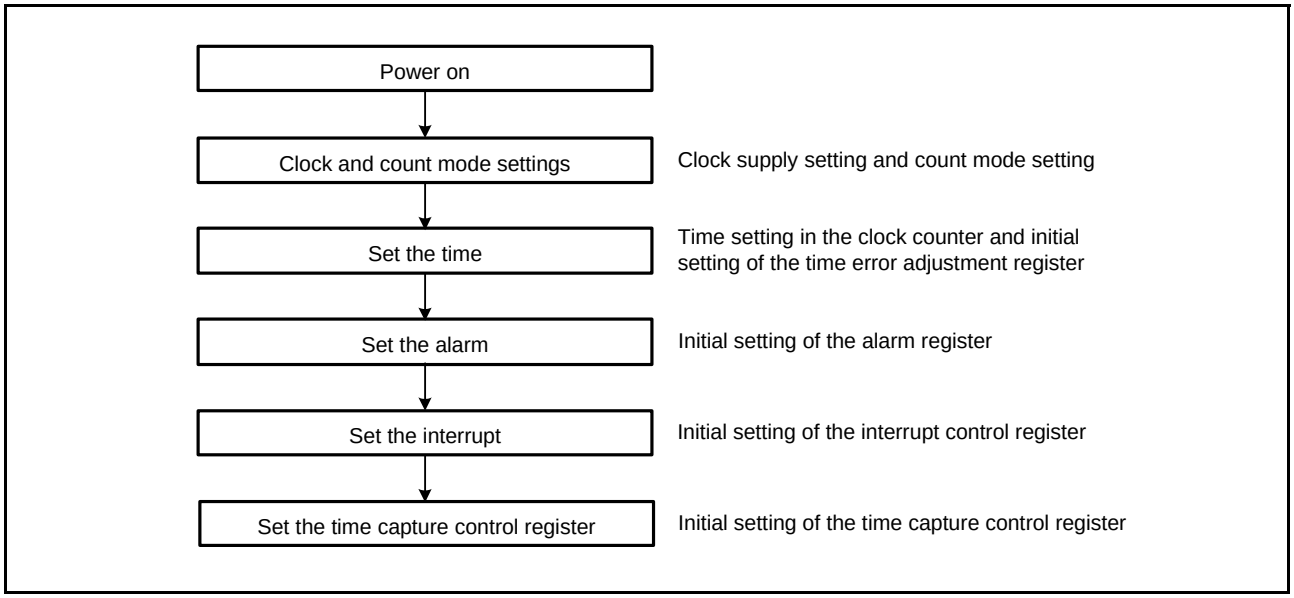
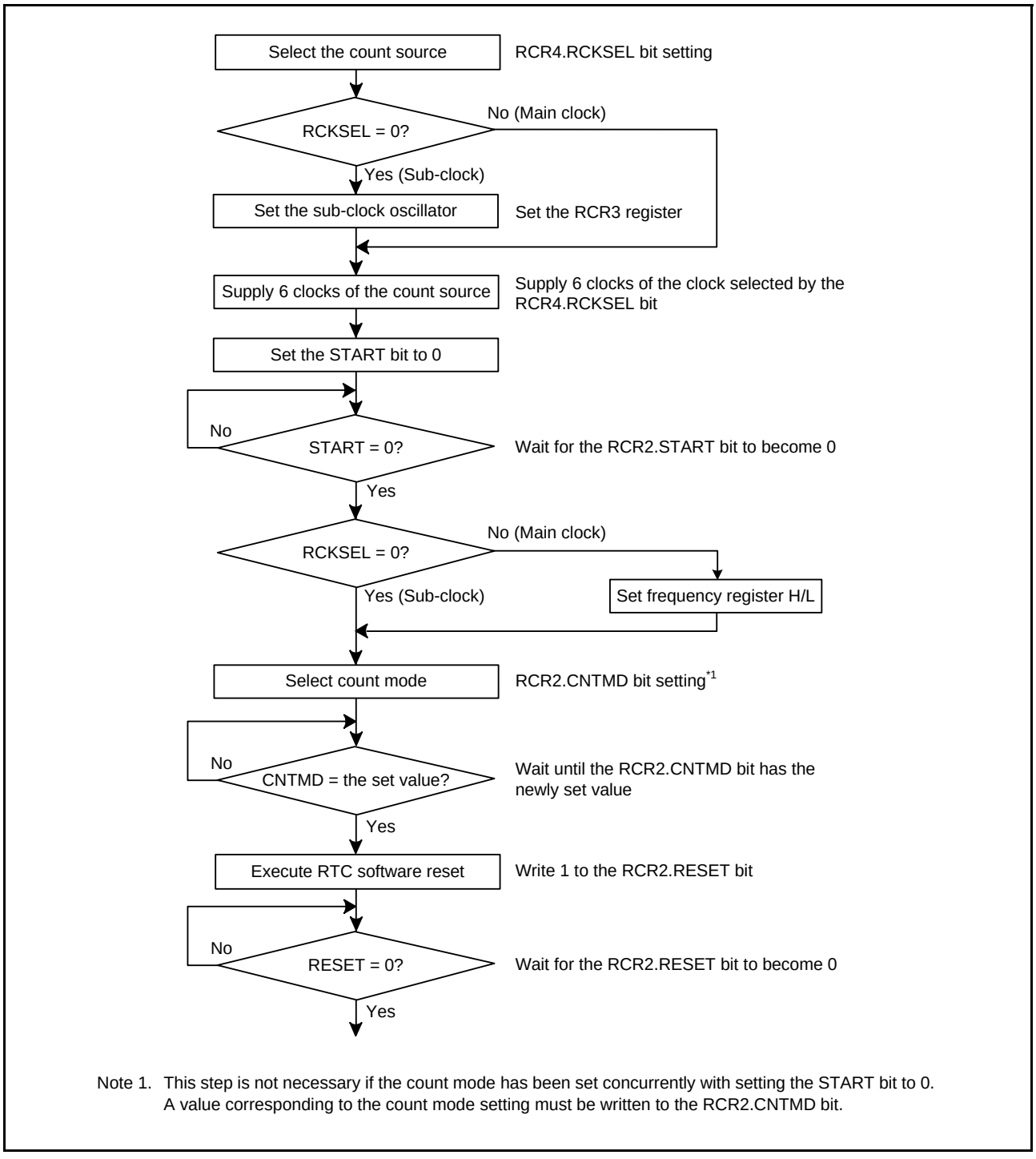


Figure 31.2 Outline of Initial Settings after Power On

31.3.2 Clock and Count Mode Setting Procedure

Figure 31.3 shows how to set the clock and the count mode.

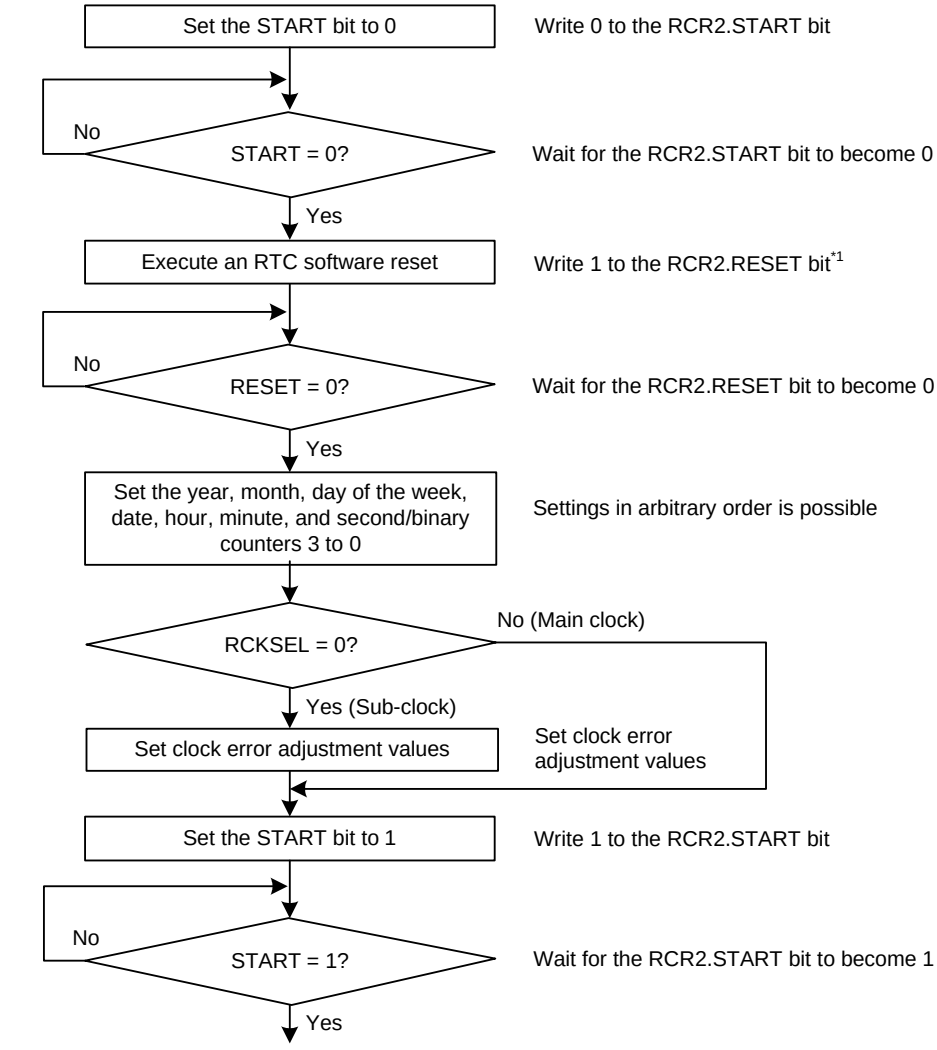


Note 1. This step is not necessary if the count mode has been set concurrently with setting the START bit to 0. A value corresponding to the count mode setting must be written to the RCR2.CNTMD bit.

Figure 31.3 Clock and Count Mode Setting Procedure

31.3.3 Setting the Time

Figure 31.4 shows how to set the time.



Note 1. This step is not necessary for the time-setting procedure because an RTC software reset is executed in the clock setting procedure of the initial settings for the power supply.

Figure 31.4 Setting the Time

31.3.4 30-Second Adjustment

Figure 31.5 shows how to execute 30-second adjustment.

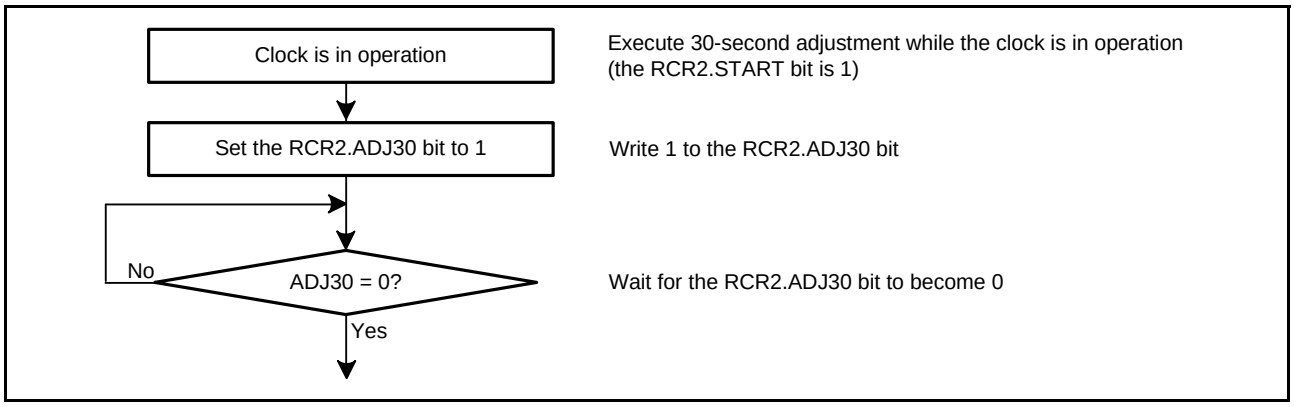


Figure 31.5 30-Second Adjustment

31.3.5 Reading 64-Hz Counter and Time

Figure 31.6 shows how to read the 64-Hz counter and time.

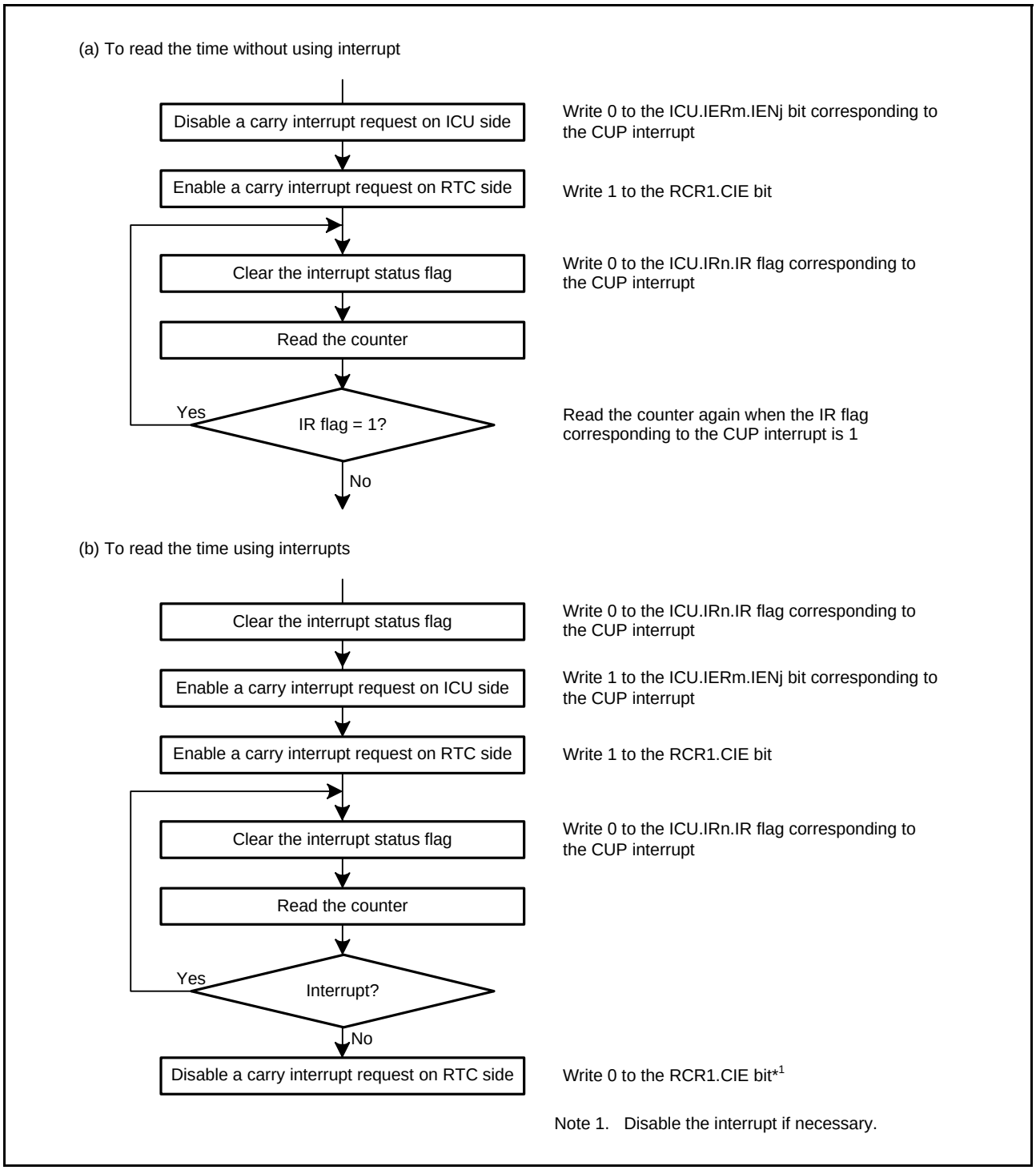


Figure 31.6 Reading Time

If a carry occurs while the 64-Hz counter and time are being read, the correct time will not be obtained, so they must be read again. The procedure for reading the time without using interrupts is shown in (a) in Figure 31.6, and the procedure using carry interrupts in (b). To keep the program simple, method (a) should be used in most cases.

31.3.6 Alarm Function

Figure 31.7 shows how to use the alarm function.

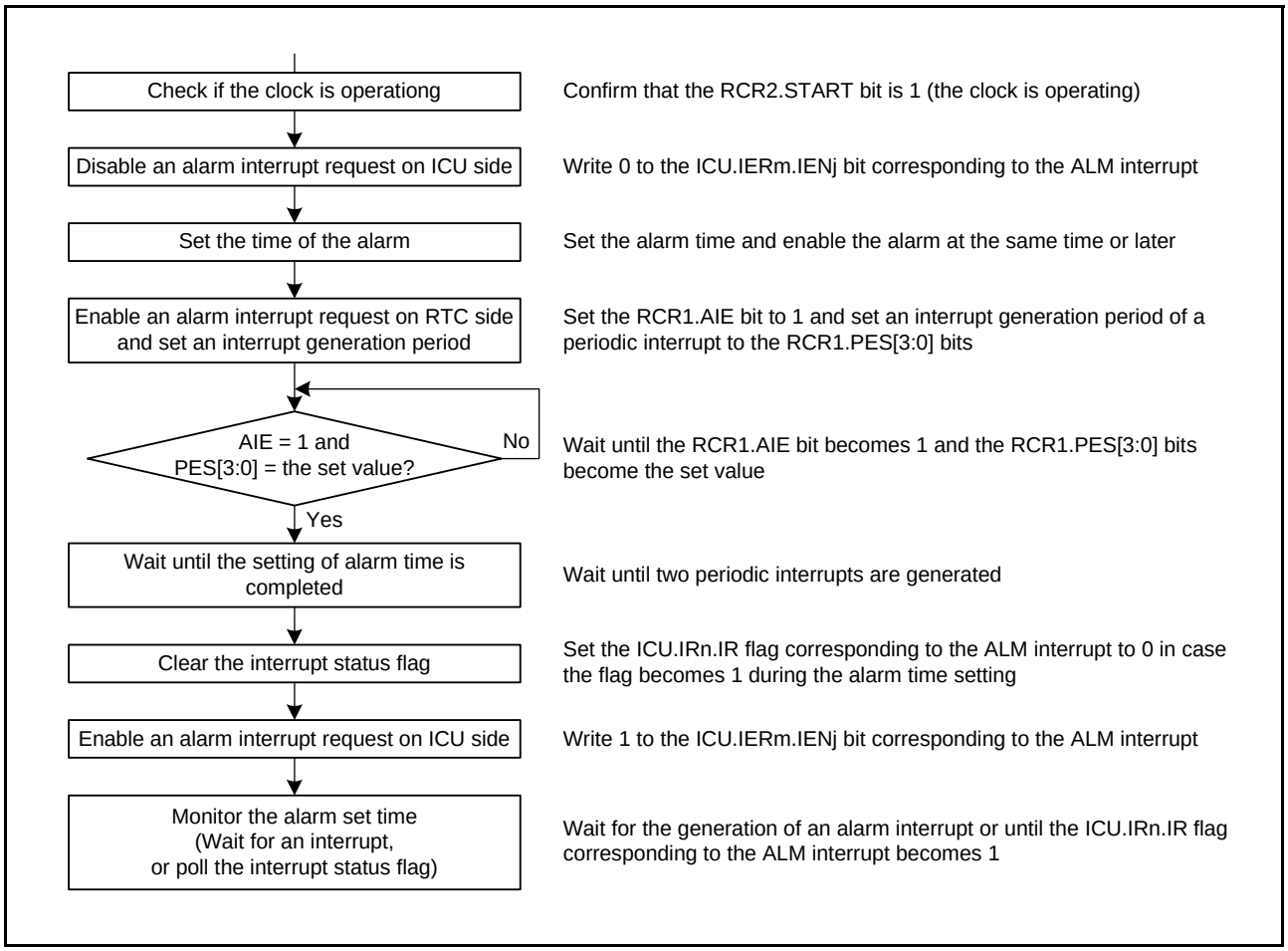


Figure 31.7 Using Alarm Function

In calendar count mode, an alarm can be generated by any one of year, month, date, day-of-week, hour, minute or second, or any combination of those. Write 1 to the ENB bit in the alarm registers involved in the alarm setting, and set the alarm time in the lower bits. Write 0 to the ENB bit in registers not involved in the alarm setting.

In binary count mode, an alarm can be generated in any bit combination of 32 bits. Write 1 to the ENB bit of the alarm enable register corresponding to the target bit of the alarm, and set the alarm time to the alarm register. For bits that are not target of the alarm, write 0 to the ENB bit of the alarm enable register.

When the counter and the alarm time match, the IR flag corresponding to the ALM interrupt is set to 1. Alarm detection can be confirmed by reading this bit, but an interrupt should be used in most cases. If 1 has been set in the interrupt request enable bit corresponding to the ALM interrupt, an alarm interrupt is generated in the event of alarm, enabling the alarm to be detected.

Writing 0 sets the IR flag corresponding to the ALM interrupt to 0.

When the counter and the alarm time match in a low power consumption state, the MCU returns from the low power consumption state. In deep software standby mode, the MCU returns from the deep software standby mode even when the alarm interrupt request is disabled.

31.3.7 Procedure for Disabling Alarm Interrupt

Figure 31.8 shows the procedure for disabling the enabled alarm interrupt request.

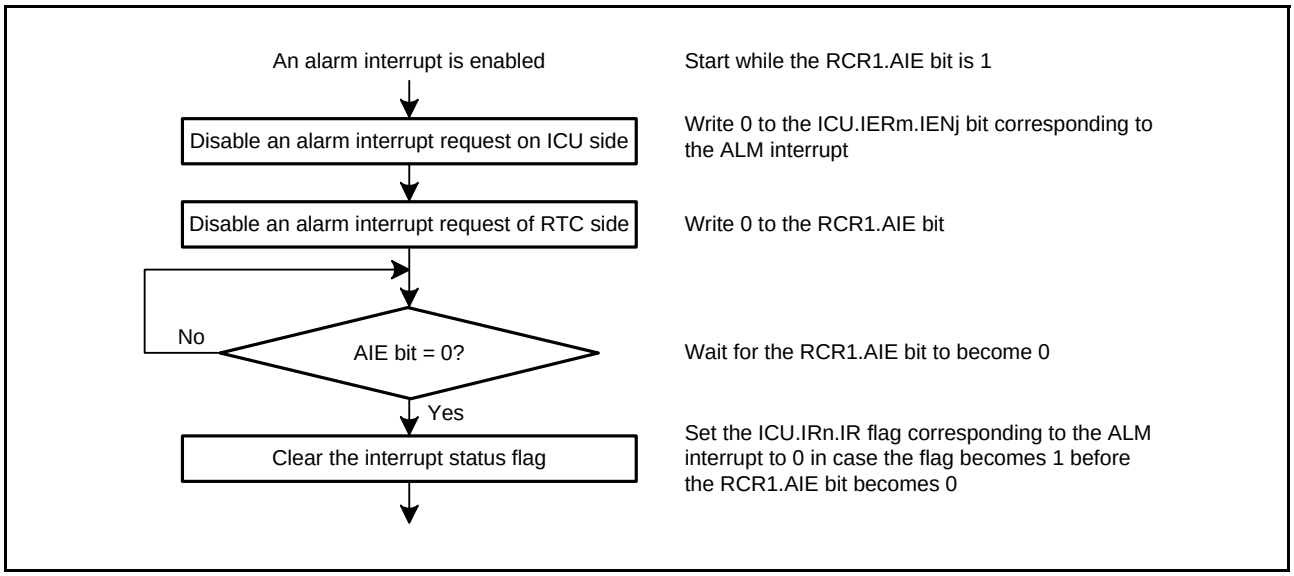


Figure 31.8 Procedure for Disabling Alarm Interrupt Request

31.3.8 Time Error Adjustment Function

The time error adjustment function is used to correct errors (running fast or slow) in the time due to the precision of oscillation by the sub-clock. Since 32,768 cycles of the sub-clock constitute 1 second of operation when the sub-clock is selected, the clock runs fast if the sub-clock frequency is high and slow if the sub-clock frequency is low. This function can be used to correct errors due to the clock running fast or slow.
Two types of time error adjustment functions are provided: automatic adjustment and adjustment by software.
Use the RCR2.AADJE bit to select automatic adjustment or adjustment by software.

31.3.8.1 Automatic Adjustment

Enable automatic adjustment by setting the RCR2.AADJE bit to 1.
Automatic adjustment is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register every time the adjustment period selected by the RCR2.AADJE bit elapses.
Examples are shown below.

[Example 1] Sub-clock running at 32.769 kHz
Adjustment procedure:
When the sub-clock is running at 32.769 kHz, 1 second elapses every 32,769 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs fast by one clock cycle every second. The time on the clock is fast by 60 clock cycles per minute, so adjustment can take the form of setting the clock back by 60 cycles every minute.

Register settings: (when RCR2.CNTMD = 0)

- RCR2.AADJP = 0 (adjustment every minute)
- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler.)
- RADJ.ADJ[5:0] = 60 (3Ch)

[Example 2] Sub-clock running at 32.766 kHz
Adjustment procedure:
When the sub-clock is running at 32.766 kHz, 1 second elapses every 32,766 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs slow by two clock cycles every second. The time on the clock is slow by 20 clock cycles every 10 seconds, so adjustment can take the form of setting the clock forward by 20 cycles every 10 seconds.

Register settings: (when RCR2.CNTMD = 0)

- RCR2.AADJP = 1 (adjustment every 10 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler.)
- RADJ.ADJ[5:0] = 20 (14h)

[Example 3] Sub-clock running at 32.764 kHz
Adjustment procedure:
At 32.764 kHz, 1 second elapses on 32,764 clock cycles. Since the RTC operates for 32,768 clock cycles as 1 second, the clock is delayed for four clock cycles per second. In 8 seconds, the delay is 32 clock cycles, so correction can be made by proceeding the clock for 32 clock cycles every 8 seconds.

Register settings when the RCR2.CNTMD bit is 1

- RCR2.AADJP = 1 (adjustment every 8 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler.)
- RADJ.ADJ[5:0] = 32 (20h)

31.3.8.2 Adjustment by Software

Enable adjustment by software by setting the RCR2.AADJE bit to 0.
Adjustment by software is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register at the time of execution of an instruction for writing to the RADJ register.
An example is shown below.

[Example 1] Sub-clock running at 32.769 kHz
Adjustment procedure:
When the sub-clock is running at 32.769 kHz, 1 second elapses every 32,769 clock cycles. The RTC is meant to run at 32,768 clock cycles, so the clock runs fast by one clock cycle every second. The time on the clock is fast by one clock cycle per second, so adjustment can take the form of setting the clock back by one cycle every second.

Register settings:

- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler.)
- RADJ.ADJ[5:0] = 1 (01h)
  This is written to the RADJ register once per 1-second interrupt.

31.3.8.3 Procedure for Changing the Mode of Adjustment

When changing the mode of adjustment, change the value of the AADJE bit in RCR2 after setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

Changing from adjustment by software to automatic adjustment:

(1) Set the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
(2) Set the RCR2.AADJE bit to 1 (automatic adjustment is enabled).
(3) Use the RCR2.AADJP bit to select the period of adjustment.
(4) In RADJ, set the PMADJ[1:0] bits for addition or subtraction and the ADJ[5:0] bits to the value for use in time error adjustment.

Changing from adjustment by software to automatic adjustment:

(1) Set the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
(2) Set the RCR2.AADJE bit to 0 (adjustment by software is enabled).
(3) Proceed with adjustment by setting the RADJ.PMADJ[1:0] bits for addition or subtraction and the RADJ.ADJ[5:0] bits to the value for use in time error adjustment at the desired time. After that, the time is adjusted every time a value is written to the RADJ register.

31.3.8.4 Procedure for Stopping Adjustment

Stop adjustment by setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

31.3.9 Time Capture Function

The RTC is capable of storing the month, date, hour, minute and second/binary counters 3 to 0 by detecting an edge of a signal on a time capture event input pin.

A noise filter can also be used on a time capture event input pin. If the noise filter is enabled, the TCST bit is set to 1 when the input level on the pin matches three times.

The noise filter can be switched on or off for each of the time capture event input pins. Operation when the noise filter is off is shown in Figure 31.9 and operation when the noise filter is on is shown in Figure 31.10.

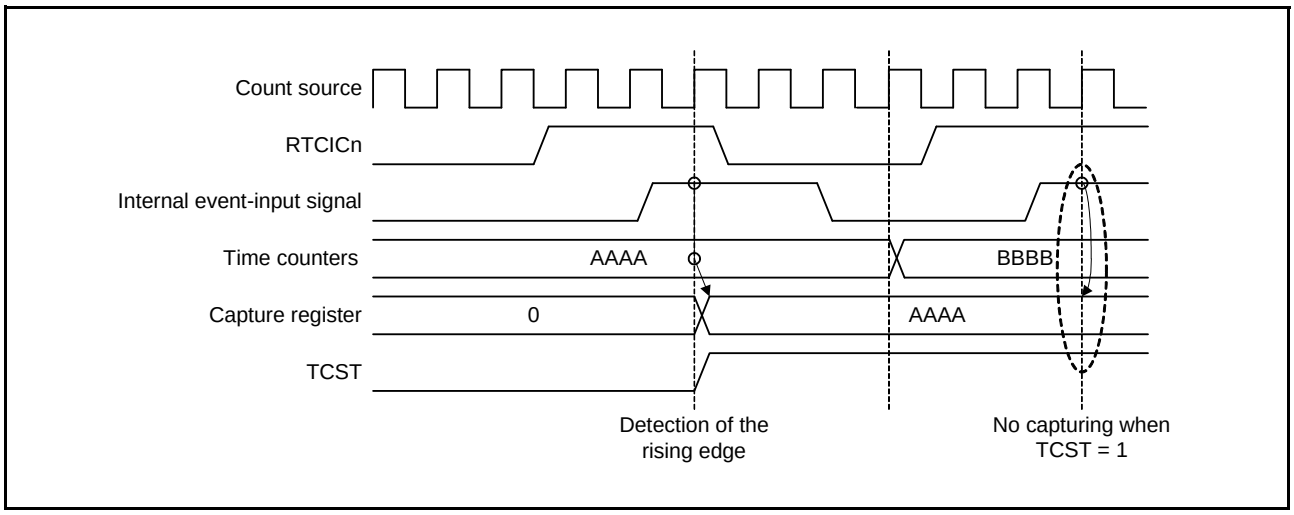


Figure 31.9 Timing of a Time Capture Operation (with the Filter Off) (n = 0 to 2)

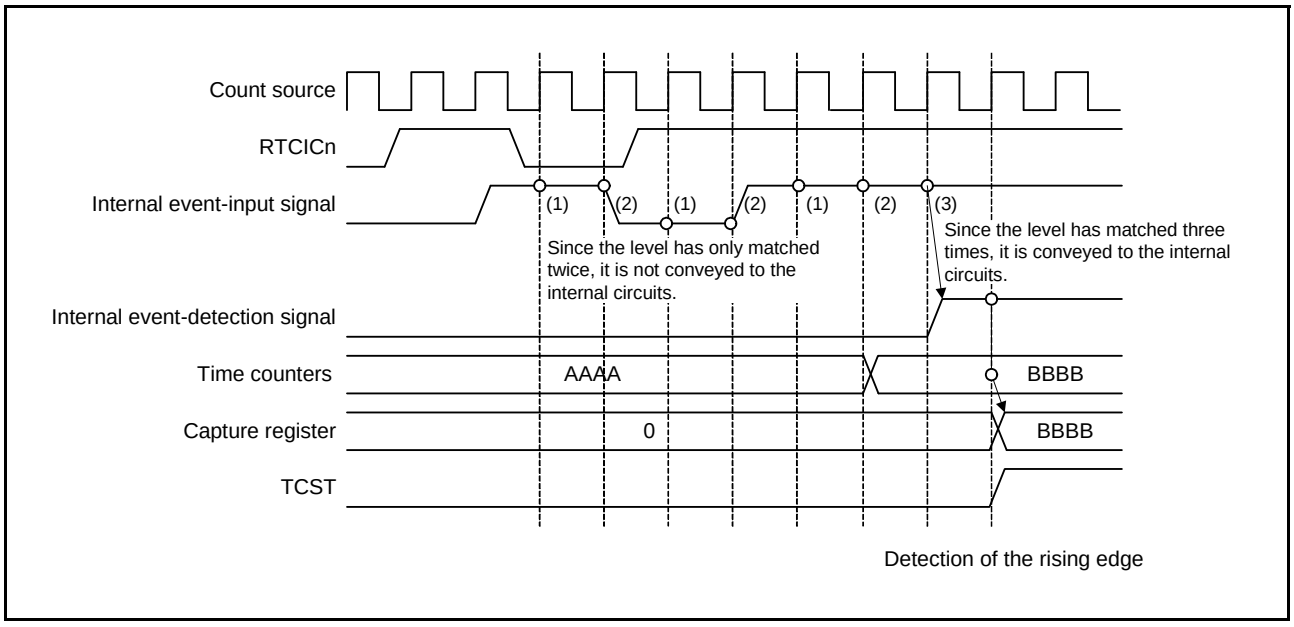


Figure 31.10 Timing of a Time Capture Operation (with the Filter On) (n = 0 to 2)

31.4 Interrupt Sources

There are three interrupt sources in the realtime clock. Table 31.4 lists interrupt sources for the RTC.

Table 31.4 RTC Interrupt Sources

Name	Interrupt Sources
ALM	Alarm interrupt
PRD	Periodic interrupt
CUP	Carry interrupt

(1) Alarm interrupt (ALM)

This interrupt is generated according to the result of comparison between the alarm registers and realtime clock counters (for details, refer to section 31.3.6, Alarm Function).

Since there is a possibility that the interrupt flag may be set to 1 when the settings of the alarm registers match the clock counters, wait for the alarm time settings to be confirmed and set the IR flag corresponding to the ALM interrupt to 0 again after modifying values of the alarm registers. Once the interrupt flag for the alarm interrupt has been set to 1 and the state has returned to non-matching of the alarm registers and clock counters, the flag will not be set again until there is a further match or the values of the alarm registers are modified again.

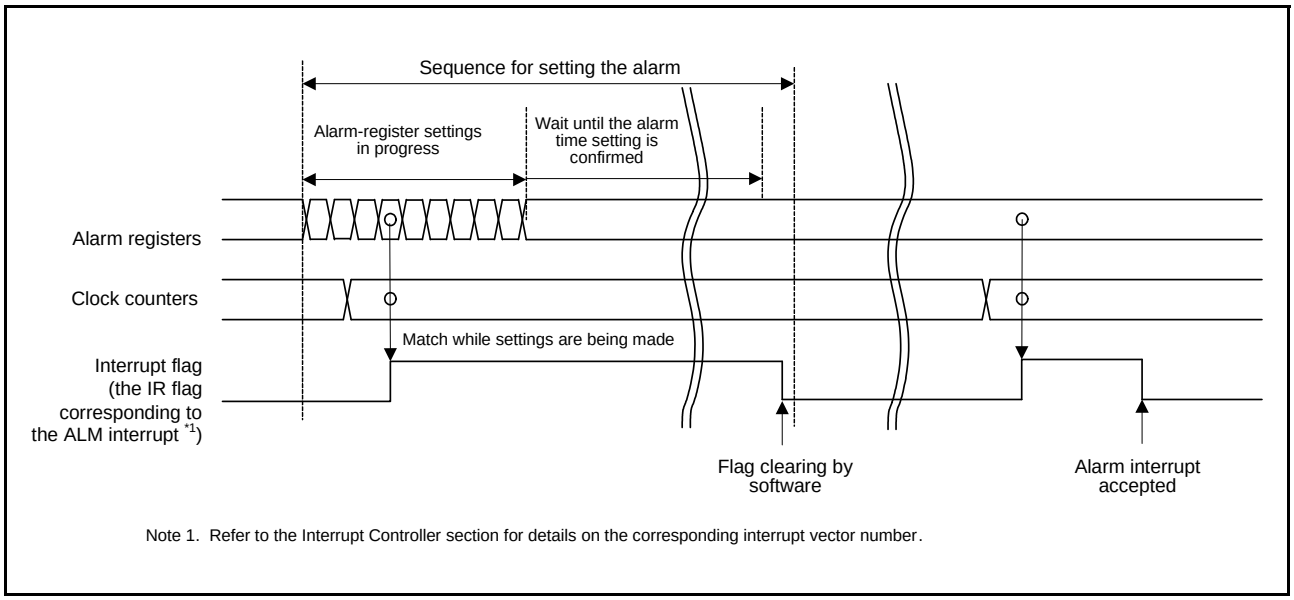


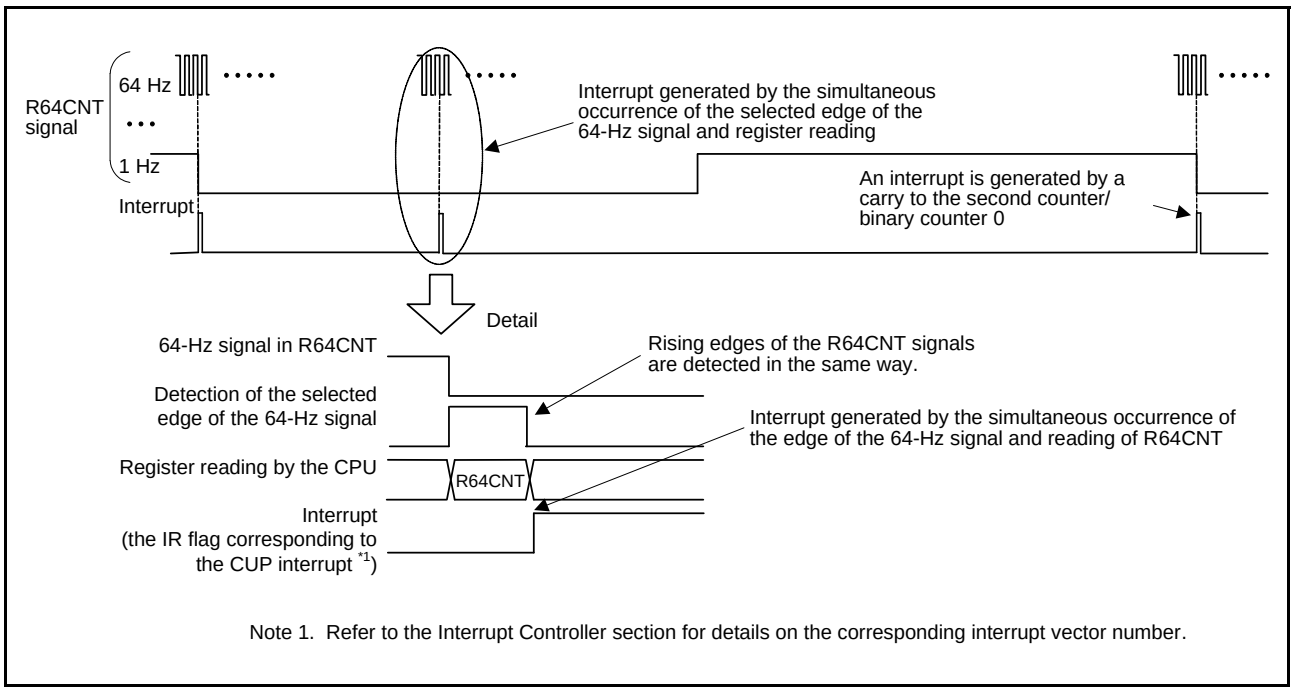
Figure 31.11 Timing Chart for the Alarm Interrupt (ALM)

(2) Periodic interrupt (PRD)

This interrupt is generated at intervals of 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, 1/128 second, or 1/256 second. The interrupt interval can be selected through the RCR1.PES[3:0] bits.

(3) Carry interrupt (CUP)

This interrupt is generated when a carry to the second counter/binary counter 0 occurred or a carry to the R64CNT counter occurred during read access to the 64-Hz counter.



Note 1. Refer to the Interrupt Controller section for details on the corresponding interrupt vector number.

Figure 31.12 Carry Interrupt (CUP) Timing Chart

31.5 Event Link Output

The RTC outputs the following event signals for the event link controller (ELC), and these can be used to initiate operations by other modules selected in advance.

(1) Periodic event output

The periodic event signal is output at the interval selected from among 1/256, 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1, and 2 seconds by the setting of the RCR1.PES[3:0] bits.

The event generation period immediately after the event generation is selected is not guaranteed.

Note: If event linking from the RTC is to be used, only make the ELC settings after making the RTC settings (initialization, time settings, etc.). Making the RTC settings after the ELC settings can lead to the output of unexpected event signals.

31.5.1 Interrupt Handling and Event Linking

The RTC has a bit to enable or disable periodic interrupts. An interrupt request signal is output for the CPU when an interrupt source is generated while the corresponding enable bit is enabled.

In contrast, an event link output signal is sent to other modules as an event signal via the ELC when an interrupt source is generated, regardless of the setting of the corresponding interrupt enable bit.

Note: Although alarm and periodic interrupts can still be output during software standby or deep software standby, the periodic event signals for the ELC are not output.

31.6 Usage Notes

31.6.1 Register Writing during Counting

The following registers should not be written to during counting (while the RCR2.START bit = 1).

RSECCNT/BCNT0, RMINCNT/BCNT1, RHRCNT/BCNT2, RDAYCNT, RWKCNT/BCNT3, RMONCNT, RYRCNT, RCR1.RTCOS, RCR2.RTCOE, RCR2.HR24, RFRH, RFRL

The counter must be stopped before writing to any of the above registers.

31.6.2 Use of Periodic Interrupts

The procedure for using periodic interrupts is shown in Figure 31.13.
The generation and period of the periodic interrupt can be changed by the setting of the RCR1.PES[3:0] bits. However, since the prescaler, R64CNT, and RSECCNT/BCNT0 are used to generate interrupts, the interrupt period is not guaranteed immediately after setting of the RCR1.PES[3:0] bits.
Furthermore, stopping/restarting or resetting counter operation, reset by RTC software, and the 30-second adjustment by changing the RCR2 value affects the interrupt period. When the time error adjustment function is used, the interrupt generation period after adjustment is added or subtracted according to the adjustment value.

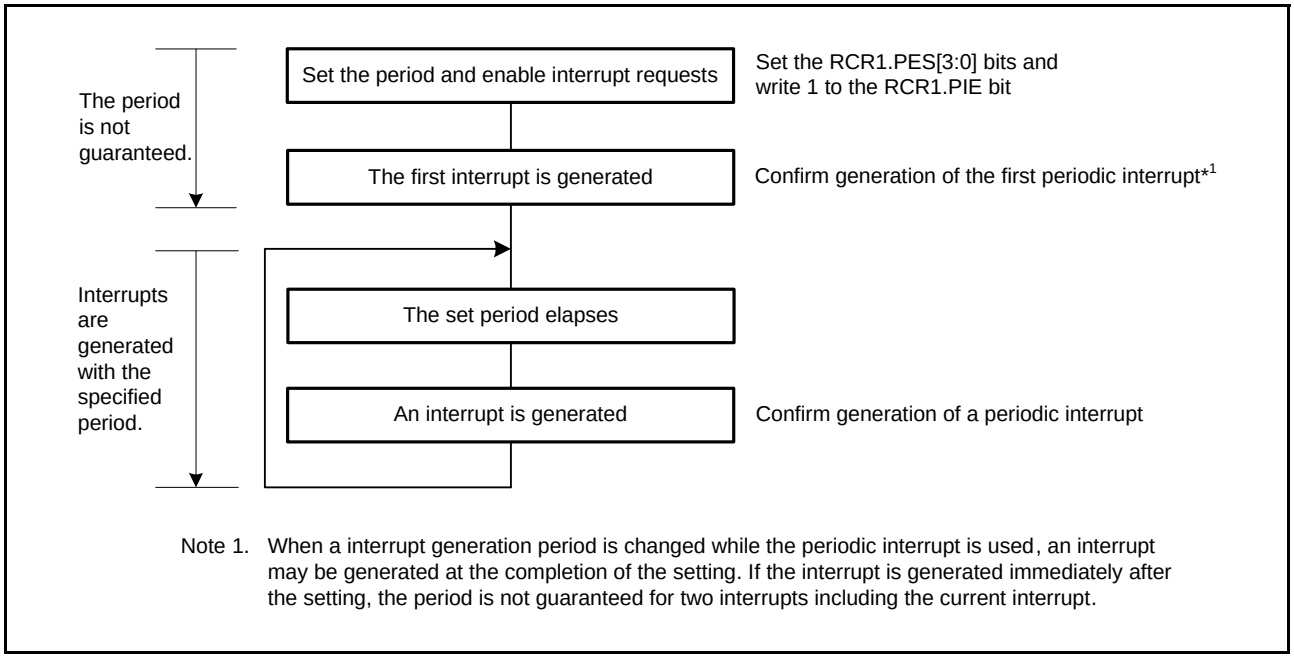


Figure 31.13 Using Periodic Interrupt Function

31.6.3 RTCOUT (1-Hz/64-Hz) Clock Output

Stopping/restarting or resetting counter operation, reset by RTC software, and the 30-second adjustment by changing the RCR2 value affects the period of RTCOUT (1-Hz/64-Hz) output. When the time error adjustment function is used, the period of RTCOUT (1-Hz/64-Hz) output after adjustment is added or subtracted according to the adjustment value.

31.6.4 Transitions to Low Power Consumption Modes after Setting Registers

A transition to a low power consumption state (software standby mode, deep software standby mode, or battery backup) during writing to or updating of an RTC register might destroy the register's value. After setting a register, confirm that the setting is in place before initiating a transition to a low power consumption state.

31.6.5 Notes on 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 31.3.5, Reading 64-Hz Counter and Time.
- The value written to the count registers, alarm registers, year alarm enable register, bits RCR2.AADJE, AADJP, and HR24, RCR3, RCR4, RFRH, or RFRL register is reflected when four read operations are performed after writing.
- The values written to the RCR1.CIE, RTCOS, and RCR2.RTCOE bits can be read immediately after writing.
- To read the value from the timer counter after return from a reset, deep software standby mode, software standby mode, or the battery backup state, wait for 1/128 second while the clock is operating (RCR2.START bit = 1).
- After a reset is generated, write to the RTC register when six cycles of the count source have elapsed.

31.6.6 Changing the Count Mode

When changing the count mode (calendar/binary), set the RCR2.START bit to 0, stop counting operation, then start again from the initial setting. For details on initial setting, refer to section 31.3.1, Outline of Initial Settings of Registers after Power On.

31.6.7 Notes on Using the Time Capture Function

When the time capture function is to be used, set both the RTCCRn.TCEN and TAMPICR1.CHnEN bits (n = 0 to 2) to 1. When tampering is not to be detected, leave the TAMPCR, TCECR, TAMPICR2, and TAMPIMR registers in the tamper detector at their initial values.
When the time capture function is to be used to obtain the time stamps in response to the detection of tampering, use the noise filter in the tamper detector and set the RTCCRn.TCNF[1:0] bits to 00b (switch the noise filter of the RTC off).

31.6.8 Initialization Procedure When the Realtime Clock is Not to be Used

Registers in the RTC are not initialized by a reset. Accordingly, depending on the initial state, the generation of an unintentional interrupt request or operation of the counter may lead to increased power consumption.

For products that do not require a realtime clock, initialize the registers by following the initialization procedure shown in Figure 31.14.

Alternatively, when the sub-clock is not used as the system clock or realtime clock, the counter can be stopped by writing 0 (sub-clock oscillator is selected) to the RCR4.RCKSEL bit and stopping the sub-clock.

When making the setting to stop the sub-clock, write 0 to the RCR3.RTCEN bit and write 1 to the SOSCCR.SOSTP bit. For details on the setting of the SOSCCR.SOSTP bit, refer to section 9, Clock Generation Circuit.

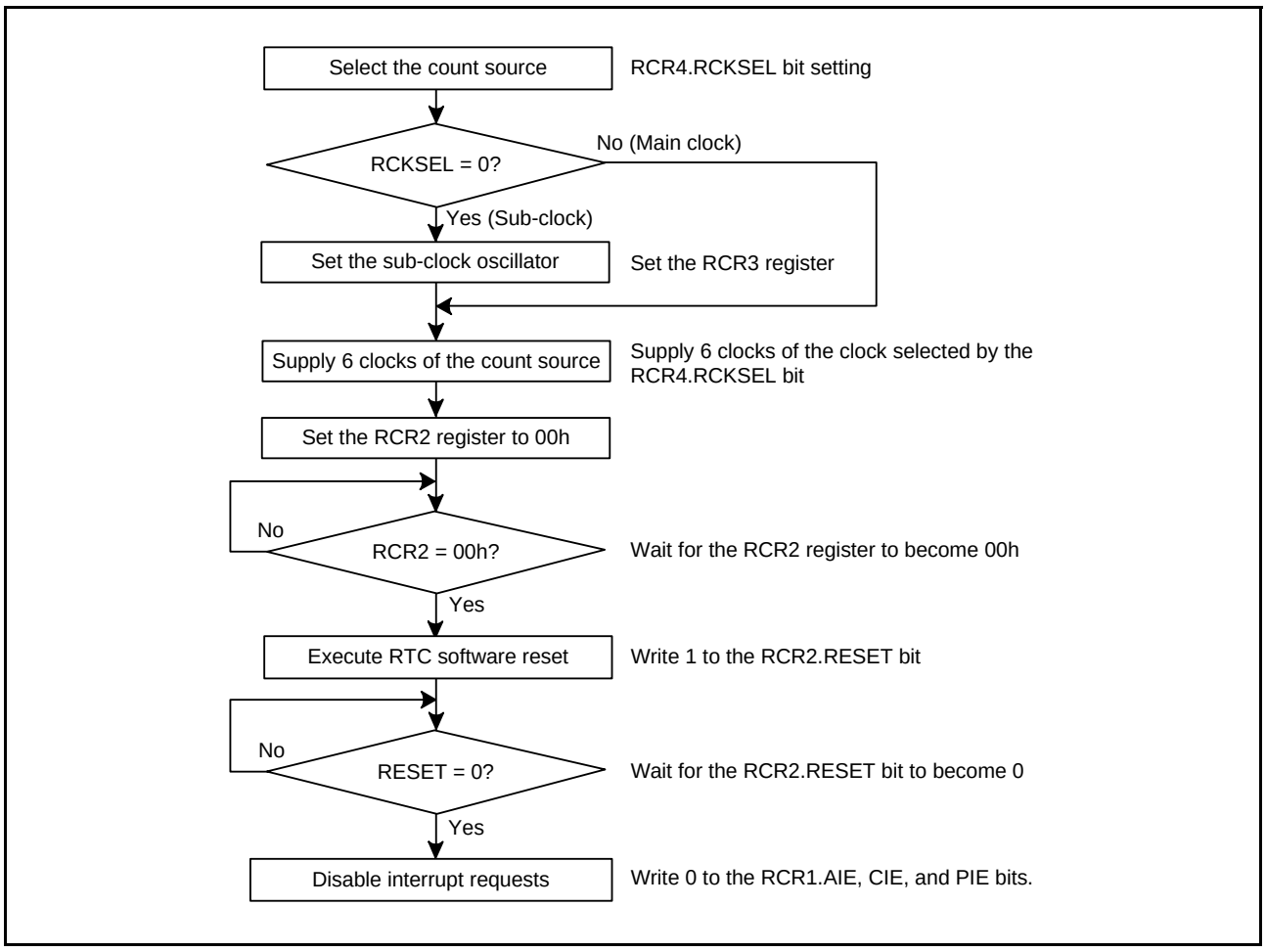


Figure 31.14 Initialization Procedure

32. Watchdog Timer (WDTA)

The watchdog timer (WDT) is a 14-bit down-counter. It can be used to reset this MCU when the counter underflows because its value cannot be refreshed due to the system being out of control.
In addition, a non-maskable interrupt can be generated by an underflow.
The refresh-permitted period can be set to refresh the counter and used as the condition to detect when the system runs out of control.
In this section, “PCLK” is used to refer to PCLKB.

32.1 Overview

Table 32.1 lists the specifications of the WDT and Figure 32.1 shows a block diagram of the WDT.

Table 32.1 WDT Specifications

Item	Specifications
Count source	Peripheral module clock (PCLK)
Clock division ratio	Divide by 4, 64, 128, 512, 2048, or 8192
Counter operation	Counting down using a 14-bit down-counter
Conditions for starting the counter	• Auto-start mode: Counting automatically starts after a reset is released • Register start mode: Counting is started by refresh operation (writing 00h and then FFh to the WDTRR register)
Conditions for stopping the counter	• Reset (the down-counter and other registers return to their initial values) • In low power consumption states • A counter underflows or a refresh error occurs (only in register start mode)
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Watchdog timer Reset sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Non-maskable interrupt/interrupt sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Reading the counter value	The down-counter value can be read by the WDTSR register.

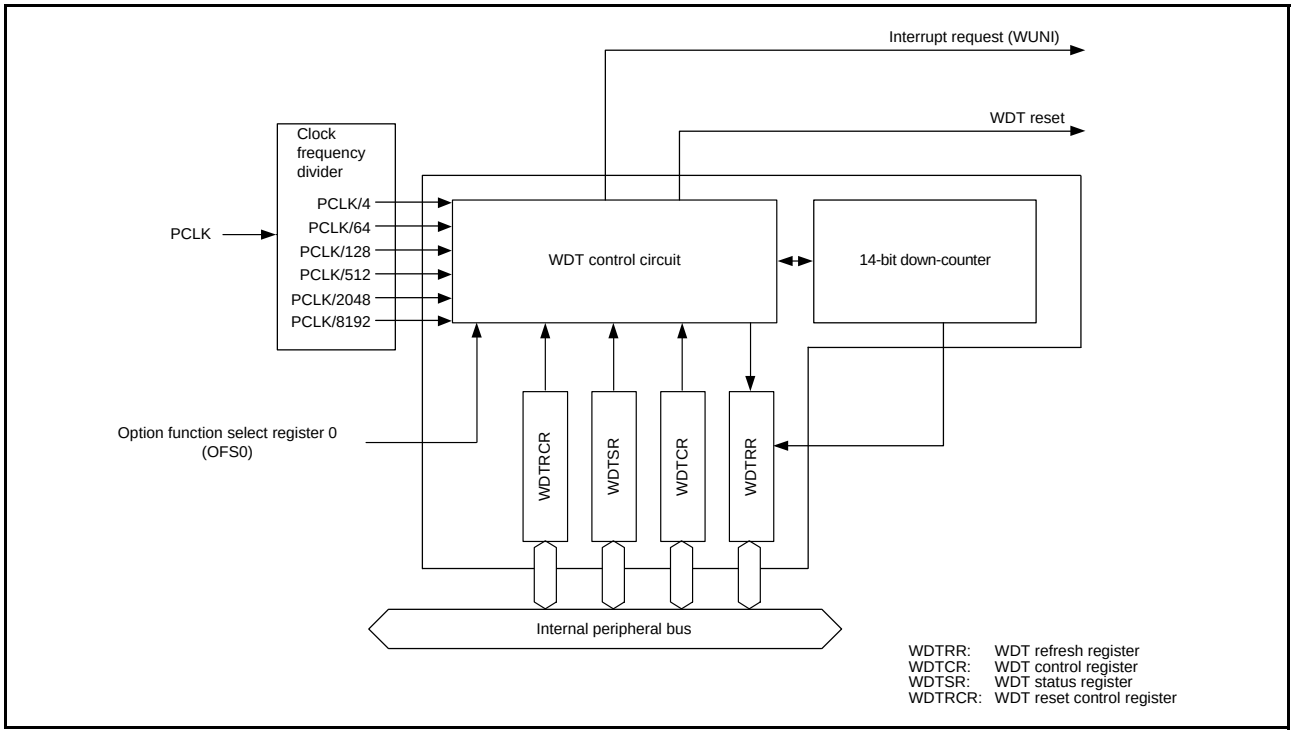
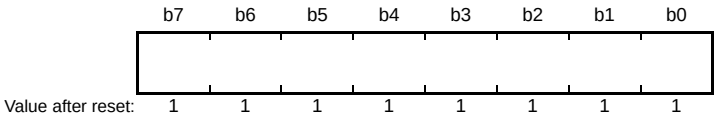


Figure 32.1 WDT Block Diagram

32.2 Register Descriptions

32.2.1 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 WDTTOPS[1:0] bits in option function select register 0 (OFS0) in auto-start mode. In register start mode, counting down starts from the value selected by setting the WDTCR.TOPS[1:0] bits.
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 32.3.3, Refresh Operation.

32.2.2 WDT Control Register (WDTCR)

Address(es): 0008 8022h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TOPS[1:0]	Timeout Period Selection	b1 b0 0 0: 1024 cycles (03FFh) 0 1: 4096 cycles (0FFFh) 1 0: 8192 cycles (1FFFh) 1 1: 16384 cycles (3FFFh)	R/W
b3, b2	—	Reserved	These bits are read as 0 and cannot be modified.	R
b7 to b4	CKS[3:0]	Clock Division Ratio Selection	b7 b4 0 0 0 1: Divide-by-4 0 1 0 0: Divide-by-64 1 1 1 1: Divide-by-128 0 1 1 0: Divide-by-512 0 1 1 1: Divide-by-2048 1 0 0 0: Divide-by-8192 Setting other than above are prohibited.	R/W
b9, b8	RPES[1:0]	Window End Position Selection	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (window end position is not specified)	R/W
b11, b10	—	Reserved	These bits are read as 0 and cannot be modified.	R
b13, b12	RPSS[1:0]	Window Start Position Selection	b13 b12 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (window start position is not specified)	R/W
b15, b14	—	Reserved	These bits are read as 0 and cannot be modified.	R

There are some restrictions on writing to the WDTCR register. For details, refer to section 32.3.2, Control over Writing to the WDTCR and WDTRCR Registers.

In auto-start mode, the settings in the WDTCR register are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting made to the WDTCR register can also be made in OFS0 register. For details, refer to section 32.3.7, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

TOPS[1:0] Bits (Timeout Period Selection)

These bits select the timeout period (period until the down-counter underflows) from among 1024, 4096, 8192, and 16384 cycles, taking the divided clock specified by the CKS[3:0] bits as one cycle.

After the down-counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of PCLK cycles) until the counter underflows.

Relations between the CKS[3:0] and TOPS[1:0] bit settings, the timeout period, and the number of PCLK cycles are listed in Table 32.2.

Table 32.2 Timeout Period Settings

CKS[3:0] Bits				TOPS[1:0] Bits		Clock Division Ratio	Timeout Period (Number of Cycles)	Cycles of PCLK Clock
b7	b6	b5	b4	b1	b0			
0	0	0	1	0	0	Divide-by-4	1024	4096
				0	1		4096	16384
				1	0		8192	32768
				1	1		16384	65536
0	1	0	0	0	0	Divide-by-64	1024	65536
				0	1		4096	262144
				1	0		8192	524288
				1	1		16384	1048576
1	1	1	1	0	0	Divide-by-128	1024	131072
				0	1		4096	524288
				1	0		8192	1048576
				1	1		16384	2097152
0	1	1	0	0	0	Divide-by-512	1024	524288
				0	1		4096	2097152
				1	0		8192	4194304
				1	1		16384	8388608
0	1	1	1	0	0	Divide-by-2048	1024	2097152
				0	1		4096	8388608
				1	0		8192	16777216
				1	1		16384	33554432
1	0	0	0	0	0	Divide-by-8192	1024	8388608
				0	1		4096	33554432
				1	0		8192	67108864
				1	1		16384	134217728

CKS[3:0] Bits (Clock Division Ratio Selection)

These bits specify the division ration of the clock used for the down-counter. The division ration can be selected from among the peripheral module clock (PCLK) divided by 4, 64, 128, 512, 2048, and 8192. Combined with the TOPS[1:0] bit setting, a count period between 4,096 and 134,217,728 cycles of the PCLK clock can be selected for the WDT.

RPES[1:0] Bits (Window End Position Selection)

These bits specify the window end position that indicates the refresh-permitted period. 75%, 50%, 25%, or 0% of the timeout period can be selected for the window end position. The selected window end position should be a value smaller than the window start position (window start position > window end position). If the window end position is greater than the window start position, only the window start position setting is enabled.

RPSS[1:0] Bits (Window Start Position Selection)

These bits specify the window start position that indicates the refresh-permitted period. 25%, 50%, 75%, or 100% of the timeout period can be selected for the window end position. The window start position should be set to a value greater the window end position. If the window start position is set to a value smaller than or equal to the window end position, the window end position is set to 0%.

Table 32.3 lists the counter values for the window start and end positions and Figure 32.2 shows the refresh-permitted period set by the RPSS[1:0], RPES[1:0], and TOPS[1:0] bits.

Table 32.3 Relationship between Timeout Period and Window Start and End Counter Values

TOPS[1:0] Bits		Timeout Period		Window Start and End Counter Value			
		Cycles	Counter Value	100%	75%	50%	25%
0	0	1024	03FFh	03FFh	02FFh	01FFh	00FFh
0	1	4096	0FFFh	0FFFh	0BFFh	07FFh	03FFh
1	0	8192	1FFFh	1FFFh	17FFh	0FFFh	07FFh
1	1	16384	3FFFh	3FFFh	2FFFh	1FFFh	0FFFh

RPSS[1:0] Bits		RPES[1:0] Bits		Window	
b13	b12	b9	b8	Start (%)	End (%)
1	1	1	1	100	0
		1	0		25
		0	1		50
		0	0		75
1	0	1	1	75	0
		1	0		25
		0	1		50
		0	0		75
0	1	1	1	50	0
		1	0		25
		0	1		50
		0	0		75
0	0	1	1	25	0
		1	0		25
		0	1		50
		0	0		75

Note: If window end setting ≥ window start setting, the window end setting is set to 0%.

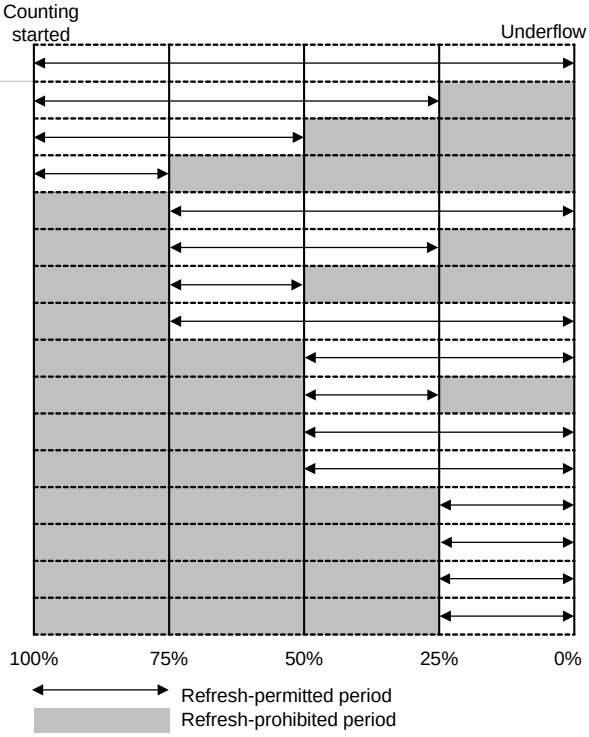


Figure 32.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period

32.2.3 WDT Status Register (WDTSR)

Address(es): 0008 8024h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REFEF	UNDFF	CNTVAL[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b13 to b0	CNTVAL[13:0]	Down-Counter Value	Value counted by the down-counter	R
b14	UNDFF	Underflow Flag	0: No underflow occurred 1: Underflow occurred	R(/W) *1
b15	REFEF	Refresh Error Flag	0: No refresh error occurred 1: Refresh error occurred	R(/W) *1

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

CNTVAL[13:0] Bits (Down-Counter Value)

Read these bits to confirm the value of the down-counter, but note that the read value may differ from the actual count by a value of one count.

UNDFF Flag (Underflow Flag)

Read this flag to confirm whether or not an underflow has occurred in the down-counter.
The value 1 indicates that the down-counter has underflowed. The value 0 indicates that the down-counter has not underflowed.
Write 0 to the UNDFF flag to set the value to 0. Writing 1 has no effect.

REFEF Flag (Refresh Error Flag)

Read this flag to confirm whether or not a refresh error (performing a refresh operation during a refresh-prohibited period) has occurred.
The value 1 indicates that a refresh error has occurred. The value 0 indicates that no refresh error has occurred.
Write 0 to the REFEF flag to set the value to 0. Writing 1 has no effect.

32.2.4 WDT Reset Control Register (WDTRCR)

Address(es): 0008 8026h

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

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0 and cannot be modified.	R
b7	RSTIRQS	Reset Interrupt Request Selection	0: Non-maskable interrupt request or interrupt request output is enabled 1: Reset output is enabled	R/W

There are some restrictions on writing to the WDTRCR register. For details, refer to section 32.3.2, Control over Writing to the WDTCR and WDTRCR Registers.

In auto-start mode, the WDTRCR register settings are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting made to the WDTCR register can also be made in the OFS0 register. For details, refer to section 32.3.7, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

32.2.5 Option Function Select Register 0 (OFS0)

For details on the OFS0 register, refer to section 32.3.7, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

32.3 Operation

32.3.1 Count Operation in Each Start Mode

The WDT has two start modes: auto-start mode, in which counting automatically starts after a reset is released, and register start mode, in which counting is started by refresh operation (writing to the register).
In auto-start mode, counting automatically starts after a reset is released in accordance with the settings in option function select register 0 (OFS0) in the ROM.
In register start mode, counting is started by refresh operation (writing to the register) after the respective registers are set after a reset is released.
Select auto-start mode or register start mode by setting the OFS0.WDTSTRT bit.
When the auto-start mode is selected, the settings in the WDTCR and WDTRCR registers are disabled, and the settings in the OFS0 register are enabled.
On the other hand, when the register start mode is selected, the setting of the OFS0 register is disabled, and the settings of the WDTCR and WDTRCR registers are enabled.

32.3.1.1 Register Start Mode

When the OFS0.WDTSTRT bit is 1, register start mode is selected, and the WDTCR and WDTRCR registers are enabled.
After a reset is released, set the clock division ratio, window start and end positions, and timeout period in the WDTCR register, and the reset output or interrupt request output in the WDTRCR register. Then, refresh the down-counter to start counting down from the value set by the WDTCR.TOPS[1:0] bits.
Thereafter, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The WDT does not output the reset signal as long as this continues. However, if the down-counter underflows because the down-counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the WDT outputs a reset signal or a non-maskable interrupt request/interrupt request (WUNI). Reset output or interrupt request output can be selected by setting the WDTRCR.RSTIRQS bit.

Figure 32.3 shows an example of operation under the following conditions.

- Register start mode (OFS0.WDTSTRT = 1)
- Reset output is enabled (WDTRCR.RSTIRQS = 1)
- The window start position is 75% (WDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (WDTCR.RPES[1:0] = 10b)

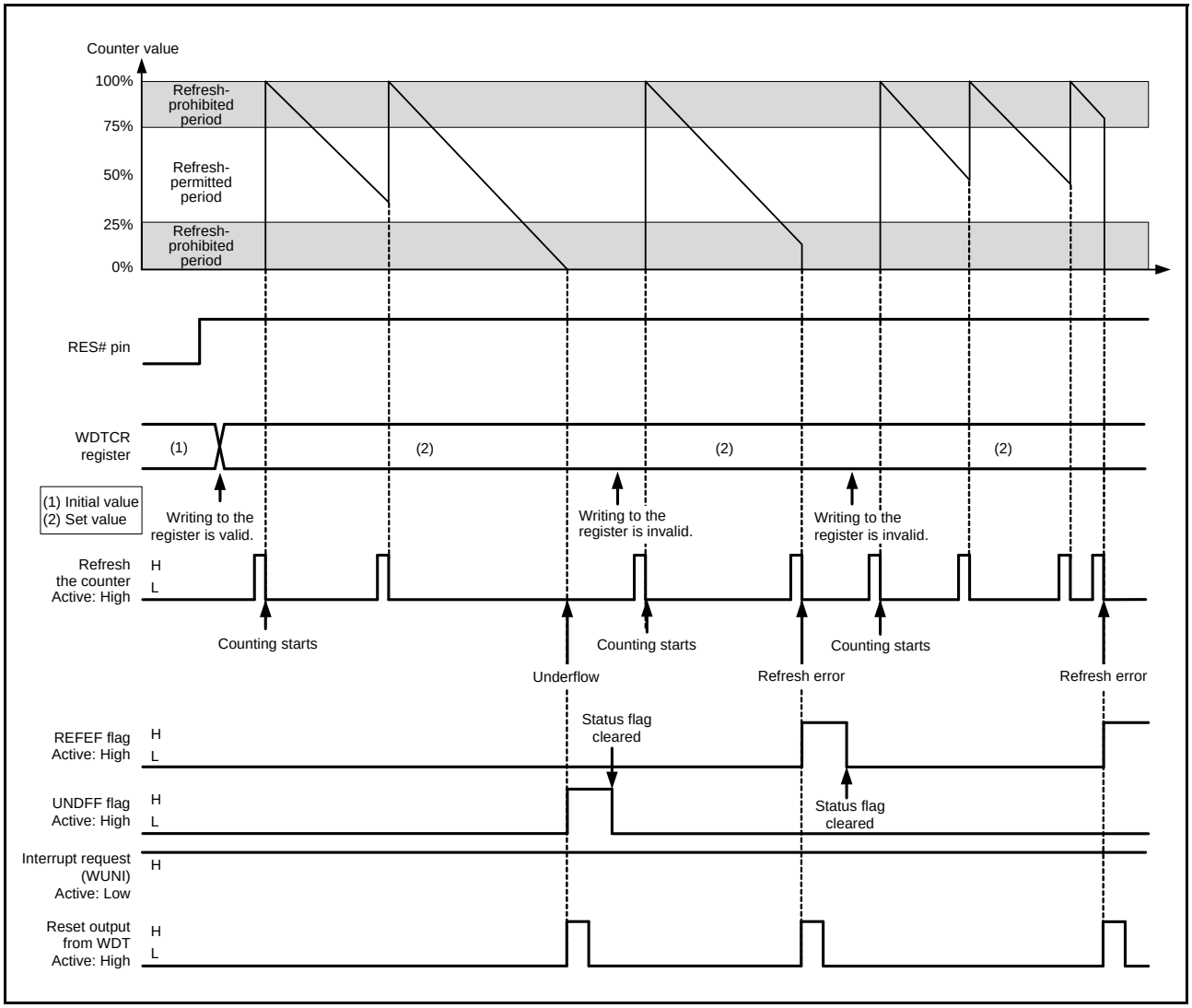


Figure 32.3 Operation Example in Register Start Mode

32.3.1.2 Auto-Start Mode

When the WDTSTRT bit in option function select register 0 (OFS0) is 0, auto-start mode is selected, the WDTCR and WDTRCR registers are disabled, and the settings in the OFS0 register are enabled.
Within the reset state, the setting values (clock division ratio, window start and end positions, timeout period, and reset output or interrupt request) of the OFS0 register are set in the WDT registers.
When the reset is released, the down-counter automatically starts counting down from the value set by the OFS0.WDTTOPS[1:0] bits.
After that, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The WDT does not output the reset signal as long as this continues.
However, if the down-counter underflows because refreshing of the down-counter is not possible due to the program having entered crashed execution or if a refresh error occurs due to refreshing outside the refresh-permitted period, the WDT outputs the reset signal or non-maskable interrupt request/interrupt request (WUNI).
After the reset signal or non-maskable interrupt request/interrupt request is output of for one cycle of counting, the value of the timeout period is set in the down-counter counting is restarted.
Reset output or interrupt request output can be selected by setting the OFS0.WDTRSTIRQS bit.

Figure 32.4 shows an example of operation (non-maskable interrupt) under the following conditions.

- Auto start mode (OFS0.WDTSTRT = 0)
- Non-maskable interrupt request output is enabled (OFS0.WDTRSTIRQS = 0)
- The window start position is 75% (OFS0.WDTRPSS[1:0] = 10b)
- The window end position is 25% (OFS0.WDTRPES[1:0] = 10b)

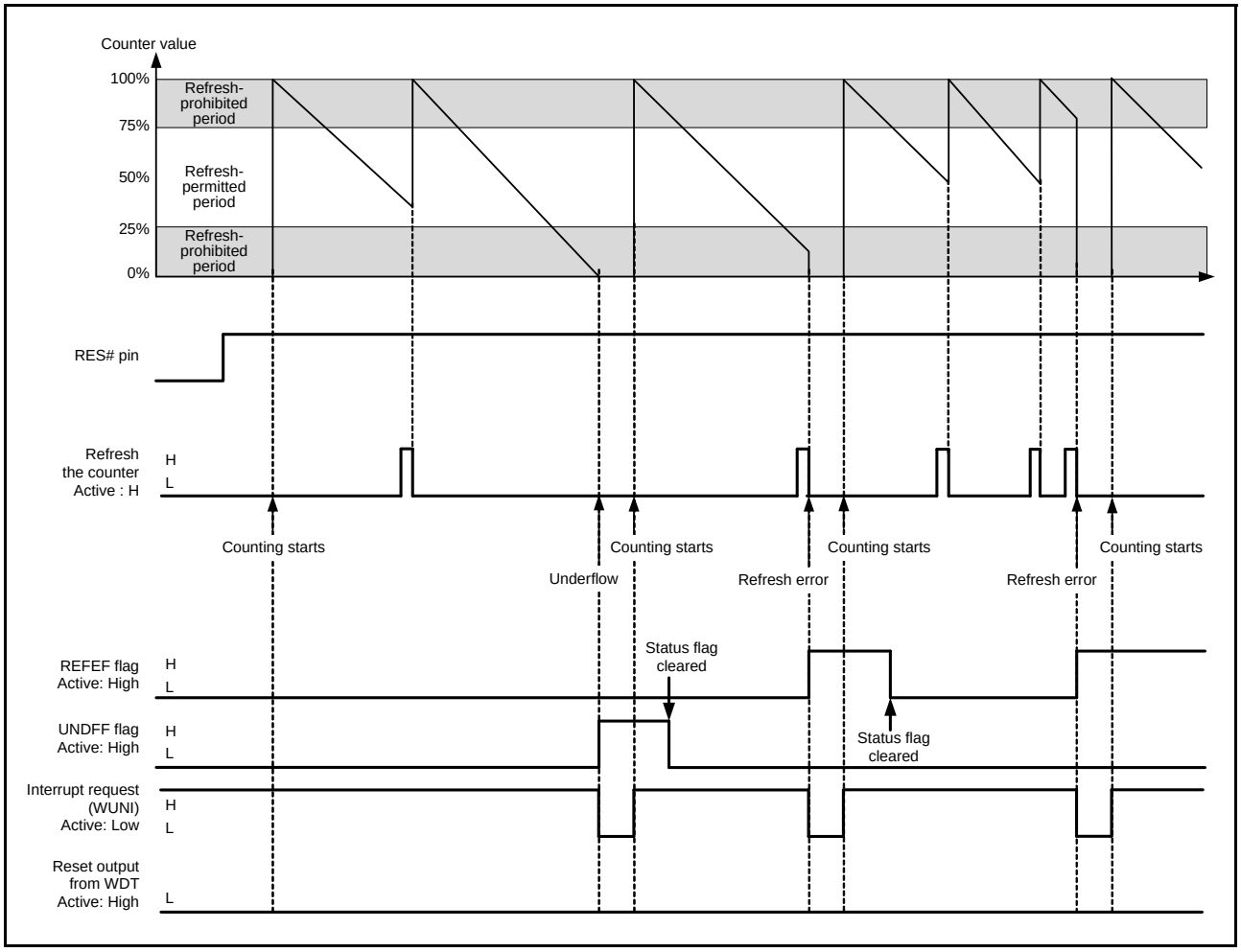


Figure 32.4 Operation Example in Auto-Start Mode

32.3.2 Control over Writing to the WDTCR and WDTRCR Registers

Writing to the WDTCR or WDTRCR register 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 the WDTCR or WDTRCR register, the protection signal in the WDT becomes 1 to protect the WDTCR and WDTRCR registers 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 32.5 shows control waveforms produced in response to writing to the WDTCR register.

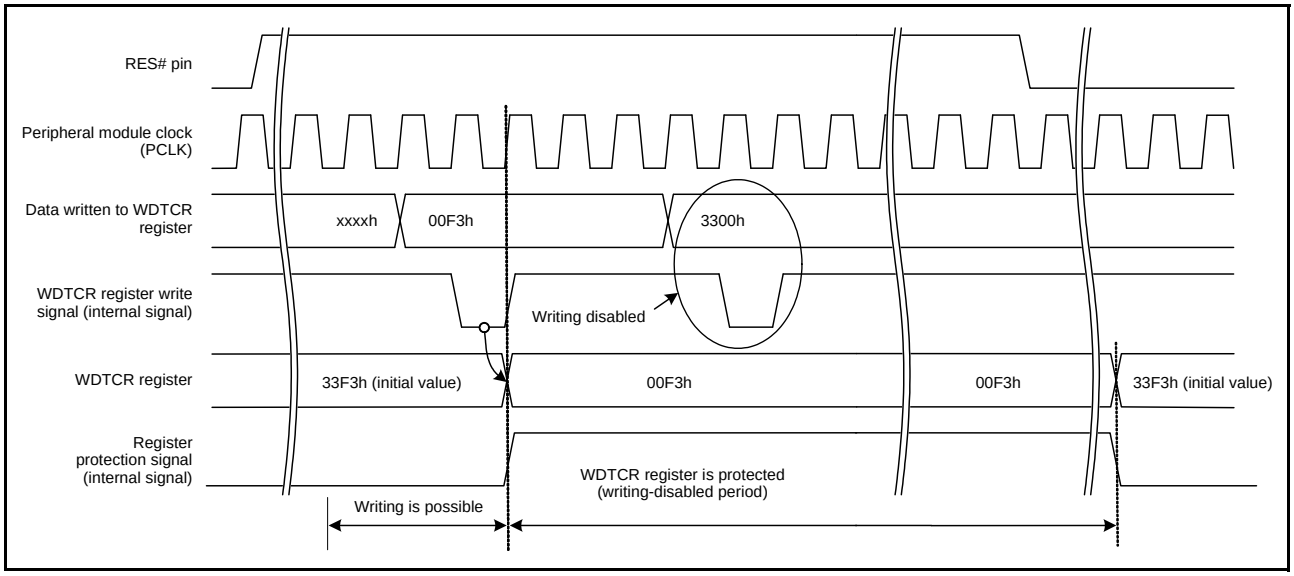


Figure 32.5 Control Waveforms Produced in Response to Writing to the WDTCR Register

32.3.3 Refresh Operation

The down-counter is refreshed by writing the values 00h and then FFh to the WDTRR register. If a value other than FFh is written after 00h, the down-counter is not refreshed. After such invalid writing, correct refreshing is performed by again writing to 00h and then FFh to the WDTRR register.
Even if a register other than the WDTRR register is accessed or the WDTRR register is read between writing 00h and writing FFh to the WDTRR register, correct refreshing will be done.
Writing to refresh the counter must be performed within the refresh-permitted period. Whether writing is done within the refresh-permitted period is determined when writing FFh. For this reason, correct refreshing will be done even if 00h is written outside the refresh-permitted period.

[Sample sequences of writing that are valid for refreshing the counter]

- 00h → FFh
- 00h (n–1-th time) → 00h (nth time) → FFh
- 00h → access to another register or read from the WDTRR register → FFh

[Sample sequences of writing that are not valid for refreshing the counter]

- 23h (a value other than 00h) → FFh
- 00h → 54h (a value other than FFh)
- 00h→ AAh (00h and a value other than FFh) → FFh

After FFh is written to the WDTRR register, refreshing the down-counter requires up to four cycles of the signal for counting. Therefore, writing FFh to the WDTRR register should be completed four-count cycles before the down-counter underflows.

Figure 32.6 shows the WDT refresh-operation waveforms when the clock division ratio = PCLK/64.

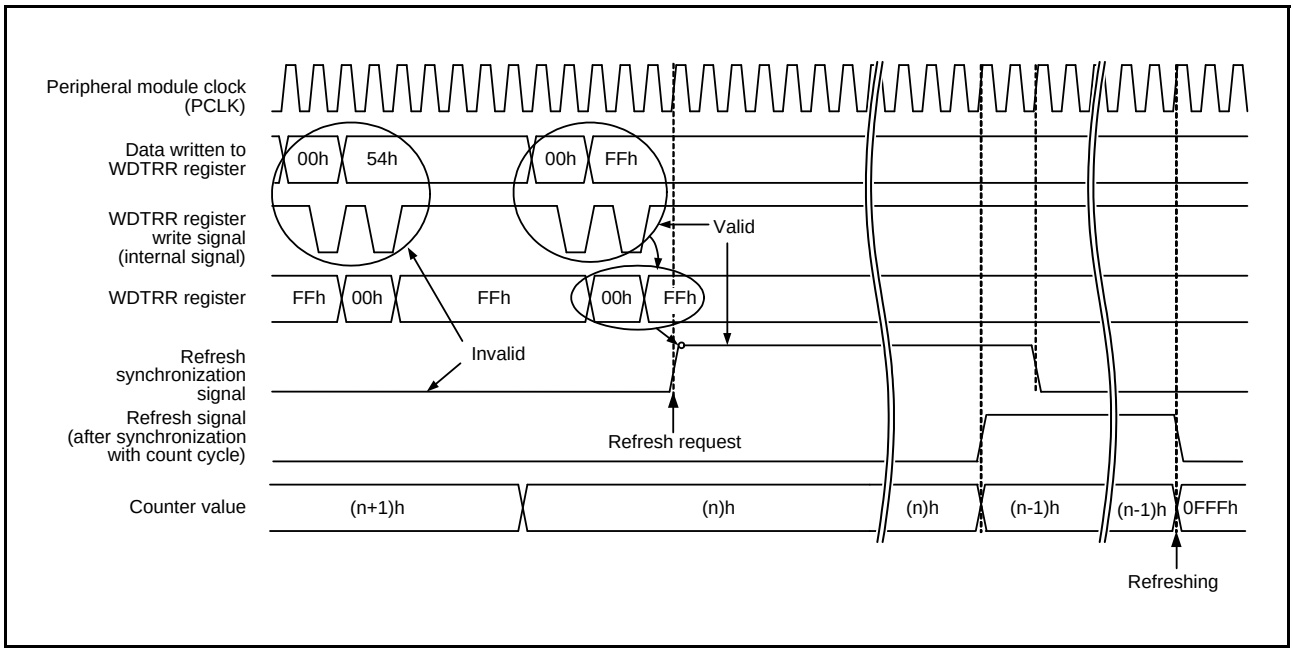


Figure 32.6 WDT Refresh Operation Waveforms (WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b)

32.3.4 Reset Output

When the WDTRCR.RSTIRQS bit is set to 1 in register start mode or when the WDTRSTIRQS bit in option function select register 0 (OFS0) is set to 1 in auto-start mode, a reset signal is output for one-count cycle when an underflow in the down-counter or a refresh error occurs.

In register start mode, the down-counter is initialized (all bits set to 0) and stopped in that state after output of the reset signal. After the reset is released and the program is restarted, the counter is set up again and counting down is started by refreshing.

In auto-start mode, counting down automatically starts after the reset is released.

32.3.5 Interrupt Source

When the WDTRCR.RSTIRQS bit is set to 0 in register start mode or when the OFS0.WDTRSTIRQS bit is set to 0 in auto-start mode, an interrupt (WUNI) signal is generated when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt and an interrupt. For details, refer to section 15, Interrupt Controller (ICUE).

Table 32.4 WDT Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
WUNI	Down-counter underflow Refresh error	Not possible	Not possible

32.3.6 Reading the Down-Counter Value

The WDT stores the counter value in the WDTSR.CNTVAL[13:0] bits. Thus, the counter value can be checked through the WDTSR.CNTVAL[13:0] bits.

Figure 32.7 shows the processing for reading the WDT down-counter value when the clock division ratio = PCLK/64.

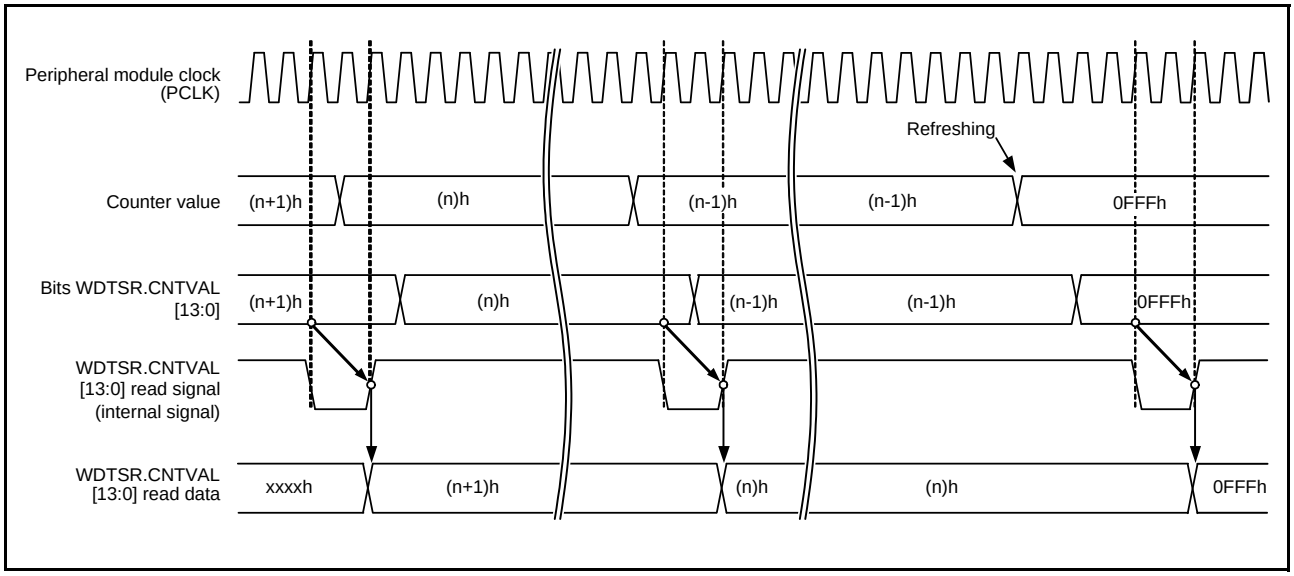


Figure 32.7 Processing for Reading WDT Down-Counter Value (WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b)

32.3.7 Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers

Table 32.5 lists the correspondence between option function select register 0 (OFS0) used in auto-start mode and the registers used in register start mode.

Do not change the OFS0 register setting during WDT operation.

For details on the OFS0 register, refer to section 7.2.3, Option Function Select Register 0 (OFS0).

Table 32.5 Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers

Target of Control	Function	OFS0 Register (Enabled in Auto-Start Mode) OFS0.WDTSTRT = 0	WDT Registers (Enabled in Register Start Mode) OFS0.WDTSTRT = 1
Down-counter	Timeout period selection	OFS0.WDTTOPS[1:0]	WDTCR.TOPS[1:0]
	Clock division ratio selection	OFS0.WDTCKS[3:0]	WDTCR.CKS[3:0]
	Window start position selection	OFS0.WDTRPSS[1:0]	WDTCR.RPSS[1:0]
	Window end position selection	OFS0.WDTRPES[1:0]	WDTCR.RPES[1:0]
Reset output or interrupt request output	Reset output or interrupt request output selection	OFS0.WDTRSTIRQS	WDTRCR.RSTIRQS

33. Independent Watchdog Timer (IWDTa)

In this section, “PCLK” is used to refer to PCLKB.

33.1 Overview

The independent watchdog timer (IWDT) can be used to detect programs being out of control.
The user can detect when a program runs out of control if an underflow occurs, by creating a program that refreshes the IWDT counter before it underflows.
The functions of the IWDT are different from those of the WDT in the following respects.

- The divided IWDT-dedicated clock (IWDTCLK) is used as the count source (not affected by the PCLK).
- When making a transition to sleep mode, software standby mode, deep software standby mode, or all-module clock stop mode, the IWDTCSTPR.SLCSTP bit or the OFS0.IWDTSLCSTP bit can be used to select whether to stop the counter or not.

Table 33.1 lists the specifications of the IWDT and Figure 33.1 shows a block diagram of the IWDT.

Table 33.1 IWDT Specifications

Item	Description
Count source*1	IWDT-dedicated clock (IWDTCLK)
Clock divide ratio	Divide by 1, 16, 32, 64, 128, or 256
Counter operation	Counting down using a 14-bit down-counter
Conditions for starting the counter	• Auto-start mode: Counting automatically starts after a reset is released • Register start mode: Counting is started by refresh operation (writing 00h and then FFh to the IWDTRR register).
Conditions for stopping the counter	• Reset (the down-counter and other registers return to their initial values) • In low power consumption states (depends on the register setting*2) • A counter underflows or a refresh error occurs (only 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)
Non-maskable interrupt/ interrupt sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Reading the counter value	The down-counter value can be read by the IWDTSR register.
Event link function (output)	• Down-counter underflow event output • Refresh error event output
Output signal (internal signal)	• Reset output • Interrupt request output • Sleep mode count stop control output
Auto-start mode (controlled by option function select register 0 (OFS0))	• Selecting the clock frequency divide ratio after a reset (OFS0.IWDTCKS[3:0] bits) • Selecting the timeout period of the independent watchdog timer (OFS0.IWDTTOPS[1:0] bits) • Selecting the window start position in the independent watchdog timer (OFS0.IWDTRPSS[1:0] bits) • Selecting the window end position in the independent watchdog timer (OFS0.IWDTRPES[1:0] bits) • Selecting the reset output or interrupt request output (OFS0.IWDTRSTIRQS bit) • Selecting the down-count stop function at transition to sleep mode, software standby mode, deep software standby mode, or all-module clock stop mode (OFS0.IWDTSLCSTP bit)
Register start mode (controlled by the IWDT registers)	• Selecting the clock frequency divide ratio after refreshing (IWDTCR.CKS[3:0] bits) • Selecting the timeout period of the independent watchdog timer (IWDTCR.TOPS[1:0] bits) • Selecting the window start position in the independent watchdog timer (IWDTCR.RPSS[1:0] bits) • Selecting the window end position in the independent watchdog timer (IWDTCR.RPES[1:0] bits) • Selecting the reset output or interrupt request output (IWDTRCR.RSTIRQS bit) • Selecting the down-count stop function at transition to sleep mode, software standby mode, deep software standby mode, or all-module clock stop mode (IWDTCSTPR.SLCSTP bit)

Note 1. Satisfy the frequency of the peripheral module clock (PCLK) ≥ 4 × (the frequency of the count source after divide).
Note 2. When the OFS0.IWDTSLCSTP bit is 1 in auto-start mode, and when the IWDTCSTPR.SLCSTP bit is 1 in register start mode.

To use the IWDT, the IWDT-dedicated clock (IWDTCLK) should be supplied so that the IWDT operates even if the peripheral module clock (PCLK) stops. The bus interface and registers operate with PCLK, and the 14-bit counter and control circuits operate with IWDTCLK.
Figure 33.1 is a block diagram of the IWDT.

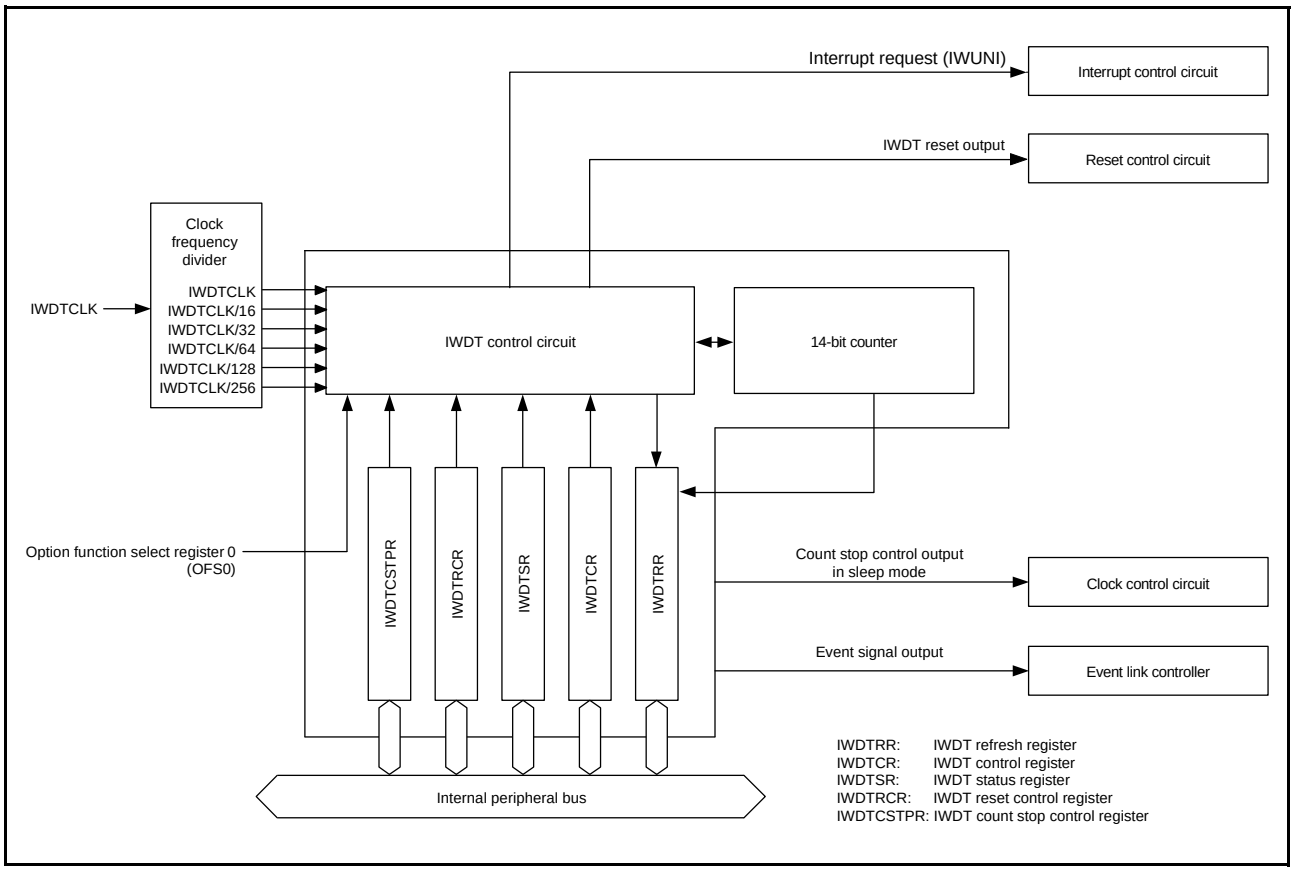
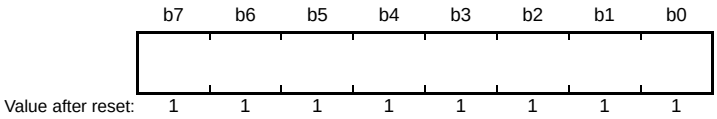


Figure 33.1 IWDT Block Diagram

33.2 Register Descriptions

33.2.1 IWDT Refresh Register (IWDTRR)

Address(es): IWDT.IWDTRR 0008 8030h



Bit	Description	R/W
b7 to b0	The counter is refreshed by writing 00h and then writing FFh to this register.	R/W

The IWDTRR register refreshes the counter of the IWDT.
The counter of the IWDT is refreshed by writing 00h and then writing FFh to the IWDTRR register (refresh operation) within the refresh-permitted period.
After the counter has been refreshed, it starts counting down from the value selected by the IWDTTOPS[1:0] bits in option function select register 0 (OFS0) in auto-start mode. In register start mode, counting down starts from the value selected by setting the IWDTCR.TOPS[1:0] bits in the first refresh operation after a reset is released.
When 00h is written, the read value is 00h. When a value other than 00h is written, the read value is FFh.
For details of the refresh operation, refer to section 33.3.3, Refresh Operation.

33.2.2 IWDT Control Register (IWDTCR)

Address(es): IWDT.IWDTCR 0008 8032h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TOPS[1:0]	Timeout Period Select	b1 b0 0 0: 1024 cycles (03FFh) 0 1: 4096 cycles (0FFFh) 1 0: 8192 cycles (1FFFh) 1 1: 16384 cycles (3FFFh)	R/W
b3, b2	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7 to b4	CKS[3:0]	Clock Divide Ratio Select	b7 b4 0 0 0 0: No division 0 0 1 0: Divide-by-16 0 0 1 1: Divide-by-32 0 1 0 0: Divide-by-64 1 1 1 1: Divide-by-128 0 1 0 1: Divide-by-256 Other settings are prohibited.	R/W
b9, b8	RPES[1:0]	Window End Position Select	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (window end position is not specified.)	R/W
b11, b10	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b13, b12	RPSS[1:0]	Window Start Position Select	b13 b12 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (window start position is not specified.)	R/W
b15, b14	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R

There are some restrictions on writing to the IWDTCR register. For details, refer to section 33.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.

In auto-start mode, the settings in the IWDTCR register are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting made to the IWDTCR register can also be made in the OFS0 register. For details, refer to section 33.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

TOPS[1:0] Bits (Timeout Period Select)

These bits select the timeout period (period until the counter underflows) from among 1024, 4096, 8196, or 16384 cycles, taking the divided clock specified by the CKS[3:0] bits as one cycle.
After the counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of IWDTCLK cycles) until the counter underflows.
Relations between the CKS[3:0] and TOPS[1:0] bit setting, the timeout period, and the number of IWDTCLK cycles are listed in Table 33.2.

Table 33.2 Settings and Timeout Periods

CKS[3:0] Bits				TOPS[1:0] Bits				
b7	b6	b5	b4	b1	b0	Clock Divide Ratio	Timeout Period (Number of Cycles)	Cycles of IWDTCLK
0	0	0	0	0	0	No division	1024	1024
				0	1		4096	4096
				1	0		8192	8192
				1	1		16384	16384
0	0	1	0	0	0	Divide-by-16	1024	16384
				0	1		4096	65536
				1	0		8192	131072
				1	1		16384	262144
0	0	1	1	0	0	Divide-by-32	1024	32768
				0	1		4096	131072
				1	0		8192	262144
				1	1		16384	524288
0	1	0	0	0	0	Divide-by-64	1024	65536
				0	1		4096	262144
				1	0		8192	524288
				1	1		16384	1048576
1	1	1	1	0	0	Divide-by-128	1024	131072
				0	1		4096	524288
				1	0		8192	1048576
				1	1		16384	2097152
0	1	0	1	0	0	Divide-by-256	1024	262144
				0	1		4096	1048576
				1	0		8192	2097152
				1	1		16384	4194304

CKS[3:0] Bits (Clock Divide Ratio Select)

These bits select the IWDTCLK clock divide ratio from among divide-by 1, 16, 32, 64, 128, and 256. Combination with the TOPS[1:0] bit setting, a count period between 1024 and 4194304 cycles of the IWDTCLK clock can be selected for the IWDT.

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

These bits select 75%, 50%, 25% or 0% of the count period for the window end position of the counter. The window end position should be a value smaller than the window start position (window start position > window end position). If the window end position is greater than the window start position, only the window start position setting is enabled.
The counter values for the window start and end positions selected by setting the RPSS[1:0] and RPES[1:0] bits change depending on the TOPS[1:0] bit setting.
Table 33.3 lists the counter values for the window start and end positions corresponding to TOPS[1:0] bit values.

Table 33.3 Relationship between Timeout Period and Window Start and End Counter Values

TOPS[1:0] Bits		Timeout Period		Window Start and End Counter Value			
b1	b0	Cycles	Counter Value	100%	75%	50%	25%
0	0	1024	03FFh	03FFh	02FFh	01FFh	00FFh
0	1	4096	0FFFh	0FFFh	0BFFh	07FFh	03FFh
1	0	8192	1FFFh	1FFFh	17FFh	0FFFh	07FFh
1	1	16384	3FFFh	3FFFh	2FFFh	1FFFh	0FFFh

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

These bits select a counter window start position from 100%, 75%, 50%, or 25% of the count period (100% when the count starts and 0% when the counter underflows). The interval between the window start position and window end position is the refresh-permitted period and the other periods are refresh-prohibited periods.
Figure 33.2 shows the relationship between of the RPSS[1:0] and RPES[1:0] bit setting and the refresh-permitted and refresh-prohibited periods.

RPSS[1:0] Bits		RPES[1:0] Bits		Window	
b13	b12	b9	b8	Start (%)	End (%)
1	1	1	1	100	0
		1	0		25
		0	1		50
		0	0		75
1	0	1	1	75	0
		1	0		25
		0	1		50
		0	0		75
0	1	1	1	50	0
		1	0		25
		0	1		50
		0	0		75
0	0	1	1	25	0
		1	0		25
		0	1		50
		0	0		75

Note: If window end setting is greater than or equal to window start setting, the window end setting is set to 0%.

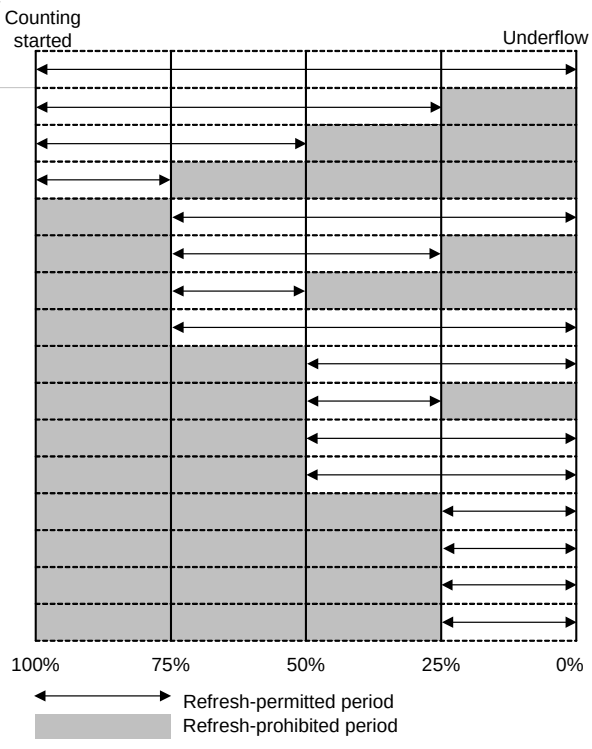


Figure 33.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period

33.2.3 IWDT Status Register (IWDTSR)

Address(es): IWDT.IWDTSR 0008 8034h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REFEF	UNDFF	CNTVAL[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b13 to b0	CNTVAL[13:0]	Counter Value	Value counted by the counter	R
b14	UNDFF	Underflow Flag	0: No underflow occurred 1: Underflow occurred	R/(W) *1
b15	REFEF	Refresh Error Flag	0: No refresh error occurred 1: Refresh error occurred	R/(W) *1

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

The IWDTSR register is initialized by the reset source of the IWDT. The IWDTSR register is not initialized by other reset sources.

CNTVAL[13:0] Bits (Counter Value)

These bits are used to confirm the counter value of the counter, but note that the read value may differ from the actual count by a value of one count.

UNDFF Flag (Underflow Flag)

This bit is used to confirm whether or not an underflow has occurred in the counter.
The value 1 indicates that the counter has underflowed. The value 0 indicates that the counter has not underflowed.
Write 0 to the UNDFF flag to set the value to 0. Writing 1 has no effect.

REFEF Flag (Refresh Error Flag)

This bit is used to confirm whether or not a refresh error (performing a refresh operation during a refresh-prohibited period).
The value 1 indicates that a refresh error has occurred. The value 0 indicates that no refresh error has occurred.
Write 0 to the REFEF flag to set the value to 0. Writing 1 has no effect.

33.2.4 IWDT Reset Control Register (IWDTRCR)

Address(es): IWDT.IWDTRCR 0008 8036h

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

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	RSTIRQS	Reset Interrupt Request Select	0: Non-maskable interrupt request or interrupt request output is enabled. 1: Reset output is enabled.	R/W

There are some restrictions on writing to the IWDTRCR register. For details, refer to section 33.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.

In auto-start mode, the IWDTRCR register setting are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting mode to the IWDTRCR register can also be made in the OFS0 register. For details, refer to section 33.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

33.2.5 IWDT Count Stop Control Register (IWDTCSTPR)

Address(es): IWDT.IWDTCSTPR 0008 8038h

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

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	SLCSTP	Sleep Mode Count Stop Control	0: Count stop is disabled. 1: Count is stopped at a transition to sleep mode, software standby mode, deep software standby mode, or all-module clock stop mode.	R/W

The IWDTCSTPR register controls whether to stop the IWDT counter in a low power consumption state. There are some restrictions on writing to the IWDTCSTPR register. For details, refer to section 33.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.
In auto-start mode, the IWDTCSTPR register setting are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting mode to the IWDTCSTPR register can also be made in the OFS0 register. For details, refer to section 33.3.8, 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.

33.2.6 Option Function Select Register 0 (OFS0)

For option function select register 0 (OFS0), refer to section 33.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

33.3 Operation

33.3.1 Count Operation in Each Start Mode

Select the IWDT start mode by setting the IWDTSTRT bit in option function select register 0 (OFS0).
When the OFS0.IWDTSTRT bit is 1 (register start mode), the IWDTCR, IWDTRCR, and IWDTCSTPR registers are enabled, and counting is started by refresh operation (writing) to the IWDTRR register. When the OFS0.IWDTSTRT bit is 0 (auto-start mode), the setting of the OFS0 register is enabled, and counting automatically starts after reset.

33.3.1.1 Register Start Mode

When the OFS0.IWDTSTRT bit in option function select register 0 is 1, register start mode is selected, and the IWDTCR, IWDTRCR, and IWDTCSTPR registers are enabled.
After a reset is released, set the clock divide ratio, window start and end positions, and timeout period in the IWDTCR register, the reset output or interrupt request output in the IWDTRCR register, and the counter stop control at transitions to low power consumption states in the IWDTCSTPR register. Then refresh the counter to start counting down from the value selected by setting the 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 counter underflows because the counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the IWDT outputs a reset signal or a non-maskable interrupt request/interrupt request (IWUNI). Set the IWDTRCR.RSTIRQS bit to select either reset output or interrupt request output.

Figure 33.3 shows an example of operation under the following conditions.

- Register start mode (OFS0.IWDTSTRT = 1)
- Reset output is enabled (IWDTRCR.RSTIRQS = 1)
- The window start position is 75% (IWDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (IWDTCR.RPES[1:0] = 10b)

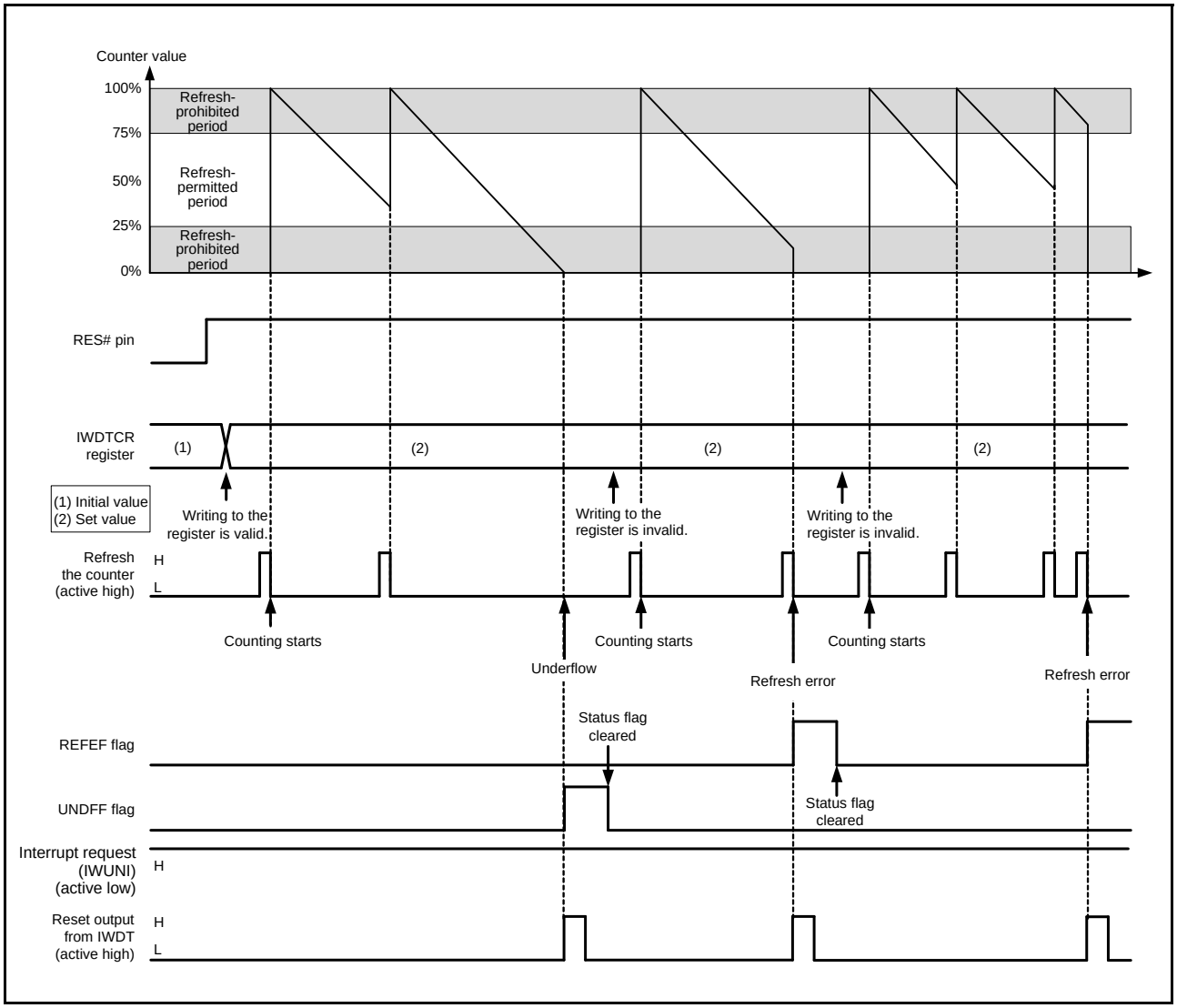


Figure 33.3 Operation Example in Register Start Mode

33.3.1.2 Auto-Start Mode

When the IWDTSTRT bit in option function select register 0 (OFS0) is 0, auto-start mode is selected, and the IWDTCR, IWDTRCR, and IWDTCSTPR registers are disabled.
Within the reset state, the clock divide ratio, window start and end positions, timeout period, reset output or interrupt request output, and counter stop control at transitions to low power consumption states are set using the values specified in the OFS0 register. When the reset is released, the counter automatically starts counting down from the value selected by the 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 each time the counter is refreshed and counting down continues. The IWDT does not output the reset signal as long as this continues. However, if the counter underflows because refreshing of the counter is not possible due to the program having entered crashed execution or if a refresh error occurs due to refreshing outside the refresh-permitted period, the IWDT outputs the reset signal or non-maskable interrupt request/interrupt request (IWUNI). After the reset signal or non-maskable interrupt request/interrupt request (IWUNI) is generated, the counter reloads the timeout period after counting for one cycle, and restarts counting. Set the OFS0.IWDTRSTIRQS bit to select either reset output or interrupt request output.

Figure 33.4 shows an example of operation under the following conditions.

- Auto-start mode (OFS0.IWDTSTRT = 0)
- Non-maskable interrupt request output is enabled (OFS0.IWDTRSTIRQS = 0)
- The window start position is 75% (OFS0.IWDTRPSS[1:0] = 10b)
- The window end position is 25% (OFS0.IWDTRPES[1:0] = 10b)

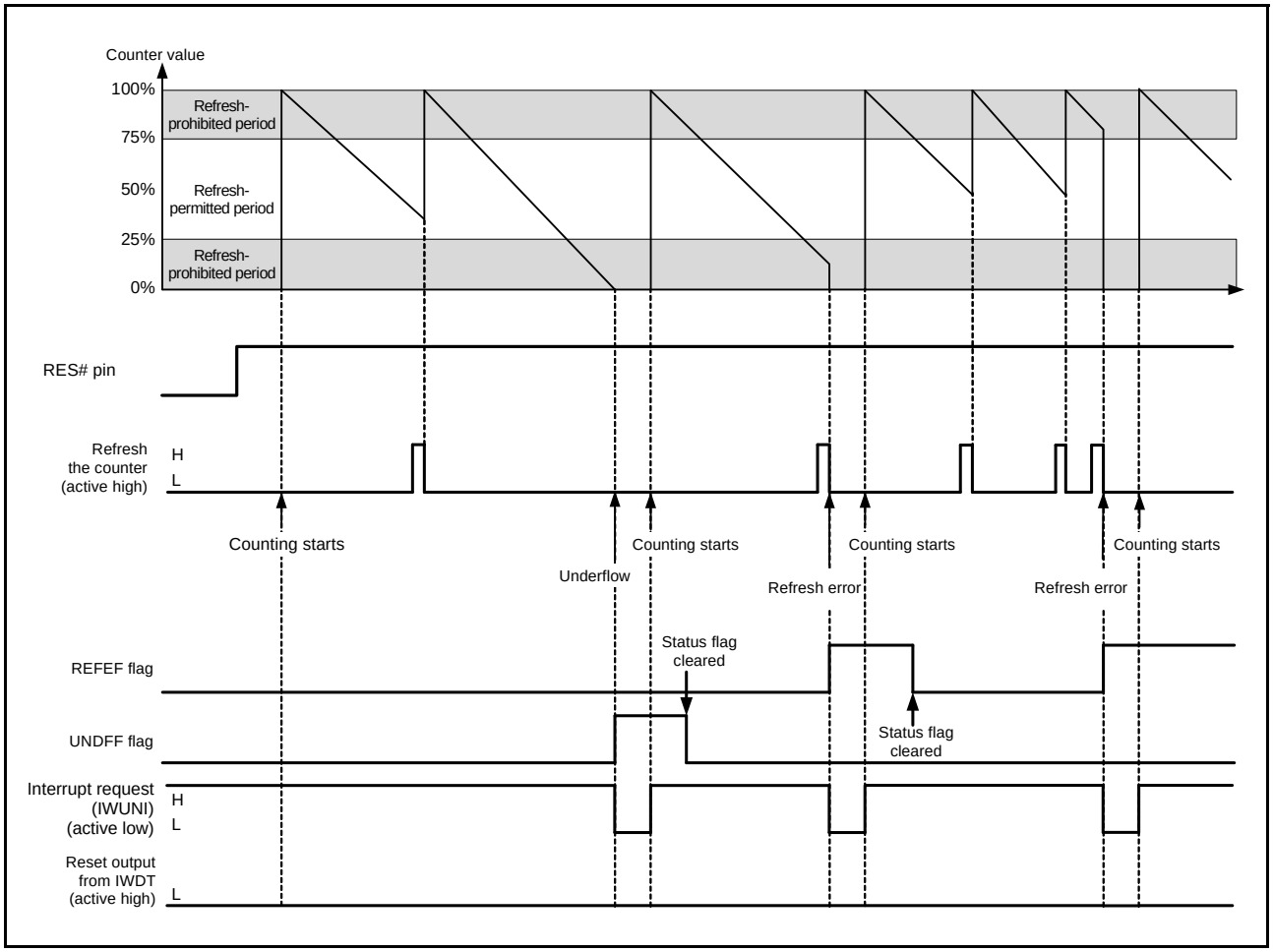


Figure 33.4 Operation Example in Auto-Start Mode

33.3.2 Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers

Writing to the IWDTCR, IWDTRCR, or IWDTCSTPR register is only possible once between the release from the reset state and the first refresh operation.

After a refresh operation (counting starts) or the IWDTCR, IWDTRCR, or IWDTCSTPR register is written to, the protection signal in the IWDT becomes 1 to protect registers IWDTCR, IWDTRCR, and IWDTCSTPR against subsequent attempts at writing.

This protection is released by the reset source of the IWDT. With other reset sources, the protection is not released.

Figure 33.5 shows control waveforms produced in response to writing to the IWDTCR register.

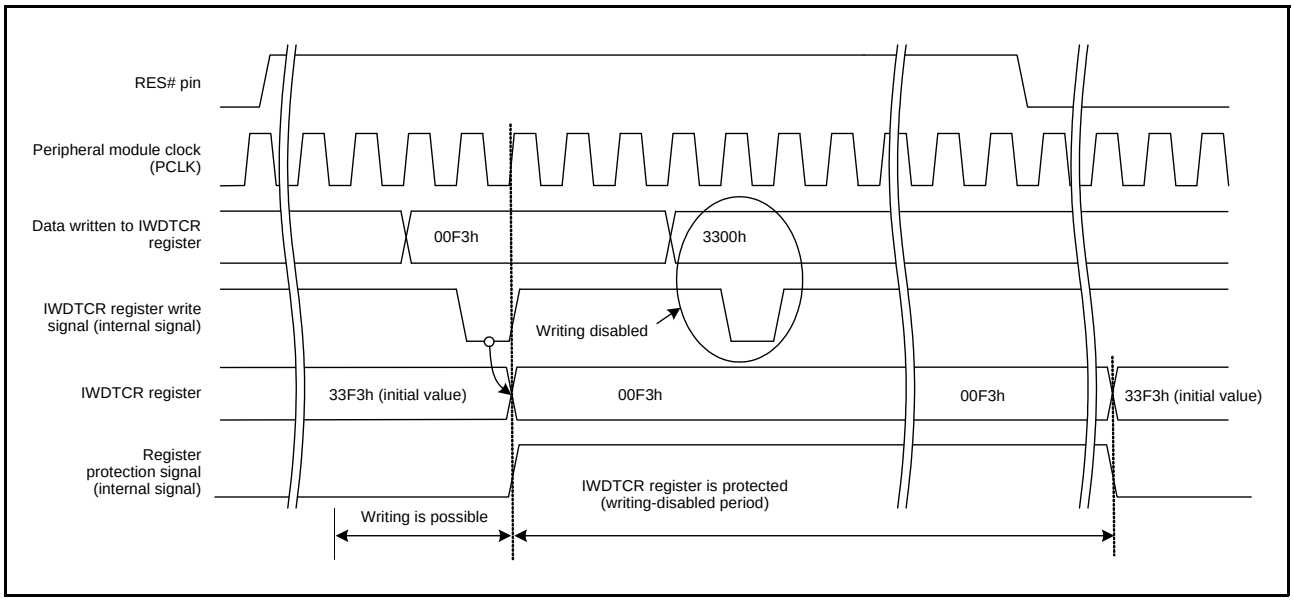


Figure 33.5 Control Waveforms Produced in Response to Writing to the IWDTCR Register

33.3.3 Refresh Operation

The counter is refreshed and starts operation (counting is started by refreshing) by writing the values 00h and then FFh to the IWDTRR register. If a value other than FFh is written after 00h, the counter is not refreshed. After such invalid writing, correct refreshing is performed by again writing 00h and then FFh to the IWDTRR register.
When writing is done in the order of 00h (first time) → 00h (second time), and if FFh is written after that, the writing order 00h → FFh is satisfied; writing 00h (n–1-th time) → 00h (nth time) → FFh is valid and correct refreshing will be done. Even when the first value written before 00h is not 00h, correct refreshing will be done if the operation contains the set of writing 00h → FFh. Moreover, even if a register other than the IWDTRR register is accessed or the IWDTRR register is read between writing 00h and writing FFh to the IWDTRR register, correct refreshing will be done.

[Sample sequences of writing that are valid for refreshing the counter]

- 00h → FFh
- 00h (n–1-th time) → 00h (nth time) → FFh
- 00h → access to another register or read from the IWDTRR register → FFh

[Sample sequences of writing that are not valid for refreshing the counter]

- 23h (a value other than 00h) → FFh
- 00h → 54h (a value other than FFh)
- 00h → AAh (00h and a value other than FFh) → FFh

Even when 00h is written to the IWDTRR register outside the refresh-permitted period, if FFh is written to the IWDTRR register in the refresh-permitted period, the writing sequence is valid and refreshing will be done.
After FFh is written to the IWDTRR register, refreshing the counter requires up to four cycles of the signal for counting (the IWDTCR.CKS[3:0] bits determine how many cycles of the IWDT-dedicated clock (IWDTCLK) make up one cycle for counting). Therefore, writing FFh to the IWDTRR register should be completed four-count cycles before the end position of the refresh-permitted period or a counter underflow. The value of the counter can be checked by the IWDTSR.CNTVAL[13:0] bits.

[Sample refreshing timings]

- When the window start position is set to 1FFFh, even if 00h is written to the IWDTRR register before 1FFFh is reached (2002h, for example), refreshing is done if FFh is written to the IWDTRR register after the value of the IWDTSR.CNTVAL[13:0] bits has reached 1FFFh.
- When the window end position is set to 1FFFh, refreshing is done if 2003h (four-count cycles before 1FFFh) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to the IWDTRR register.
- When the refresh-permitted period continues until count 0000h, refreshing can be done immediately before an underflow. In this case, if 0003h (four-count cycles before an underflow) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to the IWDTRR register, no underflow occurs and refreshing is done.

Figure 33.6 shows the IWDT refresh-operation waveforms when PCLK > IWDTCLK and clock divide ratio = IWDTCLK.

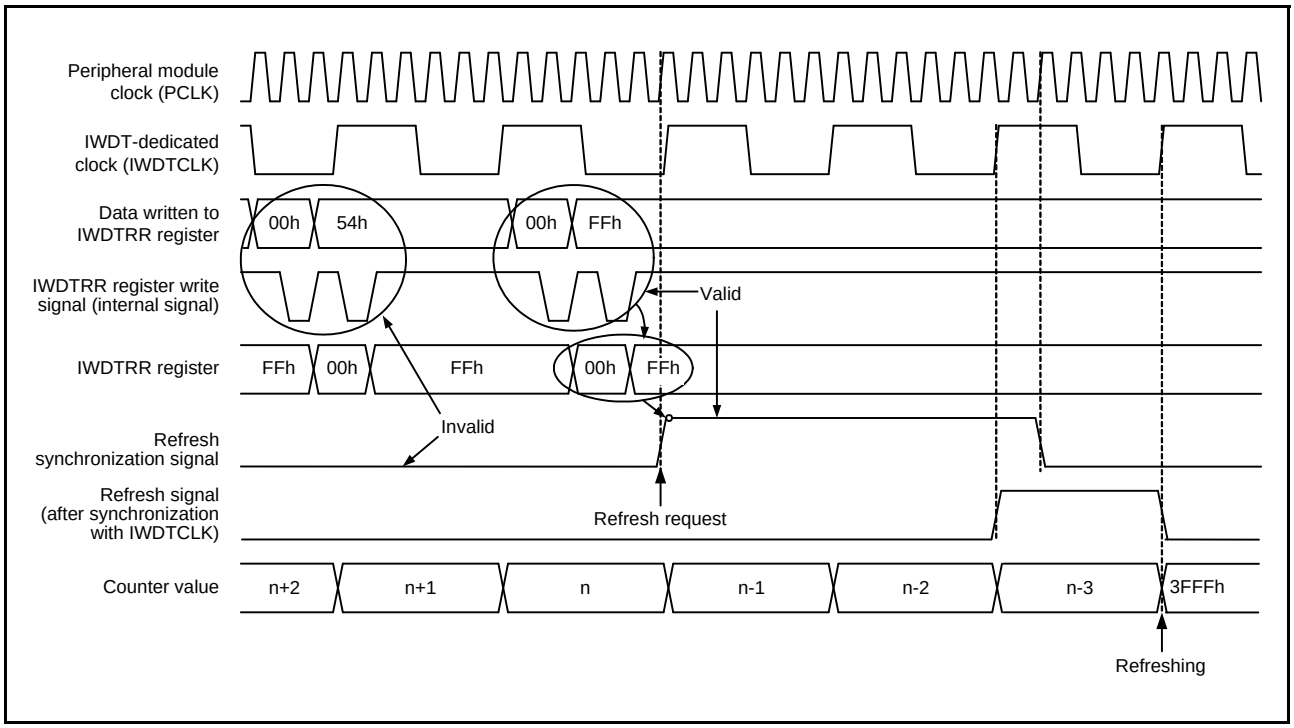


Figure 33.6 IWDT Refresh Operation Waveforms (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

33.3.4 Status Flags

The IWDTSR.REFEF and IWDTSR.UNDFF flags retain the source of the reset signal output from the IWDT or the source of the interrupt request from the IWDT.
Thus, after release from the reset state or interrupt request generation, read the IWDTSR.REFEF and IWDTSR.UNDFF flags to check for the reset or interrupt source.
For each flag, writing 0 clears the bit and writing 1 has no effect.
Leaving the status flags unchanged does not affect operation. If the flags are not cleared, at the time of the next reset or interrupt request from the IWDT, the earlier reset or interrupt source is cleared and the new reset or interrupt source is written.
After 0 is written to each flag, up to three IWDTCLK cycles and two PCLK cycles are required before the value is reflected.

33.3.5 Reset Output

When the IWDTRCR.RSTIRQS bit is set to 1 in register start mode or when the IWDTRSTIRQS bit in option function select register 0 (OFS0) is set to 1 in auto-start mode, a reset signal is output when an underflow in the counter or a refresh error occurs.
In register start mode, the counter is initialized (0000h) and kept in that state after assertion of the reset signal. After the reset is released and the program is restarted, the counter is set up again and counting down is started by refreshing.
In auto-start mode, counting down automatically starts after the reset output.

33.3.6 Interrupt Sources

When the IWDTRCR.RSTIRQS bit is set to 0 in register start mode or when the OFS0.IWDTRSTIRQS bit is set to 0 in auto-start mode, an interrupt (IWUNI) signal is output when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt or a maskable interrupt. For details, refer to section 15, Interrupt Controller (ICUE).

Table 33.4 IWDT Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
IWUNI	Counter underflow Refresh error	Not possible	Not possible

33.3.7 Reading the Counter Value

As the counter in IWDT-dedicated clock (IWDTCLK), the counter value cannot be read directly. The IWDT synchronizes the counter value with the peripheral module clock (PCLK) and stores it in the IWDTSR.CNTVAL[13:0] bits. Thus, the counter value can be checked indirectly through the IWDTSR.CNTVAL[13:0] bits.
Reading the counter value requires multiple PCLK clock cycles (up to four clock cycles), and the read counter value may differ from the actual counter value by a value of one count.
Figure 33.7 shows the processing for reading the IWDT counter value when PCLK > IWDTCLK and clock divide ratio = IWDTCLK.

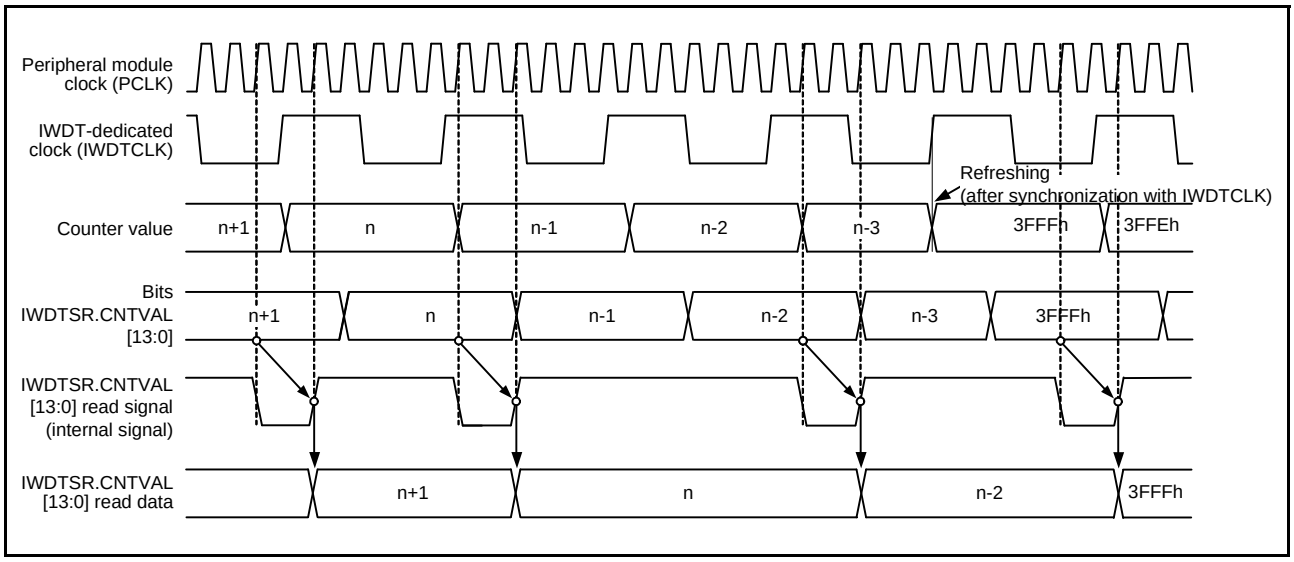


Figure 33.7 Processing for Reading IWDT Counter Value (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

33.3.8 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Table 33.5 lists the correspondence between option function select register 0 (OFS0) used in auto-start mode and the registers used in register start mode.
Do not change the OFS0 register setting during IWDT operation.
For details on the OFS0 register, refer to section 7.2.3, Option Function Select Register 0 (OFS0).

Table 33.5 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Target of Control	Function	OFS0 Register (Enabled in Auto-Start Mode) OFS0.IWDTSTRT = 0	IWDT Registers (Enabled in Register Start Mode) OFS0.IWDTSTRT = 1
Counter	Timeout period selection	OFS0.IWDTTOPS[1:0]	IWDTCR.TOPS[1:0]
	Clock frequency divide ratio selection	OFS0.IWDTCKS[3:0]	IWDTCR.CKS[3:0]
	Window start position selection	OFS0.IWDTRPSS[1:0]	IWDTCR.RPSS[1:0]
	Window end position selection	OFS0.IWDTRPES[1:0]	IWDTCR.RPES[1:0]
Reset output or interrupt request output	Reset output or interrupt request output selection	OFS0.IWDTRSTIRQS	IWDTRCR.RSTIRQS
Count stop	Sleep mode count stop control	OFS0.IWDTSLCSTP	IWDTCSTPR.SLCSTP

33.4 Link Operation by ELC

The event link controller (ELC) can use the interrupt request signal generated by the IWDT as the event signal. Therefore, the ELC generates an event to the module specified previously when the IWDT outputs an interrupt request. The event signal is output by the counter underflow and refresh error.
An event signal is output regardless of the setting of the IWDTRCR.RSTIRQS bit in register start mode or the OFS0.IWDTRSTIRQS bit in auto-start mode. An event signal can also be output upon generation of the next interrupt source while the IWDTSR.REFEF or IWDTSR.UNDFF flag is 1.
For details, see section 21, Event Link Controller (ELC).

33.5 Usage Notes

33.5.1 Refresh Operations

When making the settings to control the timing of refreshing, consider variations in the range of errors due to the accuracy of the PCLK and IWDTCLK and set values which ensure that refreshing is possible.

33.5.2 Clock Divide Ratio Setting

Satisfy the frequency of the peripheral module clock (PCLK) ≥ 4 × (the frequency of the count source after divide).

34. USB 2.0 FS Host/Function Module (USBb)

34.1 Overview

This MCU incorporates two channels of USB 2.0 FS host/function module (USB0, USB1).
The USB module is a USB controller that is equipped to operate as a host controller or function controller. The module supports full-speed and low-speed (only for the host controller) transfer as defined in Universal Serial Bus (USB) Specification 2.0. The module has an internal USB transceiver and supports all of the transfer types defined in USB Specification 2.0.
The USB has buffer memory for data transfer, providing a maximum of 10 pipes. Pipes 1 to 9 can be assigned any endpoint number based on peripheral devices used for communication or based on the user system.
Table 34.1 shows the specifications of the USB.

Table 34.1 USB Specifications

Item	Specifications
Features	• USB Device Controller (UDC) and transceiver for USB 2.0 are incorporated. Host controller, function controller, and On-The-Go (OTG) are supported (two channels) • The host controller and the function controller can be switched by software. • Self-power mode or bus power mode can be selected.
	When the host controller is selected: • Full-speed transfer (12 Mbps) and low-speed transfer (1.5 Mbps) are supported • Automatic scheduling for SOF and packet transmissions • Programmable intervals for isochronous and interrupt transfers • Multiple peripheral devices can be connected for communication via a one-stage hub.
	When the function controller is selected: • Full-speed transfer (12 Mbps)*1 is supported • Control transfer stage control function • Device state control function • Auto response function for SET_ADDRESS request • SOF interpolation function
Communication data transfer type	• Control transfer • Bulk transfer • Interrupt transfer • Isochronous transfer
Pipe configuration	• Buffer memory for USB communication is provided. • Up to 10 pipes can be selected (including the default control pipe). • Pipes 1 to 9 can be assigned any endpoint number.
	Transfer conditions that can be set for each pipe: • Pipe 0: Control transfer, 64-byte single buffer • Pipes 1 and 2: 64-byte double buffer can be specified for bulk transfer 256-byte double buffer for isochronous transfer • Pipes 3 to 5: Bulk transfer, 64-byte double buffer • Pipes 6 to 9: Interrupt transfer, 64-byte single buffer
Others	• Reception ending function using transaction count • Function that changes the BRDY interrupt event notification timing (BFRE) • Function that automatically clears the buffer memory after the data for the pipe specified at the DnFIFO (n = 0, 1) port has been read (DCLRM) • NAK setting function for response PID generated by end of transfer (SHTNAK) • On-chip pull-up and pull-down resistors of D+/D–
Low power consumption function	Module-stop state can be set.

Note 1. When the function controller is selected, low-speed transfer (1.5 Mbps) is not supported.

Figure 34.1 shows a block diagram of the USB.

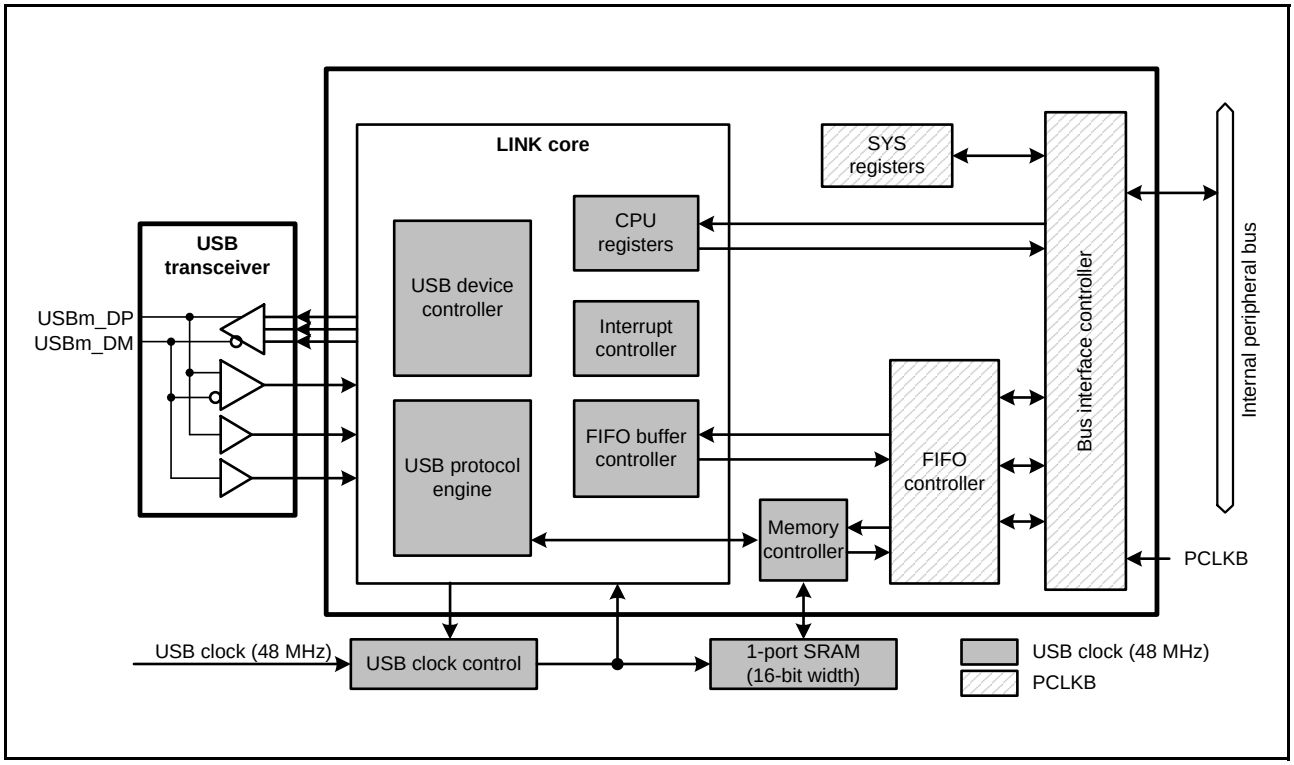


Figure 34.1 USB Block Diagram (m = 0, 1)

Table 34.2 lists the I/O pins of the USB.

Table 34.2 USB Pin Configuration (1/2)

Port	Pin Name	I/O	Function
USB0	USB0_DP	I/O	D+ I/O pin of the USB on-chip transceiver This pin should be connected to the D+ pin of the USB bus.
	USB0_DM	I/O	D– I/O pin of the USB on-chip transceiver This pin should be connected to the D– pin of the USB bus.
	USB0_VBUS	Input	USB cable connection monitor pin This pin should be connected to VBUS of the USB bus. The VBUS pin status (connected or disconnected) can be detected when the USB module is operating as a function controller.
	USB0_EXICEN	Output	Low-power control signal for external power supply (OTG) chip
	USB0_VBUSEN	Output	VBUS (5 V) supply enable signal for external power supply chip
	USB0_OVRCURA USB0_OVRCURB	Input	External overcurrent detection signals should be connected to these pins. VBUS comparator signals should be connected to these pins when the OTG power supply chip is connected.
	USB0_ID	Input	MicroAB connector ID input signal should be connected to this pin during operation in OTG mode.

Table 34.2 USB Pin Configuration (2/2)

Port	Pin Name	I/O	Function
USB1	USB1_DP	I/O	D+ I/O pin of the USB on-chip transceiver This pin should be connected to the D+ pin of the USB bus.
	USB1_DM	I/O	D– I/O pin of the USB on-chip transceiver This pin should be connected to the D– pin of the USB bus.
	USB1_VBUS	Input	USB cable connection monitor pin This pin should be connected to VBUS of the USB bus. The VBUS pin status (connected or disconnected) can be detected when the USB module is operating as a function controller.
	USB1_EXICEN	Output	Low-power control signal for external power supply (OTG) chip
	USB1_VBUSEN	Output	VBUS (5 V) supply enable signal for external power supply chip
	USB1_OVRCURA USB1_OVRCURB	Input	External overcurrent detection signals should be connected to these pins. VBUS comparator signals should be connected to these pins when the OTG power supply chip is connected.
	USB1_ID	Input	MicroAB connector ID input signal should be connected to this pin during operation in OTG mode.
Common	VCC_USB	Input	USB power supply pin
	VSS_USB	Input	USB ground pin

34.2 Register Descriptions

34.2.1 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	DRPD	DPRPU	—	—	—	USBE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	USBE	USB Operation Enable	0: USB operation is disabled. 1: USB operation is enabled.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DPRPU	D+ Line Resistor Control	0: Pulling up the line is disabled. 1: Pulling up the line is enabled.	R/W
b5	DRPD	D+/D– Line Resistor Control	0: Pulling down the lines is disabled. 1: Pulling down the lines is enabled.	R/W
b6	DCFM	Controller Function Select	0: Function controller is selected. 1: Host controller is selected.	R/W
b9 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10	SCKE	USB Clock Enable*1	0: Stops supplying the clock signal to the USB. 1: Enables supplying the clock signal to the USB.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. After writing 1 to the SCKE bit, read it and confirm it is set to 1.

USBE Bit (USB Operation Enable)

The USBE bit enables or disables operation of the USB.

Modifying the USBE bit from 1 to 0 initializes the register bits listed in Table 34.3.

This bit should be modified while the SCKE bit is 1.

When the host controller is selected, this bit should be set to 1 after setting the DRPD bit to 1, eliminating SYSSTS0.LNST[1:0] flag chattering, and confirming that the USB bus state is stabilized.

Table 34.3 Registers Initialized by Writing 0 to the SYSCFG.USBE Bit

Selected Function	Register	Bit	Remarks
Function controller	SYSSTS0	LNST[1:0]	The value is retained when the host controller is selected.
	DVSTCTR0	RHST[2:0]	
	INTSTS0	DVSQ[2:0]	The value is retained when the host controller is selected.
	USBADDR	USBADDR[6:0]	The value is retained when the host controller is selected.
	USBREQ	BREQUEST[7:0], BMREQUESTTYPE[7:0]	The value is retained when the host controller is selected.
	USBVAL	—	The value is retained when the host controller is selected.
	USBINDX	—	The value is retained when the host controller is selected.
	USBLENG	—	The value is retained when the host controller is selected.
Host controller	DVSTCTR0	RHST[2:0]	
	FRMNUM	FRNM[10:0]	The value is retained when the function controller is selected.

DPRPU Bit (D+ Line Resistor Control)

The DPRPU bit enables or disables pulling up the D+ line when the function controller is selected.
When the DPRPU bit is set to 1 while the function controller is selected, the bit forces a pull-up of the D+ line to notify the USB host of connection. Modifying the DPRPU bit from 1 to 0 allows the USB to release the D+ line, thus notifying the USB host of disconnection.
This bit should be set to 1 if the function controller is selected, and should be set to 0 if the host controller is selected.

DRPD Bit (D+/D– Line Resistor Control)

The DRPD bit enables or disables pulling down D+ and D– lines when the host controller is selected.
This bit should be set to 1 if the host controller is selected, and should be set to 0 if the function controller is selected.

DCFM Bit (Controller Function Select)

The DCFM bit selects the function of the USB.
This bit should be modified when the DPRPU and DRPD bits are all 0.

SCKE Bit (USB Clock Enable)

The SCKE bit stops or enables supplying 48-MHz clock signals to the USB.
When this bit is 0, only SYSCFG can be read from and written to; the other registers related to the USB cannot be read from or written to.

34.2.2 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
	OVCMON[1:0]		—	—	—	—	—	—	—	HTACT	SOFEA	—	—	IDMON	LNST[1:0]	
Value after reset:	0*1	0*1	0	0	0	0	0	0	0	0	0	0	0	0*1	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LNST[1:0]	USB Data Line Status Monitor Flag	Refer to Table 34.4.	R
b2	IDMON	ID Input Pin Monitor Flag	0: USBm_ID pin is low (m = 0, 1) 1: USBm_ID pin is high	R
b4, b3	—	Reserved	These bits are read as 0 and cannot be modified.	R
b5	SOFEA	SOF Active Monitor Flag When the Host Controller is Selected	Indicates the SOF output status. 0: SOF output is stopped. 1: SOF output is operating.	R
b6	HTACT	USB Host Sequencer Status Monitor Flag	0: Host sequencer of the USB is completely stopped. 1: Host sequencer of the USB is not completely stopped.	R
b13 to b7	—	Reserved	These bits are read as 0 and cannot be modified.	R
b15, b14	OVCMON[1:0]	OVRCURA/OVRCURB Input Pin Monitor Flag	The OVCMON[1] flag indicates the status of the USBm_OVRCURA pin (m = 0, 1). The OVCMON[0] flag indicates the status of the USBm_OVRCURB pin.	R

Note 1. Depends on the status of the USBm_OVRCURA/USBm_OVRCURB and USBm_ID pins (m = 0, 1).

LNST[1:0] Flags (USB Data Line Status Monitor Flag)

The LNST[1:0] flags indicate the state of the USB data lines (D+ and D– lines). Refer to Table 34.4.

The LNST[1:0] flags should be read after the connection processing (SYSCFG.DPRPU bit = 1) when the function controller is selected; whereas after enabling pull-down of the lines (SYSCFG.DRPD bit = 1) when the host controller is selected.

SOFEA Flag (SOF Active Monitor Flag When the Host Controller is Selected)

When the USB module is suspended while host mode is used, this bit can be used to confirm whether the last SOF has been output after DVSTCTR0.UACT is set to 0.

Confirm that both the HTACT and SOFEA flags are 0 before stopping the USB module while SYSCFG.USBE is 0 and setting the SYSCFG.SCKE bit to 0 to stop the clock during host mode communication.

HTACT Flag (USB Host Sequencer Status Monitor Flag)

The HTACT flag is 0 when the host sequencer of the USB is completely stopped.

Confirm that the HTACT flag is 0 before setting this controller to the USB suspended state while DVSTCTR0.UACT bit is 0 and setting the SCKE bit to 0 to stop the clock during host mode communication.

OVCMON[1:0] Flags (OVRCURA/OVRCURB Input Pin Monitor Flag)

The OVCMON[1:0] flags indicate the status of overcurrent from an external power supply chip.

Table 34.4 Status of USB Data Bus Lines (D+ Line, D– Line)

LNST[1:0] Flags	During Low-Speed Operation (Only in Host Controller Operation)	During Full-Speed Operation
00b	SE0	SE0
01b	K-State	J-State
10b	J-State	K-State
11b	SE1	SE1

34.2.3 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	EXICEN	VBUSEN	WKUP	RWUPE	USBRST	RESUME	UACT	—	RHST[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	RHST[2:0]	USB Bus Reset Status Flag	• When the host controller is selected b2 b0 0 0 0: Communication speed not determined (powered state or no connection) 1 x x: USB bus reset in progress 0 0 1: Low-speed connection 0 1 0: Full-speed connection • When the function controller is selected b2 b0 0 0 0: Communication speed not determined 0 0 1: USB bus reset in progress 0 1 0: USB bus reset in progress or full-speed connection	R
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	UACT	USB Bus Enable	0: Downstream port is disabled (SOF transmission is disabled). 1: Downstream port is enabled (SOF transmission is enabled).	R/W
b5	RESUME	Resume Output	0: Resume signal is not output. 1: Resume signal is output.	R/W
b6	USBRST	USB Bus Reset Output	0: USB bus reset signal is not output. 1: USB bus reset signal is output.	R/W
b7	RWUPE	Wakeup Detection Enable	0: Downstream port wakeup is disabled. 1: Downstream port wakeup is enabled.	R/W
b8	WKUP	Wakeup Output	0: Remote wakeup signal is not output. 1: Remote wakeup signal is output.	R/W
b9	VBUSEN	VBUSEN Output Pin Control	0: USBm_VBUSEN pin outputs low (m = 0, 1). 1: USBm_VBUSEN pin outputs high.	R/W
b10	EXICEN	EXICEN Output Pin Control	0: USBm_EXICEN pin outputs low (m = 0, 1). 1: USBm_EXICEN pin outputs high.	R/W
b11	HNPBTOA	Host Negotiation Protocol (HNP) Control	This bit is used when switching from device B to device A while in OTG mode. If the HNPBTOA bit is 1, the internal function control keeps the suspended state until the HNP processing ends even though SYSCFG.DPRPU = 0 or SYSCFG.DCFM = 1.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

x: Don't care

RHST[2:0] Flags (USB Bus Reset Status Flag)

The RHST[2:0] flags indicate the status of the USB bus reset.
When the host controller is selected, the RHST[2:0] flags indicate 100b after the USBRST bit has been set to 1 by software.
The USB fixes the value of the RHST[2:0] flags when 0 is written to the USBRST bit by software and the USB completes SE0 driving.
When the function controller is selected, the RHST[2:0] flags indicate 010b (connection while DPRPU = 1) when the USB detects the USB bus reset, and a DVST interrupt is generated.

UACT Bit (USB Bus Enable)

The UACT bit enables operation of the USB bus (controls the SOF packet transmission to the USB bus) when the host controller is selected.
With this bit set to 1, the USB puts the USB port to the USB-bus enabled state and performs SOF output and data transmission and reception.
This module starts outputting SOF packets within one frame after 1 has been written to the UACT bit by software.
With this bit set to 0, the USB enters the idle state after outputting SOF packets.
The USB sets the UACT bit to 0 on any of the following conditions.

- A DTCH interrupt is detected during communication (while UACT = 1).
- An EOFERR interrupt is detected during communication (while UACT = 1).

Writing 1 to this bit should be done at the end of the USB reset processing (writing 0 to the USBRST bit) or at the end of the resume processing from the suspended state (writing 0 to the RESUME bit).
This bit should be set to 0 if the function controller is selected.

RESUME Bit (Resume Output)

The RESUME bit controls the resume signal output when the host controller is selected.
Setting the RESUME bit to 1 allows the USB to drive the port to the K-state and output the resume signal.
The USB sets the RESUME bit to 1 on detecting the remote wakeup signal while RWUPE is 1 in the USB suspended state.
The USB continues outputting K-state while the RESUME bit = 1 (until the RESUME bit is set to 0 by software). The RESUME bit should be 1 (= resume period) for the time defined by USB Specification 2.0.
This bit should be set to 1 in the suspended state.
Write 1 to the UACT bit simultaneously with the end of the resume processing (writing 0 to the RESUME bit).
This bit should be set to 0 if the function controller is selected.

USBRST Bit (USB Bus Reset Output)

The USBRST bit controls the USB bus reset signal output when the host controller is selected.
When the host controller is selected, setting this bit to 1 allows the USB to drive SE0 of the USB port to reset the USB bus.
The USB continues outputting SE0 while USBRST = 1 (until the USBRST bit is set to 1 by software). The USBRST bit should be 1 (= USB bus reset period) for the time defined by USB Specification 2.0.
Writing 1 to this bit during communication (the UACT bit = 1) or during the resume processing (the RESUME bit = 1) prevents the USB from starting the USB bus reset processing until both the UACT and RESUME bits become 0.
Write 1 to the UACT bit simultaneously with the end of the USB bus reset processing (writing 0 to the USBRST bit).
This bit should be set to 0 if the function controller is selected.

RWUPE Bit (Wakeup Detection Enable)

The RWUPE bit enables or disables the downstream port peripheral device to use the remote wakeup function (resume signal output) when the host controller is selected.

With this bit set to 1, on detecting the remote wakeup signal, the USB detects the resume signal (K-state for 2.5 µs) from the downstream port device and performs the resume processing (drives the port to the K-state).

With this bit set to 0, the USB ignores the detected remote wakeup signal (K-state) from the peripheral device connected to the USB port.

While the RWUPE bit is 1, the internal clock should not be stopped even in the suspended state (SYSCFG.SCKE bit should be set to 1).

This bit should be set to 0 if the function controller is selected.

WKUP Bit (Wakeup Output)

The WKUP bit enables or disables outputting the remote wakeup signal (resume signal) to the USB bus when the function controller is selected.

The USB controls the output time of a remote wakeup signal. When this bit is set to 1, the USB sets this bit to 0 after outputting the 10-ms K-state.

According to USB Specification 2.0, the USB bus idle state must be kept for 5 ms or longer before a remote wakeup signal is sent. If the USB writes 1 to this bit right after detection of the suspended state, the K-state will be output after 2 ms.

Do not write 1 to this bit, unless the device state is in the suspended state (INTSTS0.DVSQ[2:0] flags = 1xxb) and the USB host enables the remote wakeup signal. When this bit is set to 1, the internal clock must not be stopped even in the suspended state (write 1 to this bit while the SYSCFG.SCKE bit = 1).

This bit should be set to 0 if the host controller is selected.

HNPBTOA Bit (Host Negotiation Protocol (HNP) Control)

The HNPBTOA bit is used when switching from device B to device A while in OTG mode. If the HNPBTOA bit is 1, the internal function control keeps the suspended state until the HNP processing ends even though the SYSCFG.DPRPU bit = 0 or SYSCFG.DCFM = 1 is set. Even if the falling edge of the D+ signal is detected at this time, no resume (RESM) interrupt is generated.

After this bit is set to 1, write 0 to this bit by software to terminate the HNP processing when connection to the host (pull-up on the target side) or timeout of the HNP processing is detected.

34.2.4 CFIFO Port Register (CFIFO), D0FIFO Port Register (D0FIFO), D1FIFO Port Register (D1FIFO)

(1) When the MBW bit is 1

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
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	FIFO Port	This port is used for reading receive data from the FIFO buffer and writing transmit data to the FIFO buffer.	R/W

(2) When the MBW bit is 0

Address(es): USB0.CFIFO.L 000A 0014h, USB0.D0FIFO.L 000A 0018h, USB0.D1FIFO.L 000A 001Ch, USB1.CFIFO.L 000A 0214h, USB1.D0FIFO.L 000A 0218h, USB1.D1FIFO.L 000A 021Ch

	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
b7 to b0	—	FIFO Port	This port is used for reading receive data from the FIFO buffer and writing transmit data to the FIFO buffer.	R/W

There are three FIFO ports: CFIFO, D0FIFO, and D1FIFO. Each FIFO port is configured of a port register (CFIFO, D0FIFO, or D1FIFO) that handles reading of data from the FIFO buffer memory and writing of data to the FIFO buffer memory, a port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) that is used to select the pipe assigned to the FIFO port, and a port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR).
Each FIFO port has the following precautions.

- The FIFO buffer for DCP (control transfer) should be accessed through the CFIFO port.
- Accessing a FIFO buffer using DMA/DTC transfer should be performed through the D0FIFO or D1FIFO port.
- The D0FIFO and D1FIFO ports can also be accessed by the CPU.
- When using functions specific to a FIFO port, the pipe number (selected pipe) specified by the CURPIPE[3:0] bits in the port select register cannot be changed (when the DMA/DTC transfer function is used, etc.).
- The same pipe should not be assigned to two or more FIFO ports.
- There are two FIFO buffer states: the access right for a FIFO buffer can be on the CPU side or on the Serial Interface Engine (SIE) side. When the access right for a FIFO buffer is on the SIE side, the FIFO buffer cannot be accessed from the CPU.

FIFO Port Bit

Accessing the FIFO port bits allows reading the received data from the FIFO buffer or writing the transmit data to the FIFO buffer.

Each FIFO port register can be accessed only while the FRDY flag in each FIFO port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1.

The valid bits in a FIFO port register depend on the settings of the corresponding MBW bit of the FIFO port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL).

When the MBW bit is 1 (16-bit width), the data arrangement may differ from the data arrangement on the RAM depending on the value of the MDE.MDE[2:0] bits and the setting of the BIGEND bit (CFIFOSEL.BIGEND, D0FIFOSEL.BIGEND, or D1FIFOSEL.BIGEND). Table 34.5 lists the endian operation in 16-bit access. Note that if the total number of transmit data bytes is odd, access the lower byte in bytes when writing the last data.

When the MBW bit is 0 (8-bit width), access the lower byte in bytes.

Table 34.5 Endian Operation in 16-Bit Access

MDE.MDE[2:0] bits	CFIFOSEL.BIGEND Bit D0FIFOSEL.BIGEND Bit D1FIFOSEL.BIGEND Bit	Bits 15 to 8	Bits 7 to 0	Remarks
000b (big endian)	0 (little endian)	Data in address N + 1	Data in address N	Bytes reversed
	1 (big endian)	Data in address N	Data in address N + 1	
111b (little endian)	0 (little endian)	Data in address N + 1	Data in address N	
	1 (big endian)	Data in address N	Data in address N + 1	Bytes reversed

34.2.5 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 Settings other than above are prohibited.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	ISEL	CFIFO Port Access Direction When DCP is Selected	0: Reading from the buffer memory is selected. 1: Writing to the buffer memory is selected.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BIGEND	CFIFO Port Endian Control	0: Little endian 1: Big endian	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	MBW	CFIFO Port Access Bit Width	0: 8-bit width 1: 16-bit width	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	REW	Buffer Pointer Rewind	0: The buffer pointer is not rewound. 1: The buffer pointer is rewound.	R/W*1
b15	RCNT	Read Count Mode	0: The DTLN[8:0] flags (CFIFOCTR.DTLN[8:0], D0FIFOCTR.DTLN[8:0], D1FIFOCTR.DTLN[8:0]) are cleared when all of the receive data has been read from the CFIFO. (In double buffer mode, the DTLN[8:0] bit value is cleared when all the data has been read from only a single plane.) 1: The DTLN[8:0] flags are decremented each time the receive data is read from the CFIFO.	R/W

Note 1. Only 0 can be read.

The same pipe should not be specified by the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected.

The pipe number should not be changed while the DMA/DTC transfer is enabled.

CURPIPE[3:0] Bits (CFIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number using which data is read or written through the CFIFO port.
After writing to these bits, read these bits to check that the written value agrees with the read value before proceeding to the next process.
Do not set the same pipe number to the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers.
Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective, thus enabling continuous access.

ISEL Bit (CFIFO Port Access Direction When DCP is Selected)

After writing to the ISEL bit with the DCP being a selected pipe, read this bit to check that the written value agrees with the read value before proceeding to the next process.
Set this bit and the CURPIPE[3:0] bits simultaneously.

MBW Bit (CFIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the CFIFO port.
When the selected pipe is in the receiving direction, once reading data is started after setting this bit, this bit should not be modified until all the data has been read.
When the selected pipe is in the receiving direction, set the CURPIPE[3:0] bits to a different value once, and then set these bits and the MBW bit simultaneously. For the procedure for modifying the CURPIPE[3:0] bits, follow the description of these bits.
When the selected pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit width while data is being written to the buffer memory.
An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

REW Bit (Buffer Pointer Rewind)

The REW bit specifies whether or not to rewind the buffer pointer.
When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed).
Do not set the REW bit to 1 simultaneously with modifying the CURPIPE[3:0] bits. Before setting the REW bit to 1, be sure to check that the FRDY flag is 1.
To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.

- D0FIFOSEL, D1FIFOSEL

Address(es): 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]	FIFO Port Access Pipe Select	b3 b0 0 0 0 0: DCP (Default control pipe) 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Settings other than above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BIGEND	FIFO Port Endian Control	0: Little endian 1: Big endian	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	MBW	FIFO Port Access Bit Width	0: 8-bit width 1: 16-bit width	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	DREQE	DMA/DTC Transfer Request Enable	0: DMA/DTC transfer request is disabled. 1: DMA/DTC transfer request is enabled.	R/W
b13	DCLRM	Automatic FIFO Buffer Memory Clear Mode after Selected Pipe is Read	0: Automatic buffer clear mode is disabled. 1: Automatic buffer clear mode is enabled.	R/W
b14	REW	Buffer Pointer Rewind	0: The buffer pointer is not rewound. 1: The buffer pointer is rewound.	R/W*1
b15	RCNT	Read Count Mode	0: The DTLN[8:0] flags (CFIFOCTR.DTLN[8:0], D0FIFOCTR.DTLN[8:0], D1FIFOCTR.DTLN[8:0]) are cleared when all of the receive data has been read from the DnFIFO. (In double buffer mode, the DTLN bit Value is cleared when all the data has been read from only a single plane.) 1: The DTLN[8:0] flags are decremented each time the receive data is read from the DnFIFO. (n = 0, 1)	R/W

Note 1. Only 0 can be read.

The same pipe should not be specified by the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected.

The pipe number should not be changed while the DMA/DTC transfer is enabled.

CURPIPE[3:0] Bits (FIFO Port Access Pipe Select)

The CURPIPE[3:0] bits specify the pipe number using which data is read or written through the D0FIFO port or D1FIFO port.
After writing to these bits, read these bits to check that the written value agrees with the read value before proceeding to the next process.
Do not set the same pipe number to the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers.
Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective, thus enabling continuous access.

MBW Bit (FIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the D0FIFO port or D1FIFO port.
When the selected pipe is in the receiving direction, once reading data is started after setting this bit, this bit should not be modified until all the data has been read.
When the selected pipe is in the receiving direction, set the CURPIPE[3:0] bits to a different value once, and then set these bits and the MBW bit simultaneously. For the procedure for modifying the CURPIPE[3:0] bits, follow the description of these bits.
When the selected pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit width while data is being written to the buffer memory.
An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

DREQE Bit (DMA/DTC Transfer Request Enable)

The DREQE bit enables or disables the DMA/DTC transfer request to be issued.
Before setting the DREQE bit to 1 to enable the DMA/DTC transfer request to be issued, set the CURPIPE[3:0] bits.
When modifying the setting of the CURPIPE[3:0] bits, set this bit to 0 first.

DCLRM Bit (Automatic FIFO Buffer Memory Clear Mode after Selected Pipe is Read)

The DCLRM bit enables or disables the buffer memory to be cleared automatically after data in the selected pipe has been read.
With this bit set to 1, the USB sets the BCLR bit to 1 for the FIFO buffer of the selected pipe on receiving a zero-length packet while the FIFO buffer assigned to the selected pipe is empty, or on receiving a short packet and reading the data while the PIPECFG.BFRE bit is 1.
When using the USB with the SOFCFG.BRDYM bit set to 1, set this bit to 0.

REW Bit (Buffer Pointer Rewind)

The REW bit specifies whether or not to rewind the buffer pointer.
When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed).
Do not set the REW bit to 1 simultaneously with modifying the CURPIPE[3:0] bits. Before setting the REW bit to 1, be sure to check that the FRDY flag is 1.
To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.

RCNT Bit (Read Count Mode)

The RCNT bit specifies the read mode for the value in the CFIFOCTR.DTLN bit.
When accessing DnFIFO with the PIPECFG.BFRE bit set to 1, set the RCNT bit to 0.

34.2.6 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 Flag	Indicate the length of the receive data. These bits indicate different values depending on the setting of the RCNT bit in the port select register. For details, refer to the description on the DTLN[8:0] flags shown below.	R
b12 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13	FRDY	FIFO Port Ready Flag	0: FIFO port access is disabled. 1: FIFO port access is enabled.	R
b14	BCLR	CPU Buffer Clear	0: Does not operate. 1: Clears the buffer memory on the CPU side.	R/W*1
b15	BVAL	Buffer Memory Valid	0: Invalid 1: Writing ended	R/W

Note 1. Only 0 can be read.

Registers CFIFOCTR, D0FIFOCTR, and D1FIFOCTR correspond to CFIFO, D0FIFO, and D1FIFO, respectively.

DTLN[8:0] Flags (Receive Data Length Flag)

The DTLN[8:0] flags indicate the length of the receive data.

While the FIFO buffer is being read, the DTLN[8:0] flags indicate different values depending on the DnFIFOSEL.RCNT bit (n = 0, 1) value as described below.

- RCNT = 0
  The USB sets the DTLN[8:0] flags to indicate the length of the receive data until the CPU or DMAC/DTC has read all the received data from a single FIFO buffer plane.
  While the PIPECFG.BFRE bit = 1, these bits retain the length of the receive data until the BCLR bit is set to 1 even after all the data has been read.
- RCNT = 1
  The USB decrements the value indicated by the DTLN[8:0] flags each time data is read from the FIFO buffer. (The value is decremented by one when the MBW bit is 0, and by two when the MBW bit is 1.)
  The USB sets these bits to 0 when all the data has been read from one FIFO buffer plane. However, in double buffer mode, if data has been received in one FIFO buffer plane before all the data has been read from the other plane, the USB sets these bits to indicate the length of the receive data in the former plane when all the data has been read from the latter plane.

FRDY Flag (FIFO Port Ready Flag)

The FRDY flag indicates whether the FIFO port can be accessed by the CPU or DMAC/DTC.
In the following cases, the USB sets the FRDY flag to 1 but data cannot be read via the FIFO port because there is no data to be read. In these cases, set the BCLR bit to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.

- A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty.
- A short packet is received and the data is completely read while the PIPECFG.BFRE bit = 1.

BCLR Bit (CPU Buffer Clear)

The BCLR bit should be set to 1 to clear the FIFO buffer on the CPU side for the selected pipe.
When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USB clears only the FIFO buffer on the CPU side even when both planes are read-enabled.
When the selected pipe is the DCP, setting the BCLR bit to 1 allows the USB to clear the FIFO buffer regardless of whether the FIFO buffer is on the CPU side or SIE side. To clear the buffer on the SIE side, set the DCPCTR.PID[1:0] bits for the DCP to 00b (NAK) before setting the BCLR bit to 1.
When the selected pipe is in the transmitting direction, if 1 is written to the BVAL bit and the BCLR bit simultaneously, the USB clears the data that has been written before it, enabling transmission of a zero-length packet.
When the selected pipe is not the DCP, writing 1 to the BCLR bit should be done while the FRDY flag in the FIFO port control register is 1 (set by the USB).

BVAL Bit (Buffer Memory Valid)

The BVAL bit should be set to 1 when data has been completely written to the FIFO buffer on the CPU side for the pipe selected using the CURPIPE[3:0] bits (selected pipe).
When the selected pipe is in the transmitting direction, set the BVAL bit to 1 in the following cases. Then, the USB switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.

- To transmit a short packet, set the BVAL bit to 1 after data has been written.
- To transmit a zero-length packet, set the BVAL bit to 1 before data is written to the FIFO buffer.

When data of the maximum packet size has been written for the pipe in continuous transfer mode, the USB sets the BVAL bit to 1 and switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.
Writing 1 to the BVAL bit should be done while the FRDY flag is 1 (set by the USB).
When the selected pipe is in the receiving direction, do not set the BVAL bit to 1.

34.2.7 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 read as 0. The write value should be 0.	R/W
b8	BRDYE	Buffer Ready Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	NRDYE	Buffer Not Ready Response Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b10	BEMPE	Buffer Empty Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b11	CTRE	Control Transfer Stage Transition Interrupt Enable *1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b12	DVSE	Device State Transition Interrupt Enable*1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b13	SOFE	Frame Number Update Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b14	RSME	Resume Interrupt Enable*1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15	VBSE	VBUS Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W

Note 1. The RSME, DVSE, and CTRE bits can be set to 1 only when the function controller is selected. Set these bits to 0 when the host controller is selected.

On detecting the interrupt corresponding to the bit in the INTENB0 register to which 1 has been set by software, the USB generates the USB interrupt request.

The USB sets 1 to each status bit in the INTSTS0 register when a detection condition of the corresponding interrupt source has been satisfied regardless of the setting in the INTENB0 register (regardless of whether the interrupt output is enabled or disabled).

While the status bit in the INTSTS0 register corresponding to the interrupt source indicates 1, the USB generates the USB interrupt request when the corresponding interrupt enable bit in the INTENB0 register is modified from 0 to 1 by software.

34.2.8 Interrupt Enable Register 1 (INTENB1)

Address(es): USB0.INTENB1 000A 0032h, USB1.INTENB1 000A 0232h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRCRE	BCHGE	—	DTCHE	ATTCHE	—	—	—	—	EOFERRE	SIGNE	SACKE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SACKE	Setup Transaction Normal Response Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	SIGNE	Setup Transaction Error Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	EOFERRE	EOF Error Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b10 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	ATTCHE	Connection Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b12	DTCHE	Disconnection Detection Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	BCHGE	USB Bus Change Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15	OVRCRE	Overcurrent Input Change Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W

Note: The bits in the INTENB1 register can be set to 1 only when the host controller is selected. Set these bits to 0 when the function controller is selected.

The INTENB1 register specifies the interrupt masks when the host controller is selected, and for the setup transaction. On detecting the interrupt corresponding to the bit in the INTENB1 register to which 1 has been set by software, the USB generates the USB interrupt request.

The USB sets 1 to each status bit in the INTSTS1 register when a detection condition of the corresponding interrupt source has been satisfied regardless of the setting in the INTENB1 register (regardless of whether the interrupt output is enabled or disabled).

While the status bit in the INTSTS1 register corresponding to the interrupt source indicates 1, the USB generates the USB interrupt request when the corresponding interrupt enable bit in the INTENB1 register is modified from 0 to 1 by software.

When the function controller is selected, the interrupts should not be enabled.

34.2.9 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 Pipe 0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1BRDYE	BRDY Interrupt Enable for Pipe 1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2BRDYE	BRDY Interrupt Enable for Pipe 2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3BRDYE	BRDY Interrupt Enable for Pipe 3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4BRDYE	BRDY Interrupt Enable for Pipe 4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5BRDYE	BRDY Interrupt Enable for Pipe 5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6BRDYE	BRDY Interrupt Enable for Pipe 6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7BRDYE	BRDY Interrupt Enable for Pipe 7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8BRDYE	BRDY Interrupt Enable for Pipe 8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9BRDYE	BRDY Interrupt Enable for Pipe 9	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The BRDYENB register enables or disables the INTSTS0.BRDY flag to be set to 1 when the BRDY interrupt is detected for each pipe.

On detecting the BRDY interrupt for the pipe corresponding to the bit in the BRDYENB register to which 1 has been set by software, the USB sets 1 to the corresponding BRDYSTS.PIPEnBRDY flag (n = 0 to 9) and the INTSTS0.BRDY flag. If INTENB0.BRDYE = 1 at this time, the USB generates the BRDY interrupt request.

While at least one PIPEnBRDY flag indicates 1, the USB generates the BRDY interrupt request when the corresponding interrupt enable bit in the BRDYENB register is modified from 0 to 1 by software.

34.2.10 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 Pipe 0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1NRDYE	NRDY Interrupt Enable for Pipe 1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2NRDYE	NRDY Interrupt Enable for Pipe 2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3NRDYE	NRDY Interrupt Enable for Pipe 3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4NRDYE	NRDY Interrupt Enable for Pipe 4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5NRDYE	NRDY Interrupt Enable for Pipe 5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6NRDYE	NRDY Interrupt Enable for Pipe 6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7NRDYE	NRDY Interrupt Enable for Pipe 7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8NRDYE	NRDY Interrupt Enable for Pipe 8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9NRDYE	NRDY Interrupt Enable for Pipe 9	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The NRDYENB register enables or disables the INTSTS0.NRDY flag to be set to 1 when the NRDY interrupt is detected for each pipe.

On detecting the NRDY interrupt for the pipe corresponding to the bit in the NRDYENB register to which 1 has been set by software, the USB sets 1 to the corresponding NRDYSTS.PIPEnNRDY flag (n = 0 to 9) and the INTSTS0.NRDY flag. If INTENB0.NRDYE = 1 at this time, the USB generates the NRDY interrupt request.

While at least one PIPEnNRDY flag indicates 1, the USB generates the NRDY interrupt request when the corresponding interrupt enable bit in the NRDYENB register is modified from 0 to 1 by software.

34.2.11 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 Pipe 0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1BEMPE	BEMP Interrupt Enable for Pipe 1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2BEMPE	BEMP Interrupt Enable for Pipe 2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3BEMPE	BEMP Interrupt Enable for Pipe 3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4BEMPE	BEMP Interrupt Enable for Pipe 4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5BEMPE	BEMP Interrupt Enable for Pipe 5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6BEMPE	BEMP Interrupt Enable for Pipe 6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7BEMPE	BEMP Interrupt Enable for Pipe 7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8BEMPE	BEMP Interrupt Enable for Pipe 8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9BEMPE	BEMP Interrupt Enable for Pipe 9	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The BEMPENB register enables or disables the INTSTS0.BEMP flag to be set to 1 when the BEMP interrupt is detected for each pipe.

On detecting the BEMP interrupt for the pipe corresponding to the bit in the BEMPENB register to which 1 has been set by software, the USB sets 1 to the corresponding BEMPSTS.PIPEnBEMP flag (n = 0 to 9) and the INTSTS0.BEMP flag. If INTENB0.BEMPE = 1 at this time, the USB generates the BEMP interrupt request.

While at least one PIPEnBEMP flag in the BEMPSTS register indicates 1, the USB generates the BEMP interrupt request when the corresponding interrupt enable bit in the BEMPENB register is modified from 0 to 1 by software.

34.2.12 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	—	BRDYM	—	EDGESTS	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	EDGESTS	Edge Interrupt Output Status Monitor Flag*1	Indicates 1 when the edge interrupt output signal is in the middle of the edge processing.	R
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	BRDYM	BRDY Interrupt Status Clear Timing	0: Software clears the status. 1: The USB clears the status when data has been read from the FIFO buffer or data has been written to the FIFO buffer.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	TRNENSEL	Transaction-Enabled Time Select*1	0: For non-low-speed communication 1: For low-speed communication	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Confirm that this bit is 0 before stopping the clock supply to the USB module.

EDGESTS Flag (Edge Interrupt Output Status Monitor Flag)

The EDGESTS flag indicates 1 when the edge interrupt output signal is in the middle of the edge processing.
Confirm that this flag is 0 before stopping the clock supply to the USB.

BRDYM Bit (BRDY Interrupt Status Clear Timing)

The BRDYM bit specifies the timing for clearing the BRDY interrupt status for each pipe.

TRNENSEL Bit (Transaction-Enabled Time Select)

The TRNENSEL bit selects, for full-speed or low-speed communication, the transaction-enabled time in which the USB issues tokens in a frame via the port.
Set the TRNENSEL bit to 1 when a low-speed device is connected.
The TRNENSEL bit is valid only when the host controller is selected.
This bit should be set to 0 if the function controller is selected.

34.2.13 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*3	0*3	0/1*3	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CTSQ[2:0]	Control Transfer Stage Flag	b2 b0 0 0 0: Idle or setup stage 0 0 1: Control read data stage 0 1 0: Control read status stage 0 1 1: Control write data stage 1 0 0: Control write status stage 1 0 1: Control write (no data) status stage 1 1 0: Control transfer sequence error	R
b3	VALID	USB Request Reception Flag	0: Setup packet is not received. 1: Setup packet is received.	R/W
b6 to b4	DVSQ[2:0]	Device State Flag	b6 b4 0 0 0: Powered state 0 0 1: Default state 0 1 0: Address state 0 1 1: Configured state 1 x x: Suspended state	R
b7	VBSTS	VBUS Input Status Flag	0: USBm_VBUS pin is low (m = 0, 1). 1: USBm_VBUS pin is high.	R
b8	BRDY	Buffer Ready Interrupt Status Flag	0: BRDY interrupts are not generated. 1: BRDY interrupts are generated.	R
b9	NRDY	Buffer Not Ready Interrupt Status Flag	0: NRDY interrupts are not generated. 1: NRDY interrupts are generated.	R
b10	BEMP	Buffer Empty Interrupt Status Flag	0: BEMP interrupts are not generated. 1: BEMP interrupts are generated.	R
b11	CTRT	Control Transfer Stage Transition Interrupt Status Flag*5	0: Control transfer stage transition interrupts are not generated. 1: Control transfer stage transition interrupts are generated.	R/W*4
b12	DVST	Device State Transition Interrupt Status Flag*5	0: Device state transition interrupts are not generated. 1: Device state transition interrupts are generated.	R/W*4
b13	SOFR	Frame Number Refresh Interrupt Status Flag	0: SOF interrupts are not generated. 1: SOF interrupts are generated.	R/W*4
b14	RESM	Resume Interrupt Status Flag *5,*6	0: Resume interrupts are not generated. 1: Resume interrupts are generated.	R/W*4
b15	VBINT	VBUS Interrupt Status Flag*6	0: VBUS interrupts are not generated. 1: VBUS interrupts are generated.	R/W*4

x: Don't care

Note 1. The value is 0 when the MCU is reset and 1 after a USB bus reset.

Note 2. The value is 1 when the USBm_VBUS pin is high and 0 when the USBm_VBUS pin is low (m = 0, 1).

Note 3. The value is 000b when the MCU is reset and 001b after a USB bus reset.

Note 4. To clear the VBINT, RESM, SOFR, DVST, CTRT, or VALID flag, write 0 only to the flags to be cleared; write 1 to the other flags. Do not write 0 to the status flags indicating 0.

Note 5. The status of the RESM, DVST, and CTRT flags are changed only when the function controller is selected. Set the corresponding interrupt enable bits to 0 (disabled) when the host controller is selected.

Note 6. A change in the status indicated by the VBINT and RESM flags can be detected even while the clock supply is stopped (the SCKE bit = 0), and the interrupts are output when the corresponding interrupt enable bits are enabled. Clearing the status through software should be done after enabling the clock supply.

CTSQ[2:0] Flags (Control Transfer Stage Flag)

When the host controller is selected, the read value is invalid.

VALID Flag (USB Request Reception Flag)

When the host controller is selected, the read value is invalid.

DVSQ[2:0] Flags (Device State Flag)

The DVSQ[2:0] flags are initialized by a USB bus reset.
When the host controller is selected, the read value is invalid.

BRDY Flag (Buffer Ready Interrupt Status Flag)

Indicates the BRDY interrupt status.
The USB sets the BRDY flag to 1 when at least one PIPEnBRDY flag (n = 0 to 9) is set to 1 among the PIPEnBRDY flags. These bits correspond to the BRDYENB.PIPEnBRDYE bits (n = 0 to 9) to which 1 has been set, when the USB detects the BRDY interrupt status in at least one pipe among the pipes for which the BRDY interrupt output is enabled by software.
For the conditions for PIPEnBRDY status assertion, refer to section 34.3.3.1, BRDY Interrupt.
The USB sets the BRDY flag to 0 when 0 is written by software to all the PIPEnBRDY flags corresponding to the PIPEnBRDYE bits that have been set to 1.
The BRDY flag cannot be set to 0 even if 0 is written to this bit by software.

NRDY Flag (Buffer Not Ready Interrupt Status Flag)

The USB sets the NRDY flag to 1 when at least one PIPEnNRDY flag (n = 0 to 9) is set to 1 among the PIPEnNRDY flags corresponding to the PIPEnNRDYE bits (n = 0 to 9) to which 1 has been set (when the USB detects the NRDY interrupt status in at least one pipe among the pipes for which software enables the NRDY interrupt output).
For the conditions for PIPEnNRDY status assertion, refer to section 34.3.3.2, NRDY Interrupt.
The USB sets the NRDY flag to 0 when 0 is written by software to all the PIPEnNRDY flags corresponding to the PIPEnNRDYE bits that have been set to 1.
The NRDY flag cannot be set to 0 even if 0 is written to this bit by software.

BEMP Flag (Buffer Empty Interrupt Status Flag)

The USB sets the BEMP flag to 1 when at least one PIPEnBEMP flag (n = 0 to 9) is set to 1 among the PIPEnBEMP flags corresponding to the PIPEnBEMPE bits (n = 0 to 9) to which 1 has been set (when the USB detects the BEMP interrupt status in at least one pipe among the pipes for which the BEMP interrupt output is enabled by software).
For the conditions for PIPEnBEMP status assertion, refer to section 34.3.3.3, BEMP Interrupt.
The USB sets the BEMP flag to 0 when 0 is written by software to all the PIPEnBEMP flags corresponding to the PIPEnBEMPE bits that have been set to 1.
The BEMP flag cannot be set to 0 even if 0 is written to this bit by software.

CTRT Flag (Control Transfer Stage Transition Interrupt Status Flag)

When the function controller is selected, the USB updates the value of the CTSQ[2:0] flags and sets the CTRT flag to 1 on detecting a change in the control transfer stage.
When a control transfer stage transition interrupt is generated, clear the status before the USB detects the next control transfer stage transition.
When the host controller is selected, the read value is invalid.

DVST Flag (Device State Transition Interrupt Status Flag)

When the function controller is selected, the USB updates the DVSQ[2:0] value and sets the DVST flag to 1 on detecting a change in the device state.
When a device state transition interrupt is generated, clear the status before the USB detects the next device state transition.
When the host controller is selected, the read value is invalid.

SOFR Flag (Frame Number Refresh Interrupt Status Flag)

(1) When the host controller is selected
The USB sets the SOFR flag to 1 on updating the frame number when the DVSTCTR0.UACT bit has been set to 1 by software. (A frame number refresh interrupt is detected every 1 ms.)
(2) When the function controller is selected
The USB sets the SOFR flag to 1 on updating the frame number. (A frame number refresh interrupt is detected every 1 ms.)
The USB can detect an SOFR interrupt through the internal interpolation function even when a damaged SOF packet is received from the USB host.

RESM Flag (Resume Interrupt Status Flag)

When the function controller is selected, the USB sets the RESM flag to 1 on detecting the falling edge of the signal on the USBm_DP pin (m = 0, 1) in the suspended state (DVSQ[2:0] = 1xxb).
When the host controller is selected, the read value is invalid.

VBINT Flag (VBUS Interrupt Status Flag)

The USB sets the VBINT flag to 1 on detecting a level change (high to low or low to high) in the USBm_VBUS pin input value. The USB sets the VBSTS flag to indicate the USBm_VBUS pin input value. When the VBUS interrupt is generated, use software to repeat reading the VBSTS flag until the same value is read three or more times, and eliminate chattering.

34.2.14 Interrupt Status Register 1 (INTSTS1)

Address(es): USB0.INTSTS1 000A 0042h, USB1.INTSTS1 000A 0242h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OVRCR	BCHG	—	DTCH	ATTCH	—	—	—	—	EOFERR	SIGN	SACK	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SACK	Setup Transaction Normal Response Interrupt Status Flag	0: SACK interrupts are not generated. 1: SACK interrupts are generated.	R/W *1
b5	SIGN	Setup Transaction Error Interrupt Status Flag	0: SIGN interrupts are not generated. 1: SIGN interrupts are generated.	R/W *1
b6	EOFERR	EOF Error Detection Interrupt Status Flag	0: EOFERR interrupts are not generated. 1: EOFERR interrupts are generated.	R/W *1
b10 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	ATTCH	ATTCH Interrupt Status Flag	0: ATTCH interrupts are not generated. 1: ATTCH interrupts are generated.	R/W *1
b12	DTCH	USB Disconnection Detection Interrupt Status Flag	0: DTCH interrupts are not generated. 1: DTCH interrupts are generated.	R/W *1
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14	BCHG	USB Bus Change Interrupt Status Flag *2	0: BCHG interrupts are not generated. 1: BCHG interrupts are generated.	R/W *1
b15	OVRCR	Overcurrent Input Change Interrupt Status Flag *2	0: OVRCR interrupts are not generated. 1: OVRCR interrupts are generated.	R/W *1

Note 1. To clear the status indicated by the flags in the INTSTS1 register, write 0 only to the flags to be cleared; write 1 to the other flags.

Note 2. A change in the status indicated by the OVRCR or BCHG flag can be detected even while the clock supply is stopped (while the SYSCFG.SCKE bit = 0), and the interrupt is output when the corresponding interrupt enable bit is enabled. Clearing the status through software should be done after setting the SYSCFG.SCKE bit to 1.
No interrupts other than those indicated by the BCHG and OVRCR flags can be detected while the clock supply is stopped (while the SYSCFG.SCKE bit = 0).

The INTSTS1 register is used to confirm the status of each interrupt when the host controller is selected.
The various status change interrupts indicated by the flags in the INTSTS1 register should be enabled only when the host controller is selected.

SACK Flag (Setup Transaction Normal Response Interrupt Status Flag)

Indicates the status of the setup transaction normal response interrupt when the host controller is selected.
The USB detects the SACK interrupt when ACK response is returned from the peripheral device during the setup transactions issued by the USB, and sets the SACK flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the SACK interrupt.
When the function controller is selected, the read value is invalid.

SIGN Flag (Setup Transaction Error Interrupt Status Flag)

Indicates the status of the setup transaction error interrupt when the host controller is selected.
The USB detects the SIGN interrupt when ACK response is not returned from the peripheral device three consecutive times during the setup transactions issued by this module, and sets the SIGN flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the SIGN interrupt.
Specifically, the USB detects the SIGN interrupt when any of the following response conditions occur for three consecutive setup transactions.

- Timeout is detected by the USB when the peripheral device has returned no response.
- A damaged ACK packet is received.
- A handshake other than ACK (NAK, NYET, or STALL) is received.

When the function controller is selected, the read value is invalid.

EOFERR Flag (EOF Error Detection Interrupt Status Flag)

Indicates the status of the EOFERR interrupt when the host controller is selected.
The USB detects the EOFERR interrupt on detecting that communication is not completed at the EOF2 timing prescribed by USB Specification 2.0, and sets the EOFERR flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the EOFERR interrupt.
After detecting the EOFERR interrupt, the USB controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the USB port should be terminated by software and perform re-enumeration of the USB port.

- Modifies the DVSTCTR0.UACT bit for the port in which an EOFERR interrupt has been detected to 0.
- Puts the port in which an EOFERR interrupt has been generated into the idle state.

When the function controller is selected, the read value is invalid.

ATTCH Flag (ATTCH Interrupt Status Flag)

Indicates the status of the ATTCH interrupt when the host controller is selected.
The USB detects the ATTCH interrupt on detecting J-state or K-state of the full-speed or low-speed signal level for 2.5 µs, and sets the ATTCH flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.
Specifically, the USB detects the ATTCH interrupt on any of the following conditions.

- K-state, SE0, or SE1 changes to J-state, and J-state continues for 2.5 µs.
- J-state, SE0, or SE1 changes to K-state, and K-state continues for 2.5 µs.

When the function controller is selected, the read value is invalid.

DTCH Flag (USB Disconnection Detection Interrupt Status Flag)

Indicates the status of the USB disconnection detection interrupt when the host controller is selected.
The USB detects the DTCH interrupt on detecting USB bus disconnection, and sets the DTCH flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.
The USB detects bus disconnection based on USB Specification 2.0.
After detecting the DTCH interrupt, the USB controls hardware as described below (irrespective of the setting of the corresponding interrupt enable bit). All the pipes in which communications are currently carried out for the USB port should be terminated by software and make a transition to the wait state for bus connection to the USB port (wait state for ATTCH interrupt generation).

- Modifies the DVSTCTR0.UACT bit for the port in which a DTCH interrupt has been detected to 0.
- Puts the port in which a DTCH interrupt has been generated into the idle state.

When the function controller is selected, the read value is invalid.

BCHG Flag (USB Bus Change Interrupt Status Flag)

Indicates the status of the USB bus change interrupt.
The USB detects the BCHG interrupt when a change in the full-speed or low-speed signal level occurs on the USB port (a change from J-state, K-state, or SE0 to J-state, K-state, or SE0), and sets the BCHG flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.
The USB sets the LNST[1:0] flags to indicate the current input state of the USB port. When the BCHG interrupt is generated, use software to repeat reading the LNST[1:0] flags until the same value is read three or more times, and eliminate chattering.
A change in the USB bus state can be detected even while the internal clock supply is stopped.
When the function controller is selected, the read value is invalid.

OVRCR Flag (Overcurrent Input Change Interrupt Status Flag)

Indicates the status of the USBm_OVRCURA and USBm_OVRCURB input pin change interrupt (m = 0, 1).
The USB detects the OVRCR interrupt when a change (high to low or low to high) occurs in at least one of the input values to the USBm_OVRCURA and USBm_OVRCURB pins, and sets the OVRCR flag to 1. Here, if the corresponding interrupt enable bit has been set to 1 by software, the USB generates the interrupt.

34.2.15 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 Flag for Pipe 0*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b1	PIPE1BRDY	BRDY Interrupt Status Flag for Pipe 1*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b2	PIPE2BRDY	BRDY Interrupt Status Flag for Pipe 2*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b3	PIPE3BRDY	BRDY Interrupt Status Flag for Pipe 3*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b4	PIPE4BRDY	BRDY Interrupt Status Flag for Pipe 4*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b5	PIPE5BRDY	BRDY Interrupt Status Flag for Pipe 5*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b6	PIPE6BRDY	BRDY Interrupt Status Flag for Pipe 6*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b7	PIPE7BRDY	BRDY Interrupt Status Flag for Pipe 7*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b8	PIPE8BRDY	BRDY Interrupt Status Flag for Pipe 8*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b9	PIPE9BRDY	BRDY Interrupt Status Flag for Pipe 9*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the SOFCFG.BRDYM bit is set to 0, to clear the status indicated by the flags in the BRDYSTS register, write 0 only to the flags to be cleared; write 1 to the other flags.

Note 2. When the SOFCFG.BRDYM bit is set to 0, clearing BRDY Interrupts should be done before accessing the FIFO.

34.2.16 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 Flag for Pipe 0	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b1	PIPE1NRDY	NRDY Interrupt Status Flag for Pipe 1	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b2	PIPE2NRDY	NRDY Interrupt Status Flag for Pipe 2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b3	PIPE3NRDY	NRDY Interrupt Status Flag for Pipe 3	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b4	PIPE4NRDY	NRDY Interrupt Status Flag for Pipe 4	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b5	PIPE5NRDY	NRDY Interrupt Status Flag for Pipe 5	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b6	PIPE6NRDY	NRDY Interrupt Status Flag for Pipe 6	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b7	PIPE7NRDY	NRDY Interrupt Status Flag for Pipe 7	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b8	PIPE8NRDY	NRDY Interrupt Status Flag for Pipe 8	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b9	PIPE9NRDY	NRDY Interrupt Status Flag for Pipe 9	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To clear the status indicated by the flags in the NRDYSTS register, write 0 only to the flags to be cleared; write 1 to the other flags.

34.2.17 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 Flag for Pipe 0	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b1	PIPE1BEMP	BEMP Interrupt Status Flag for Pipe 1	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b2	PIPE2BEMP	BEMP Interrupt Status Flag for Pipe 2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b3	PIPE3BEMP	BEMP Interrupt Status Flag for Pipe 3	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b4	PIPE4BEMP	BEMP Interrupt Status Flag for Pipe 4	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b5	PIPE5BEMP	BEMP Interrupt Status Flag for Pipe 5	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b6	PIPE6BEMP	BEMP Interrupt Status Flag for Pipe 6	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b7	PIPE7BEMP	BEMP Interrupt Status Flag for Pipe 7	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b8	PIPE8BEMP	BEMP Interrupt Status Flag for Pipe 8	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b9	PIPE9BEMP	BEMP Interrupt Status Flag for Pipe 9	0: Interrupts are not generated. 1: Interrupts are generated.	R/W *1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To clear the status indicated by the flags in the BEMPSTS register, write 0 only to the flags to be cleared; write 1 to the other flags.

34.2.18 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 Flag	Latest frame number	R
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	CRCE	Receive Data Error Flag	0: No error 1: An error occurred	R/W*1
b15	OVRN	Overrun/Underrun Detection Status Flag	0: No error 1: An error occurred	R/W*1

Note 1. To clear the status, write 0 only to the bits to be cleared; write 1 to the other bits.

FRNM[10:0] Flags (Frame Number Flag)

These bits indicate the latest frame number for the USB after the issuing of an SOF packet every 1 ms or writing to the FRNM[10:0] flags at the SOF packet reception.

CRCE Flag (Receive Data Error Flag)

Indicates whether a CRC error or bit stuffing error has been detected in the pipe during isochronous transfer.
The CRCE flag can be set to 0 by writing 0 to the CRCE flag by software. Here, 1 should be written to the other bits in FRMNUM.
On detecting a CRC error, the USB generates the internal NRDY interrupt request.

OVRN Flag (Overrun/Underrun Detection Status Flag)

Indicates whether an overrun/underrun error has been detected in the pipe during isochronous transfer.
The OVRN flag can be set to 0 by writing 0 to the OVRN flag by software. Here, 1 should be written to the other bits in FRMNUM.

(1) When the host controller is selected

The USB sets the OVRN flag to 1 on any of the following conditions.

- For the isochronous transfer pipe in the transmitting direction, the time to issue an OUT token comes before all the transmit data has been written to the FIFO buffer.
- For the isochronous transfer pipe in the receiving direction, the time to issue an IN token comes when no FIFO buffer planes are empty.

(2) When the function controller is selected

The USB sets the OVRN flag to 1 on any of the following conditions.

- For the isochronous transfer pipe in the transmitting direction, the IN token is received before all the transmit data has been written to the FIFO buffer.
- For the isochronous transfer pipe in the receiving direction, the OUT token is received when no FIFO buffer planes are empty.

34.2.19 Device State Change Register (DVCHGR)

Address(es): USB0.DVCHGR 000A 004Eh

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

Bit	Symbol	Bit Name	Description	R/W
b14 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	DVCHG	Device State Change	0: Disables the writing to the USBADDR.STSRECOV[3:0] bits and USBADDR.USBADDR[6:0] bits. 1: Enables the writing to the USBADDR.STSRECOV[3:0] bits and USBADDR.USBADDR[6:0] bits.	R/W

For details, refer to section 34.3.1.5, Release from Deep Software Standby Mode Due to USB Suspend/ Resume Interrupts.

34.2.20 USB Address Register (USBADDR)

Address(es): USB0.USBADDR 000A 0050h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	STSRECOV[3:0]				—	USBADDR[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	USBADDR[6:0]	USB Address	When the function controller is selected, these bits indicate the USB address assigned by the host when the SET_ADDRESS request is successfully processed.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b11 to b8	STSRECOV[3:0]	Status Recovery	• Recovery when the function controller is selected b11 b8 1 0 0 1: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b), bits INTSTS0.DVSQ[2:0] = 001b (Default state) 1 0 1 0: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b), bits INTSTS0.DVSQ[2:0] = 010b (Address state) 1 0 1 1: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b), bits INTSTS0.DVSQ[2:0] = 011b (Configured state) Settings other than above are prohibited. • Recovery when the host controller is selected b11 b8 1 0 0 0: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b) Settings other than above are prohibited.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

USBADDR[6:0] Bits (USB Address)

On detecting the USB bus reset, the USB sets the USBADDR[6:0] bits to 00h.

The writing to these bits is enabled while the DVCHGR.DVCHG bits are set to 1. On returning from the USB power shut-off, the operation can resume to the USB address before the shut-off by the software.

When the host controller is selected, the USBADDR[6:0] bits are invalid.

The USBADDR[6:0] bits are initialized when a USB bus reset is detected.

STSRECOV[3:0] Bits (Status Recovery)

These bits are used to resume the state of the internal sequencer on returning from the USB power shut-off. For details, refer to section 34.3.1.5, Release from Deep Software Standby Mode Due to USB Suspend/Resume Interrupts.

The writing to the STSRECOV[3:0] bits is enabled while the DVCHGR.DVCHG bit is set to 1.

34.2.21 USB Request Type Register (USBREQ)

Address(es): USB0.USBREQ 000A 0054h, USB1.USBREQ 000A 0254h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BREQUEST[7:0]								BMREQUESTTYPE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	BMREQUESTTYPE[7:0]	Request Type	These bits store the USB request bmRequestType value.	R/W *1
b15 to b8	BREQUEST[7:0]	Request	These bits store the USB request bRequest value.	R/W *1

Note 1. When the function controller is selected, these bits can only be read from, and writing to these bits is invalid. When the host controller is selected, these bits can be read from and written to.

The USBREQ register stores setup requests for control transfers.
When the function controller is selected, the values of bRequest and bmRequestType that have been received are stored.
When the host controller is selected, the values of bRequest and bmRequestType to be transmitted are set.
The USBREQ register is initialized by a USB bus reset.

BMREQUESTTYPE[7:0] Bits (Request Type)

These bits hold the value of the bmRequestType field of a USB request.

- When the host controller is selected:
  Set these bits to the value of the USB request data in setup transactions for transmission. Do not modify the value of the BMREQUESTTYPE[7:0] bits while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
  These bits indicate the value of the USB request data in setup transactions for reception. Writing to the bits has no effect.

BREQUEST[7:0] Bits (Request)

These bits store bRequest value of the USB request.

- When the host controller is selected:
  Set these bits to the value of the USB request data in setup transactions for transmission. Do not modify the value of the BREQUEST[7:0] bits while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
  These bits indicate the value of the USB request data in setup transactions for reception. Writing to the bits has no effect.

34.2.22 USB Request Value Register (USBVAL)

Address(es): USB0.USBVAL 000A 0056h, USB1.USBVAL 000A 0256h

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

When the function controller is selected, the value of wValue that has been received is stored in the USBVAL register. When the host controller is selected, the value of wValue to be transmitted is set.
The USBVAL register is initialized by a USB bus reset.

- When the host controller is selected:
  Set these bits to the value of the wValue field in USB requests of setup transactions for transmission. Do not modify the value of the USBVAL register while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
  These bits indicate the value of the wValue field in USB requests received in setup transactions for reception. Writing to the USBVAL register has no effect.

34.2.23 USB Request Index Register (USBINDX)

Address(es): USB0.USBINDX 000A 0058h, USB1.USBINDX 000A 0258h

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

The USBINDX register stores setup requests for control transfers.
When the function controller is selected, the value of wIndex that has been received is stored. When the host controller is selected, the value of wIndex to be transmitted is set.
The USBINDX register is initialized by a USB bus reset.

- When the host controller is selected:
  Set these bits to the value of the wIndex field in USB requests of setup transactions for transmission. Do not modify the value of the USBINDX register while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
  These bits indicate the value of the wIndex field in USB requests received in setup transactions for reception. Writing to the USBINDX register has no effect.

34.2.24 USB Request Length Register (USBLENG)

Address(es): USB0.USBLENG 000A 005Ah, USB1.USBLENG 000A 025Ah

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

The USBLENG register stores setup requests for control transfers.
When the function controller is selected, the value of wLength that has been received is stored. When the host controller is selected, the value of wLength to be transmitted is set.
The USBLENG register is initialized by a USB bus reset.

- When the host controller is selected:
  Set these bits to the value of the wLength field in USB requests of setup transactions for transmission. Do not modify the value of the USBLENG register while the DCPCTR.SUREQ bit is 1.
- When the function controller is selected:
  These bits indicate the value of the wLength field in USB requests received in setup transactions for reception. Writing to the USBLENG register has no effect.

34.2.25 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	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DIR	Transfer Direction*1	0: Data receiving direction 1: Data transmitting direction	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer*1	0: Pipe continued at the end of transfer 1: Pipe disabled at the end of transfer	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Modify this bit while DCPCTR.PID[1:0] bits are 00b (NAK). Before modifying this bit after modifying the DCPCTR.PID[1:0] bits for the DCP from 01b (BUF) to 00b (NAK), check that the DCPCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

DIR Bit (Transfer Direction)

When the host controller is selected, the DIR bit sets the transfer direction of the data stage and status stage.
When the function controller is selected, the DIR bit should be set to 0.

SHTNAK Bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to modify PID[1:0] bits to 00b (NAK) upon the end of transfer when the selected pipe is in the receiving direction.
The SHTNAK bit is valid when the selected pipe in the receiving direction.
When the SHTNAK bit is set to 1, the USB modifies the DCPCTR.PID[1:0] bits for the DCP to 00b (NAK) on determining the end of the transfer. The USB determines that the transfer has ended on the following condition.

- A short packet (including a zero-length packet) is successfully received.

34.2.26 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]				—	—	—	—	—	MXPS[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	MXPS[6:0]	Maximum Packet Size*1	These bits set the maximum amount of data (maximum packet size) in payloads for the DCP.	R/W
b11 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b12	DEVSEL[3:0]	Device Select*2	b15 b12 0 0 0 0: Address 0000 0 0 0 1: Address 0001 0 0 1 0: Address 0010 0 0 1 1: Address 0011 0 1 0 0: Address 0100 0 1 0 1: Address 0101 Settings other than above are prohibited.	R/W

Note 1. Modify the MXPS[6:0] bits while PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the DCPCTR.PID[1:0] bits for the DCP from 01b (BUF) to 00b (NAK), check that the DCPCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software. After modifying the MXPS[6:0] bits and the DCP has been set to the CURPIPE[3:0] bits in a port select register, clear the buffer by setting the BCLR bit the port control register to 1.

Note 2. Modify the DEVSEL[3:0] bits while PID[1:0] bits are 00b (NAK) and the DCPCTR.SUREQ bit is 0. To modify these bits after modifying the DCPCTR.PID[1:0] bits for the DCP from 01b (BUF) to 00b (NAK), check that the DCPCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

MXPS[6:0] Bits (Maximum Packet Size)

The MXPS[6:0] bits specify the maximum amount of data (maximum packet size) in payloads for the DCP. The initial value of the bits is 40h (64 bytes).

Ensure that the setting of the MXPS[6:0] bits is in compliance with USB Specification 2.0.

Do not write to the FIFO buffer or set PID[1:0] = 01b (BUF) while the setting of the MXPS[6:0] bits is 0.

DEVSEL[3:0] Bits (Device Select)

When the host controller is selected, these bits specify the address of the peripheral device which is the communication target during control transfer.

The DEVSEL[3:0] bits should be set after setting the address to the DEVADDn (n = 0 to 5) register corresponding to the value to be set in the DEVSEL[3:0] bits. For example, before setting the DEVSEL[3:0] bits to 0010b, the address should be set to DEVADD2.

When the function controller is selected, the DEVSEL[3:0] bits should be set to 0000b.

34.2.27 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	—	—	SUREQCLR	—	—	SQCLR	SQSET	SQMON	PBUSY	—	—	CCPL	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
b2	CCPL	Control Transfer End Enable	0: Invalid 1: Completion of control transfer is enabled.	R/W
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy Flag	0: DCP is not used for the transaction. 1: DCP is used for the transaction.	R
b6	SQMON	Sequence Toggle Bit Monitor Flag	0: DATA0 1: DATA1	R
b7	SQSET	Sequence Toggle Bit Set*2	0: Invalid 1: Specifies DATA1.	R/W*1
b8	SQCLR	Sequence Toggle Bit Clear*2	0: Invalid 1: Specifies DATA0.	R/W*1
b10, b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	SUREQCLR	SUREQ Bit Clear	0: Invalid 1: Clears the SUREQ bit to 0.	R/W
b13, b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	SUREQ	Setup Token Transmission	0: Invalid 1: Transmits the setup packet.	R/W
b15	BSTS	Buffer Status Flag	0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note 1. Only 0 can be read.
Note 2. Write 1 to the SQSET and SQCLR bits while PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PID[1:0] bits for the DCP from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

PID[1:0] Bits (Response PID)

The PID[1:0] bits control the response type of the USB during control transfer.

(1) When the host controller is selected

Modify the setting of the PID[1:0] bits from 00b (NAK) to 01b (BUF) using the following procedure.

- When the transmitting direction is set
  Write all the transmit data to the FIFO buffer while the DVSTCTR0.UACT bit is 1 and PID[1:0] bits are 00b (NAK), and then write 01b (BUF response). After PID[1:0] have been set to 01b (BUF), the USB executes the OUT transaction.
- When the receiving direction is set
  Check that the FIFO buffer is empty (or empty the buffer) while the DVSTCTR0.UACT bit is 1 and PID[1:0] bits are 00b (NAK), and then set PID[1:0] bits to 01b (BUF). After PID[1:0] bits have been set to 01b (BUF), the USB executes the IN transaction.

The USB modifies the setting of the PID[1:0] bits as follows.

- The USB sets PID[1:0] bits to 11b (STALL) on receiving the data of a size exceeding the maximum packet size when the PID[1:0] bits has been set to 01b (BUF) by software.
- The USB sets PID[1:0] bits to 00b (NAK) on detecting a receive error, such as a CRC error, three consecutive times.
- The USB also sets PID[1:0] bits to 11b (STALL) on receiving the STALL handshake.

(2) When the function controller is selected

The USB modifies the setting of the PID[1:0] bits as follows.

- The USB modifies the PID[1:0] bits to 00b (NAK) on receiving the setup packet. Here, the USB sets the INTSTS0.VALID flag to 1. The setting of the PID[1:0] bits cannot be modified until the VALID flag is set to 0 by software.
- The USB sets PID[1:0] bits to 11b (STALL) on receiving the data of a size exceeding the maximum packet size when the PID[1:0] bits have been set to 01b (BUF) by software.
- The USB sets PID[1:0] bits to 1xb (STALL) on detecting the control transfer sequence error.
- The USB sets PID[1:0] bits to 00b (NAK) on detecting the USB bus reset.

The USB does not check the setting of the PID[1:0] bits while the SET_ADDRESS request is processed.
The PID[1:0] bits are initialized by a USB bus reset.

CCPL Bit (Control Transfer End Enable)

When the function controller is selected, setting the CCPL bit to 1 enables the status stage of the control transfer to be completed.
When the CCPL bit is set to 1 by software while the corresponding PID[1:0] bits are set to 01b (BUF), the USB completes the control transfer status stage.
During control read transfer, the USB transmits the ACK handshake in response to the OUT transaction from the USB host, and transmits the zero-length packet in response to the IN transaction from the USB host during control write or no-data control transfer. However, on detecting the SET_ADDRESS request, the USB operates in auto response mode from the setup stage up to the status stage completion irrespective of the setting of the CCPL bit.
The USB modifies the CCPL bit from 1 to 0 on receiving a new setup packet.
1 cannot be written to the CCPL bit by software while the INTSTS0.VALID flag is 1.
The CCPL bit is initialized by a USB bus reset.
When the host controller is selected, be sure to write 0 to the CCPL bit.

PBUSY Flag (Pipe Busy Flag)

The PBUSY flag indicates whether DCP is used or not for the transaction when USB changes the PID[1:0] bits from 01b (BUF) to 00b (NAK).
The USB modifies the PBUSY flag from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY flag from 1 to 0 upon completion of one transaction.
Reading the PBUSY flag after the PID[1:0] bits have been set to 00b (NAK) by software allows checking whether modification of the pipe settings is possible.
For details, refer to section 34.3.4.1, Pipe Control Register Switching Procedures.

SQMON Flag (Sequence Toggle Bit Monitor Flag)

The SQMON flag indicates the expected value of the sequence toggle bit for the next transaction during the DCP transfer.
The USB allows the SQMON flag to toggle upon normal completion of the transaction. However, the SQMON flag is not allowed to toggle when a data PID mismatch occurs during the transfer in the receiving direction.
When the function controller is selected, the USB sets the SQMON flag to 1 (specifies DATA1 as the expected value) upon successful reception of the setup packet.
When the function controller is selected, the USB does not reference the SQMON flag during the IN/OUT transaction of the status stage, and does not allow the SQMON flag to toggle upon normal completion.

SQSET Bit (Sequence Toggle Bit Set)

The SQSET bit specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer.
Do not set the SQCLR and SQSET bits to 1 simultaneously.

SQCLR Bit (Sequence Toggle Bit Clear)

The SQCLR bit specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer. The SQCLR bit indicates 0.
Do not set the SQCLR and SQSET bits to 1 simultaneously.

SUREQCLR Bit (SUREQ Bit Clear)

When the host controller is selected, setting the SUREQCLR bit to 1 clears the SUREQ bit to 0. The SUREQCLR bit indicates 0.
Set the SUREQCLR bit to 1 through software when communication has stopped with the SUREQ bit being 1 during the setup transaction. However, for normal setup transactions, the USB automatically clears the SUREQ bit to 0 upon completion of the transaction; therefore, clearing the SUREQ bit through software is not necessary.
Controlling the SUREQ bit through the SUREQCLR bit must be done while the DVSTCTR0.UACT bit is 0 and thus communication is halted or while no transfer is being performed with bus disconnection detected.
When the function controller is selected, be sure to write 0 to the SUREQCLR bit.

SUREQ Bit (Setup Token Transmission)

The USB transmits the setup packet by setting the SUREQ bit to 1 when the host controller is selected.
After completing the setup transaction process, the USB generates either the SACK or SIGN interrupt and sets the SUREQ bit to 0.
The USB also sets the SUREQ bit to 0 when software sets the SUREQCLR bit to 1.
Before setting the SUREQ bit to 1, set the DCPMAXP.DEVSEL[3:0] bits, registers USBREQ, USBVAL, USBINDX, and USBLENG appropriately to transmit the desired USB request in the setup transaction. Before setting this bit to 1, check that the PID[1:0] bits for the DCP are set to 00b (NAK). After setting the SUREQ bit to 1, do not modify the DCPMAXP.DEVSEL[3:0] bits, registers USBREQ, USBVAL, USBINDX, or USBLENG until the setup transaction is completed (the SUREQ bit = 1).
Write 1 to the SUREQ bit only when transmitting the setup token; for other purposes, write 0.
When the function controller is selected, be sure to write 0 to the SUREQ bit.

BSTS Flag (Buffer Status Flag)

Indicates whether DCP FIFO buffer access is enabled or disabled.
The meaning of the BSTS flag depends on the setting of ISEL bit in the port select register as shown below.

- When the ISEL bit = 0, the BSTS flag indicates whether the received data can be read from the buffer.
- When the ISEL bit = 1, the BSTS flag indicates whether the data to be transmitted can be written to the buffer.

34.2.28 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: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Settings other than above are prohibited.	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Pipes 1 to 9 should be set using registers PIPESEL, PIPECFG, PIPEMAXP, PIPEPERI, PIPEnCTR, PIPEnTRE, and PIPEnTRN.

After selecting the pipe using the PIPESEL register, functions of the pipe should be set using registers PIPECFG, PIPEMAXP, and PIPEPERI. PIPEnCTR, PIPEnTRE, and PIPEnTRN can be set regardless of the pipe selection in the PIPESEL register.

PIPESEL[3:0] Bits (Pipe Window Select)

The PIPESEL[3:0] bits select the pipe number corresponding to registers PIPECFG, PIPEMAXP, and PIPEPERI which data are written to or read from.

Selecting a pipe number through the PIPESEL[3:0] bits allows writing to and reading from registers PIPECFG, PIPEMAXP, and PIPEPERI which correspond to the selected pipe number.

When PIPESEL[3:0] = 0000b, 0 is read from all of the bits in registers PIPECFG, PIPEMAXP, and PIPEPERI. Writing to these bits is invalid.

34.2.29 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 0000b means an unused pipe.	R/W
b4	DIR	Transfer Direction*2,*3	0: Receiving direction 1: Transmitting direction	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer*1	0: Pipe assignment continued at the end of transfer 1: Pipe assignment disabled at the end of transfer	R/W
b8	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b9	DBLB	Double Buffer Mode*2,*3	0: Single buffer 1: Double buffer	R/W
b10	BFRE	BRDY Interrupt Operation Specification*2,*3	0: BRDY interrupt upon transmitting or receiving data 1: BRDY interrupt upon completion of reading data	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	TYPE[1:0]	Transfer Type*1	• Pipes 1 and 2 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Setting prohibited 1 1: Isochronous transfer • Pipes 3 to 5 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Setting prohibited 1 1: Setting prohibited • Pipes 6 to 9 b15 b14 0 0: Pipe not used 0 1: Setting prohibited 1 0: Interrupt transfer 1 1: Setting prohibited	R/W

Note 1. Modify the TYPE[1:0], SHTNAK, and EPNUM[3:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

Note 2. Modify the BFRE, DBLB, and DIR bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK) and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), checking the PIPEnCTR.PBUSY flag through software is not necessary.

Note 3. To modify the BFRE, DBLB, and DIR bits after completing USB communication using the selected pipe, write 1 and then 0 to the PIPEnCTR.ACLRM bit continuously through software to clear the FIFO buffer assigned to the selected pipe while the PID[1:0] bits and CURPIPE[3:0] bits are in the state described in the above note 2.

The PIPECFG register specifies the transfer type, buffer memory access direction, and endpoint numbers for pipes 1 to 9. It also selects single or double buffer mode, and whether to continue or disable pipe operation at the end of transfer.

EPNUM[3:0] Bits (Endpoint Number)

The EPNUM[3:0] bits specify the endpoint number for the selected pipe.
Setting 0000b means an unused pipe.
Do not make the settings such that the combination of the settings of the DIR and EPNUM[3:0] bits should be the same for two or more pipes (EPNUM[3:0] bits = 0000b can be set for all of the pipes).

DIR Bit (Transfer Direction)

The DIR bit specifies the transfer direction for the selected pipe.
When the DIR bit has been set to 0 by software, the USB uses the selected pipe in the receiving direction, and when software has set the DIR bit to 1, the USB uses the selected pipe in the transmitting direction.

SHTNAK Bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to modify the PID[1:0] bits to 00b (NAK) upon the end of transfer when the selected pipe is in the receiving direction.
The SHTNAK bit is valid when the selected pipe is pipe 1 to 5 in the receiving direction.
When the SHTNAK bit has been set to 1 by software for the selected pipe in the receiving direction, the USB modifies the PIPEnCTR.PID[1:0] bits corresponding to the selected pipe to 00b (NAK) on determining the end of the transfer. The USB determines that the transfer has ended on any of the following conditions.

- A short packet (including a zero-length packet) is successfully received.
- The transaction counter is used and the number of packets specified by the counter are successfully received.

DBLB Bit (Double Buffer Mode)

The DBLB bit selects either single or double buffer mode for the FIFO buffer used by the selected pipe.
The DBLB bit is valid when the selected pipe is pipe 1 to 5.

BFRE Bit (BRDY Interrupt Operation Specification)

The BFRE bit specifies the BRDY interrupt generation timing from the USB to the CPU with respect to the selected pipe.
When the BFRE bit has been set to 1 by software and the selected pipe is in the receiving direction, the USB detects the transfer completion and generates the BRDY interrupt on having read the relevant packet.
When the BRDY interrupt is generated with the above conditions, 1 should be written to the BCLR bit in the port control register by software. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to the BCLR bit.
When the BFRE bit has been set to 1 by software and the selected pipe is in the transmitting direction, the USB does not generate the BRDY interrupt.
For details, refer to section 34.3.3.1, BRDY Interrupt.

TYPE[1:0] Bits (Transfer Type)

The TYPE[1:0] bits select the transfer type for the pipe selected by the PIPESEL.PIPESEL[3:0] bits (selected pipe).
Before setting the PID[1:0] bits to 01b (BUF) for the selected pipe (before starting USB communication using the selected pipe), set the TYPE[1:0] bits to a value other than 00b.

34.2.30 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]				—	—	—	MXPS[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0/1 *1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	MXPS[8:0]	Maximum Packet Size*2	• Pipes 1 and 2: 1 byte (001h) to 256 bytes (100h) • Pipes 3 to 5: 8 bytes (008h), 16 bytes (010h), 32 bytes (020h), 64 bytes (040h) (Bits [8:7] and [2:0] are not provided.) • Pipes 6 to 9: 1 byte (001h) to 64 bytes (040h) (Bits [8:7] are not provided.)	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b12	DEVSEL[3:0]	Device Select*3	b3 b0 0 0 0 0: Address 0000 0 0 0 1: Address 0001 0 0 1 0: Address 0010 0 0 1 1: Address 0011 0 1 0 0: Address 0100 0 1 0 1: Address 0101 Settings other than above are prohibited.	R/W

Note 1. The value of these bits is 0000h when no pipe is selected with the PIPESEL.PIPESEL[3:0] bits and 0040h when a pipe is selected.

Note 2. Modify the MXPS[8:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK) and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

Note 3. Modify the DEVSEL[3:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK). To modify these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

The PIPEMAXP register specifies the maximum packet size for pipes 1 to 9.

MXPS[8:0] Bits (Maximum Packet Size)

The MXPS[8:0] bits specify the maximum data payload (maximum packet size) for the selected pipe.

These bits should be set to the appropriate value for each transfer type based on USB Specification 2.0. Note that the maximum value of pipes 1 and 2 is 256. While MXPS[8:0] = 000h, do not write to the FIFO buffer or do not set the PID[1:0] bits to 01b (BUF).

DEVSEL[3:0] Bits (Device Select)

When the host controller is selected, these bits specify the USB device address of the peripheral device which is the communication target.

The DEVSEL[3:0] bits should be set after setting the address to the DEVADDn (n = 0 to 5) register corresponding to the value to be set in the DEVSEL[3:0] bits. For example, before setting the DEVSEL[3:0] bits to 0010b, the address should be set to DEVADD2.

When the function controller is selected, the DEVSEL[3:0] bits should be set to 0000b.

34.2.31 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]	Interval Error Detection Interval *1	Specify the interval error detection timing for the selected pipe in terms of frames, which is expressed as nth power of 2.	R/W
b11 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	IFIS	Isochronous IN Buffer Flush	0: The buffer is not flushed. 1: The buffer is flushed.	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Modify the IITV[2:0] bits while the PIPEnCTR.PID[1:0] bits are 00b (NAK). To modify these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

The PIPEPERI register selects whether the buffer is flushed or not when an interval error occurred during isochronous IN transfer, and sets the interval error detection interval for pipes 1 to 9.

IITV[2:0] Bits (Interval Error Detection Interval)

Before modifying the IITV[2:0] bits after USB communication has been completed with the IITV[2:0] bits set to a certain value, set the PID[1:0] bits to 00b (NAK) and then set the PIPEnCTR.ACLRM bit to 1 to initialize the interval timer.

The IITV[2:0] bits are invalid for pipes 3 to 5; set the IITV[2:0] bits to 000b for pipes 3 to 5.

IFIS Bit (Isochronous IN Buffer Flush)

Specifies whether to flush the buffer when the pipe selected by the PIPESEL.PIPESEL[3:0] bits (selected pipe) is used for isochronous IN transfers.

When the function controller is selected and the selected pipe is for isochronous IN transfers, the USB automatically clears the FIFO buffer when the USB fails to receive the IN token from the USB host within the interval set by the IITV[2:0] bits in terms of frames.

In double buffer mode (the PIPECFG.DBLB bit = 1), the USB only clears the data in the plane used earlier.

The USB clears the FIFO buffer on receiving the SOF packet immediately after the frame in which the USB has expected to receive the IN token. Even if the SOF packet is damaged, the USB also clears the FIFO buffer at the right timing to receive the SOF packet by using the internal interpolation function.

When the host controller is selected, set the IITV[2:0] bits to 000b.

When the selected pipe is not for isochronous transfer, set the IITV[2:0] bits to 000b.

34.2.32 Pipe n 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 read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy Flag	0: The relevant pipe is not used for the transaction. 1: The relevant pipe is used for the transaction.	R
b6	SQMON	Sequence Toggle Bit Confirmation	0: DATA0 1: DATA1	R
b7	SQSET	Sequence Toggle Bit Set*2	0: Write disabled 1: Specifies DATA1.	R/W*1
b8	SQCLR	Sequence Toggle Bit Clear *2	0: Write disabled 1: Specifies DATA0.	R/W*1
b9	ACLRM	Auto Buffer Clear Mode*3	0: Disabled 1: Enabled (all buffers are initialized)	R/W
b10	ATREPM	Auto Response Mode*2	0: Auto response is disabled. 1: Auto response is enabled.	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	INBUFM	Transmit Buffer Monitor Flag	0: There is no data to be transmitted in the buffer memory. 1: There is data to be transmitted in the buffer memory.	R
b15	BSTS	Buffer Status Flag	0: Buffer access by the CPU is disabled. 1: Buffer access by the CPU is enabled.	R

Note 1. Only 0 can be read.
Note 2. Modify the ATREPM bit or write 1 to the SQCLR or SQSET bit while the PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.
Note 3. Modify the ACLRM bit while PID[1:0] bits are 00b (NAK) and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying this bit after modifying the PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

The PIPEnCTR register can be set regardless of the pipe selection in the PIPESEL register.

PID[1:0] Bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction of the relevant pipe.
The default setting of the PID[1:0] bits are 00b (NAK). Modify the setting of the PID[1:0] bits to 01b (BUF) to use the relevant pipe for USB transfer. Table 34.6 and Table 34.7 show the basic operation (operation when there are no errors in the transmitted and received packets) of the USB depending on the PID[1:0] bit setting.
After modifying the setting of the PID[1:0] bits through software from 01b (BUF) to 00b (NAK) during USB communication using the relevant pipe, check that the PBUSY flag is 1 to see if USB transfer using the relevant pipe has actually entered the NAK state. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.
The USB modifies the setting of the PID[1:0] bits in the following cases.

- The USB sets the PID[1:0] bits to 00b (NAK) on recognizing the completion of the transfer when the relevant pipe is in the receiving direction and the PIPECFG.SHTNAK bit for the selected pipe has been set to 1 by software.
- The USB sets the PID[1:0] bits to 11b (STALL) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB sets the PID[1:0] bits to 00b (NAK) on detecting a USB bus reset when the function controller is selected.
- The USB sets the PID[1:0] bits to 00b (NAK) on detecting a receive error, such as a CRC error, three consecutive times when the host controller is selected.
- The USB sets the PID[1:0] bits to 11b (STALL) on receiving the STALL handshake when the host controller is selected.

To specify each response type, set the PID[1:0] bits as follows.

- To make a transition from NAK (00b) to STALL, set 10b.
- To make a transition from BUF (01b) to STALL, set 11b.
- To make a transition from STALL (11b) to NAK, set 10b and then 00b.
- To make a transition from STALL (11b) to BUF, set 10b, 00b, and then 01b.
- To make a transition from STALL (10b) to BUF, set 00b and then 01b.

PBUSY Flag (Pipe Busy Flag)

The PBUSY flag indicates whether the relevant pipe is being currently used or not for the transaction.
The USB modifies the PBUSY flag from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY flag from 1 to 0 upon completion of one transaction.
Reading the PBUSY flag after the PID[1:0] bits have been set to 00b (NAK) by software allows checking whether modification of the pipe settings is possible.
For details, refer to section 34.3.4.1, Pipe Control Register Switching Procedures.

SQMON Bit (Sequence Toggle Bit Confirmation)

The SQMON flag indicates the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
When the relevant pipe is not for the isochronous transfer, the USB allows the SQMON flag to toggle upon normal completion of the transaction. However, the SQMON flag is not allowed to toggle when a data PID mismatch occurs during the transfer in the receiving direction.

SQSET Bit (Sequence Toggle Bit Set)

The SQSET bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
Setting the SQSET bit to 1 through software allows the USB to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQSET bit to 0.

SQCLR Bit (Sequence Toggle Bit Clear)

The SQCLR bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the relevant pipe.
Setting the SQCLR bit to 1 through software allows the USB to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQCLR bit to 0.

ACLRM Bit (Auto Buffer Clear Mode)

Enables or disables auto buffer clear mode for the relevant pipe.
To delete the information in the FIFO buffer assigned to the relevant pipe completely, write 1 and then 0 to the ACLRM bit continuously.
Table 34.8 shows the information cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which clearing the information is necessary.

ATREPM Bit (Auto Response Mode)

Enables or disables auto response mode for the relevant pipe.
When the function controller is selected and the relevant pipe is for bulk transfer, the ATREPM bit can be set to 1.
When the ATREPM bit is set to 1, the USB responds to the token from the USB host as described below.
(1) When the relevant pipe is for bulk IN transfer (the PIPECFG.TYPE[1:0] bits = 01b and the PIPECFG.DIR bit = 1)
When the ATREPM bit = 1 and PID[1:0] = 01b (BUF), the USB transmits a zero-length packet in response to the IN token.
The USB updates (allows toggling of) the sequence toggle bit (data PID) each time the USB receives ACK from the USB host (in a single transaction, IN token is received, zero-length packet is transmitted, and then ACK is received.).
In this case, the USB does not generate the BRDY or BEMP interrupt.
(2) When the relevant pipe is for bulk OUT transfer (the PIPECFG.TYPE[1:0] bits = 01b and the PIPECFG.DIR bit = 0)
When the ATREPM bit = 1 and PID[1:0] = 01b (BUF), the USB returns NAK in response to the OUT token and generates the NRDY interrupt.
For USB communication in auto response mode, set the ATREPM bit to 1 while the FIFO buffer is empty. Do not write to the FIFO buffer during USB communication in auto response mode.
When the relevant pipe is for isochronous transfer, be sure to set the ATREPM bit to 0.
When the host controller is selected, be sure to set the ATREPM bit to 0.

INBUFM Flag (Transmit Buffer Monitor Flag)

Indicates the relevant FIFO buffer status when the relevant pipe is in the transmitting direction.
When the relevant pipe is in the transmitting direction (the PIPECFG.DIR bit = 1), the USB sets the INBUFM flag to 1 when the CPU or DMAC/DTC completes writing data to at least one FIFO buffer plane.
The USB sets the INBUFM flag to 0 when the USB completes transmitting the data from the FIFO buffer plane to which all the data has been written. In double buffer mode (the PIPECFG.DBLB bit = 1), the USB sets the INBUFM flag to 0 when the USB completes transmitting the data from the two FIFO buffer planes before the CPU or DMAC/DTC completes writing data to one FIFO buffer plane.
The INBUFM flag indicates the same value as the BSTS flag when the relevant pipe is in the receiving direction (the PIPECFG.DIR bit = 0).

BSTS Flag (Buffer Status Flag)

Indicates the FIFO buffer status for the relevant pipe.

The meaning of the BSTS flag depends on the settings of the PIPECFG.DIR bit, PIPECFG.BFRE bit, and DnFIFOSEL.DCLRM bits as shown in Table 34.9.

Table 34.6 Operation of USB depending on PID[1:0] Bits Setting (When Host Controller is Selected)

PID[1:0] Bits	Transfer Type	Transfer Direction (DIR Bit)	Operation of USB
00b (NAK)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.
01b (BUF)	Bulk or interrupt	Operation does not depend on the setting.	Issues tokens while the DVSTCTR0.UACT bit is 1 and the FIFO buffer corresponding to the relevant pipe is ready for transmission and reception. Does not issue tokens while the DVSTCTR0.UACT bit is 0 or the FIFO buffer corresponding to the relevant pipe is not ready for transmission or reception.
	Isochronous	Operation does not depend on the setting.	Issues tokens irrespective of the status of the FIFO buffer corresponding to the relevant pipe.
10b (STALL) or 11b (STALL)	Operation does not depend on the setting.	Operation does not depend on the setting.	Does not issue tokens.

Table 34.7 Operation of USB depending on PID[1:0] Bits Setting (When Function Controller is Selected)

PID[1:0] Bits	Transfer Type	Transfer Direction (DIR Bit)	Operation of USB
00b (NAK)	Bulk or interrupt	Operation does not depend on the setting.	Returns NAK in response to the token from the USB host.
	Isochronous	Operation does not depend on the setting.	Returns nothing in response to the token from the USB host.
01b (BUF)	Bulk	Receiving direction (DIR bit = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception.
	Interrupt	Receiving direction (DIR bit = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception.
	Bulk or interrupt	Transmitting direction (DIR bit = 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Returns NAK if not ready.
	Isochronous	Receiving direction (DIR bit = 0)	Receives data in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception. Discards data if not ready.
	Isochronous	Transmitting direction (DIR bit = 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Transmits the zero-length packet if not ready.
10b (STALL) or 11b (STALL)	Bulk or interrupt	Operation does not depend on the setting.	Returns STALL in response to the token from the USB host.
	Isochronous	Operation does not depend on the setting.	Returns nothing in response to the token from the USB host.

Table 34.8 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 34.9 Operation of BSTS Flag

DIR Bit	BFRE Bit	DCLRM Bit	BSTS Flag Function
0	0	0	The received data can be read from the FIFO buffer. The received data has been completely read from the FIFO buffer.
		1	Setting prohibited
	1	0	The received data can be read from the FIFO buffer. The BCLR bit in the port control register has been set to 1 by software after the received data has been completely read from the FIFO buffer.
		1	The received data can be read from the FIFO buffer. The received data has been completely read from the FIFO buffer.
1	0	0	The transmit data can be written to the FIFO buffer. The transmit data has been completely written to the FIFO buffer.
		1	Setting prohibited
	1	0	Setting prohibited
		1	Setting prohibited

- PIPEnCTR (n = 6 to 9)

Address(es): USB0.PIPE6CTR 000A 007Ah, USB0.PIPE7CTR 000A 007Ch, USB0.PIPE8CTR 000A 007Eh, USB0.PIPE9CTR 000A 0080h, USB1.PIPE6CTR 000A 027Ah, USB1.PIPE7CTR 000A 027Ch, USB1.PIPE8CTR 000A 027Eh, USB1.PIPE9CTR 000A 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 read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy Flag	0: The relevant pipe is not used at the USB bus. 1: The relevant pipe is used at the USB bus.	R
b6	SQMON	Sequence Toggle Bit Confirmation Flag	0: DATA0 1: DATA1	R
b7	SQSET	Sequence Toggle Bit Set*2	0: Invalid 1: Specifies DATA1.	R/W *1
b8	SQCLR	Sequence Toggle Bit Clear *2	0: Invalid 1: Specifies DATA0.	R/W *1
b9	ACLRM	Auto Buffer Clear Mode*2,*3	0: Auto buffer clear mode is disabled. 1: Auto buffer clear mode is enabled (all buffers are initialized)	R/W
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	BSTS	Buffer Status Flag	0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note 1. Only 0 can be read. Only 1 can be written.

Note 2. Write 1 to the SQCLR or SQSET bit while the PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

Note 3. Modify the ACLRM bit while the PID[1:0] bits are 00b (NAK) and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying this bit after modifying the PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

PID[1:0] Bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction of the relevant pipe.
The default setting of the PID[1:0] bits are 00b (NAK). Modify the setting of the PID[1:0] bits to 01b (BUF) to use the relevant pipe for USB transfer. Table 34.6 and Table 34.7 show the basic operation (operation when there are no errors in the transmitted and received packets) of the USB depending on the setting of the PID[1:0] bits.
After modifying the setting of the PID[1:0] bits through software from 01b (BUF) to 00b (NAK) during USB communication using the relevant pipe, check that the PBUSY flag is 1 to see if USB transfer using the relevant pipe has actually entered the NAK state. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.
The USB modifies the setting of the PID[1:0] bits in the following cases.

- The USB sets the PID[1:0] bits to 11b (STALL) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB sets the PID[1:0] bits to 00b (NAK) on detecting a USB bus reset when the function controller is selected.
- The USB sets the PID[1:0] bits to 00b (NAK) on detecting a receive error, such as a CRC error, three consecutive times when the host controller is selected.
- The USB sets the PID[1:0] bits to 11b (STALL) on receiving the STALL handshake when the host controller is selected.

To specify each response type, set the PID[1:0] bits as follows.

- To make a transition from NAK (00b) to STALL, set 10b.
- To make a transition from BUF (01b) to STALL, set 11b.
- To make a transition from STALL (11b) to NAK, set 10b and then 00b.
- To make a transition from STALL (11b) to BUF, set 10b, 00b, and then 01b.
- To make a transition from STALL (10b) to BUF, set 00b and then 01b.

PBUSY Flag (Pipe Busy Flag)

The PBUSY flag indicates whether the relevant pipe is being currently used or not for the transaction.
The USB modifies the PBUSY flag from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY flag from 1 to 0 upon completion of one transaction.
Reading the PBUSY flag after the PID[1:0] bits has been set to 00b (NAK) by software allows checking whether modification of the pipe settings is possible.

SQMON Flag (Sequence Toggle Bit Confirmation Flag)

The SQMON flag indicates the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
The USB allows the SQMON flag to toggle upon normal completion of the transaction. However, the SQMON flag is not allowed to toggle when a data PID mismatch occurs during the transfer in the receiving direction.

SQSET Bit (Sequence Toggle Bit Set)

The SQSET bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the relevant pipe.
Setting the SQSET bit through software allows the USB to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQSET bit to 0.

SQCLR Bit (Sequence Toggle Bit Clear)

The SQCLR bit should be set to 1 to clear the expected value (to set DATA0 as the expected value) of the sequence toggle bit for the next transaction of the relevant pipe.
Setting the SQCLR bit to 1 through software allows the USB to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB sets the SQCLR bit to 0.

ACLRM Bit (Auto Buffer Clear Mode)

Enables or disables auto buffer clear mode for the relevant pipe.

To delete the information in the FIFO buffer assigned to the relevant pipe completely, write 1 and then 0 to the ACLRM bit continuously.

Table 34.10 shows the information cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which clearing the information is necessary.

BSTS Flag (Buffer Status Flag)

Indicates the FIFO buffer status for the relevant pipe.

The meaning of the BSTS flag depends on the settings of the PIPECFG.DIR bit, PIPECFG.BFRE bit, and DnFIFOSEL.DCLRM bits as shown in Table 34.9.

Table 34.10 Information Cleared by USB by Setting the ACLRM Bit = 1

No.	Information Cleared by ACLRM Bit Manipulation	Cases in which Clearing Information is Necessary
1	All the information in the FIFO buffer assigned to the selected pipe	When the pipe is to be initialized
2	The interval count value when the selected pipe is for interrupt transfer and the host controller is selected	When the interval count value is to be reset
3	Internal flags concerning the PIPECFG.BFRE bit	When the PIPECFG.BFRE setting is modified
4	Internal flags concerning the transaction count	When the transaction count function is forcibly terminated

34.2.33 Pipe n Transaction Counter Enable Register (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 read as 0. The write value should be 0.	R/W
b8	TRCLR	Transaction Counter Clear	0: Invalid 1: The current counter value is cleared.	R/W
b9	TRENB	Transaction Counter Enable	0: Transaction counter is disabled. 1: Transaction counter is enabled.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Modify each bit in the PIPEnTRE register while the PID[1:0] bits are 00b (NAK). Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from 01b (BUF) to 00b (NAK), check that the PIPEnCTR.PBUSY flag is 0. However, if the USB changes the PID[1:0] bits to 00b (NAK), the PBUSY flag does not need to be checked by software.

TRCLR Bit (Transaction Counter Clear)

Clears the current value of the transaction counter corresponding to the relevant pipe and then sets the TRCLR bit to 0.

TRENB Bit (Transaction Counter Enable)

Enables or disables the transaction counter.

For the pipe in the receiving direction, setting the TRENB bit to 1 after setting the total number of the packets to be received in the PIPEnTRN register through software allows the USB to control hardware as described below on having received the number of packets equal to the setting of the PIPEnTRN register.

- While the PIPECFG.SHTNAK bit is 1, the USB modifies the PID[1:0] bits to 00b (NAK) for the corresponding pipe on having received the number of packets equal to the setting of the PIPEnTRN register.
- 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 PIPEnTRN register and then reading the last received data.

For the pipe in the transmitting direction, set the TRENB bit to 0.

When the transaction counter is not used, set the TRENB bit to 0.

When the transaction counter is used, set the PIPEnTRN register 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.

34.2.34 Pipe n Transaction Counter Register (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

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

The PIPEnTRN register retains the setting by a USB bus reset.

- When written to:
  This register specifies the total of packets (number of transactions) to be received in corresponding pipe.
- When read from:
  This register indicates the specified number of transactions if the PIPEnTRE.TRENB bit is 0.
  This register indicates the number of currently counted transactions if the PIPEnTRE.TRENB bit is 1.

The USB increments the value of the PIPEnTRN register by one when all of the following conditions are satisfied on receiving the packet.

- The PIPEnTRE.TRENB bit = 1
- (PIPEnTRN set value ≠ current counter value + 1) on receiving the packet.
- The payload of the received packet agrees with the setting of the PIPEMAXP.MXPS[8:0] bits.

The USB sets the value of the PIPEnTRN register to 0000h when any of the following conditions are satisfied.

(1) All of the following conditions are satisfied.
- The PIPEnTRE.TRENB bit = 1
- (PIPEnTRN set value = current counter value + 1) on receiving the packet.
- The payload of the received packet agrees with the setting of the PIPEMAXP.MXPS[8:0] bits.

(2) All of the following conditions are satisfied.
- The PIPEnTRE.TRENB bit = 1
- The USB has received a short packet.

(3) All of the following conditions are satisfied.
- The PIPEnTRE.TRENB bit = 1
- The PIPEnTRE.TRCLR bit has been set to 1 by software.

For the pipe in the transmitting direction, set the PIPEnTRN register to 0000h.
When the transaction counter is not used, set the PIPEnTRN register to 0000h.
Setting the number of transactions to be transferred to the PIPEnTRN register 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.

34.2.35 Device Address n Configuration Register (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, USB1.DEVADD0 000A 02D0h, USB1.DEVADD1 000A 02D2h, USB1.DEVADD2 000A 02D4h, USB1.DEVADD3 000A 02D6h, USB1.DEVADD4 000A 02D8h, USB1.DEVADD5 000A 02DAh

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

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7, b6	USBSPD[1:0]	Transfer Speed of Communication Target Device	b7 b6 0 0: DEVADDn is not used 0 1: Low-speed 1 0: Full-speed 1 1: Setting prohibited	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The DEVADDn register specifies the transfer speed of the peripheral device which is the communication target for pipes 0 to 9.
When the host controller is selected, the bits in the DEVADDn register should be set before starting communication using each pipe.
The bits in the DEVADDn register should be modified while no valid pipes are using the settings of the bits. Valid pipes refer to the pipes satisfying both of the following conditions:

- The DEVADDn register is selected by the DEVSEL[3:0] bits.
- The PID[1:0] bits are set to 01b (BUF) for the selected pipe or the selected pipe is the DCP with the DCPCTR.SUREQ bit set to 1.

USBSPD[1:0] Bits (Transfer Speed of Communication Target Device)

The USBSPD[1:0] bits specify the USB transfer speed of the communication target peripheral device.
Set these bits to 10b when a full-speed device is connected via the HUB.
When the host controller is selected, the USB refers to the setting of the USBSPD[1:0] bits to generate packets.
When the function controller is selected, set these bits to 00b.

34.2.36 PHY Cross Point Adjustment Register (PHYSLEW)

Address(es): USB0.PHYSLEW 000A 00F0h, USB1.PHYSLEW 000A 02F0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	SLEWF01	SLEWF00	SLEWR01	SLEWR00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	SLEWR00	Driver Cross Point Adjustment 00	Set this bit to 1.	R/W
b1	SLEWR01	Driver Cross Point Adjustment 01	Set this bit to 0.	R/W
b2	SLEWF00	Driver Cross Point Adjustment 00	Set this bit to 1.	R/W
b3	SLEWF01	Driver Cross Point Adjustment 01	Set this bit to 0.	R/W
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The PHYSLEW register adjusts the cross point of the driver.
Set the value (00000005h) to the register before starting the USB operation.

34.2.37 Deep Standby USB Transceiver Control/Pin Monitoring Register (DPUSR0R)

Address(es): USB.DPUSR0R 000A 0400h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DVBSTS0	—	DOVCB0	DOVCA0	—	—	DM0	DP0
Value after reset:	0	0	0	0	0	0	0	0	x	0	x	x	0	0	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	FIXPHY0	DRPD0	—	RPUE0	SRPC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	SRPC0	USB0 Single End Receiver Control	0: Input through the D+ and D– inputs is disabled. 1: Input through the D+ and D– inputs is enabled.	R/W
b1	RPUE0*1	USB0 D+ Pull-Up Resistor Control	0: Disables D+ pull-up resistor. 1: Enables D+ pull-up resistor.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	DRPD0*1	USB0 D+/D– Pull-Down Resistor Control	0: Disables D+/D– pull-down resistor. 1: Enables D+/D– pull-down resistor.	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
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	DP0	USB0 D+ Input Flag	Indicates the D+ input signal of the USB0.	R
b17	DM0	USB0 D– Input Flag	Indicates the D– input signal of the USB0.	R
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20	DOVCA0	USB0 OVRCURA Input Flag	Indicates the OVRCURA input signal of the USB0.	R
b21	DOVCB0	USB0 OVRCURB Input Flag	Indicates the OVRCURB input signal of the USB0.	R
b22	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b23	DVBSTS0	USB0 VBUS Input Flag	Indicates the VBUS input signal of the USB0.	R
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Use this bit during operation in deep standby mode. For details, refer to section 34.3.1.5, Release from Deep Software Standby Mode Due to USB Suspend/Resume Interrupts.

SRPC0 Bit (USB0 Single End Receiver Control)

The SRPC0 bit controls the D+ and D– inputs of the USB0 transceiver.

This bit is only valid when the FIXPHY0 bit is 1.

FIXPHY0 Bit (USB0 Transceiver Output Fix)

The FIXPHY0 bit keeps the outputs of the USB0 transceiver disabled.

34.2.38 Deep Standby USB Suspend/Resume Interrupt Register (DPUSR1R)

Address(es): USB.DPUSR1R 000A 0404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DVBINT0	—	DOVRCRB0	DOVRCRA0	—	—	DMINT0	DPINT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	DVBSE0	—	DOVRCRBE0	DOVRCRAE0	—	—	DMINTE0	DPINTE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DPINTE0	USB0 D+ Interrupt Enable/Clear	0: Recovery from deep software standby mode is disabled. 1: Recovery from deep software standby mode is enabled.	R/W
b1	DMINTE0	USB0 D– Interrupt Enable/Clear	0: Recovery from deep software standby mode is disabled. 1: Recovery from deep software standby mode is enabled.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DOVRCRAE0	USB0 OVRCURA Interrupt Enable/Clear	0: Recovery from deep software standby mode is disabled. 1: Recovery from deep software standby mode is enabled.	R/W
b5	DOVRCRBE0	USB0 OVRCURB Interrupt Enable/Clear	0: Recovery from deep software standby mode is disabled. 1: Recovery from deep software standby mode is enabled.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	DVBSE0	USB0 VBUS Interrupt Enable/Clear	0: Recovery from deep software standby mode is disabled. 1: Recovery from deep software standby mode is enabled.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	DPINT0	USB0 D+ Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode. 1: The system has returned from deep software standby mode.	R
b17	DMINT0	USB0 D– Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode. 1: The system has returned from deep software standby mode.	R
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20	DOVRCRA0	USB0 OVRCURA Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode. 1: The system has returned from deep software standby mode.	R
b21	DOVRCRB0	USB0 OVRCURB Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode. 1: The system has returned from deep software standby mode.	R
b22	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b23	DVBINT0	USB0 VBUS Interrupt Source Recovery Flag	0: The system has not returned from deep software standby mode. 1: The system has returned from deep software standby mode.	R
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DPINTE0 Bit (USB0 D+ Interrupt Enable/Clear)

This bit enables or disables triggering of recovery from deep software standby mode by the D+ input of the USB0. Writing 0 to the DPINTE0 bit while the DPINT0 flag is 1 sets the DPINT0 flag to 0.

DMINTE0 Bit (USB0 D– Interrupt Enable/Clear)

This bit enables or disables triggering of recovery from deep software standby mode by the D– input of the USB0. Writing 0 to the DMINTE0 bit while the DMINT0 flag is 1 sets the DMINT0 flag to 0.

DOVRCRAE0 Bit (USB0 OVRCURA Interrupt Enable/Clear)

This bit enables or disables triggering of recovery from deep software standby mode by the OVRCURA input of the USB0. Writing 0 to the DOVRCRAE0 bit while the DOVRCRA0 flag is 1 sets the DOVRCRA0 flag to 0.

DOVRCRBE0 Bit (USB0 OVRCURB Interrupt Enable/Clear)

This bit enables or disables triggering of recovery from deep software standby mode by the OVRCURB input of the USB0. Writing 0 to the DOVRCRBE0 bit while the DOVRCRB0 flag is 1 sets the DOVRCRB0 flag to 0.

DVBSE0 Bit (USB0 VBUS Interrupt Enable/Clear)

This bit enables or disables triggering of recovery from deep software standby mode by the VBUS input of the USB0. Writing 0 to the DVBSE0 bit while the DVBINT0 flag is 1 sets the DVBINT0 flag to 0.

DPINT0 Flag (USB0 D+ Interrupt Source Recovery Flag)

This bit indicates that the system has returned from deep software standby mode due to the D+ input of the USB0. Recovery from deep software standby mode due to D+ input is only enabled when the DPINTE0 bit is 1.
Writing 0 to the DPINTE0 bit while the DPINT0 flag is 1 sets the DPINT0 flag to 0.

DMINT0 Flag (USB0 D– Interrupt Source Recovery Flag)

This bit indicates that the system has returned from deep software standby mode due to the D– input of the USB0. Recovery from deep software standby mode due to D– input is only enabled when the DMINTE0 bit is 1.
Writing 0 to the DPINTE0 bit while the DMINT0 flag is 1 sets the DMINT0 flag to 0.

DOVRCRA0 Flag (USB0 OVRCURA Interrupt Source Recovery Flag)

This bit indicates that the system has returned from deep software standby mode due to the OVRCURA input of the USB0. Recovery from deep software standby mode due to OVRCURA input is only enabled when the DOVRCRAE0 bit is 1.
Writing 0 to the DOVRCRAE0 bit while the DOVRCRA0 flag is 1 sets the DOVRCRA0 flag to 0.

DOVRCRB0 Flag (USB0 OVRCURB Interrupt Source Recovery Flag)

This bit indicates that the system has returned from deep software standby mode due to the OVRCURB input of the USB0. Recovery from deep software standby mode due to OVRCURB input is only enabled when the DOVRCRBE0 bit is 1.
Writing 0 to the DOVRCRBE0 bit while the DOVRCRB0 flag is 1 sets the DOVRCRB0 flag to 0.

DVBINT0 Flag (USB0 VBUS Interrupt Source Recovery Flag)

This indicates that the system has returned from deep software standby mode due to the VBUS input of the USB0. Recovery from deep software standby mode due to VBUS input is only enabled when the DVBSE0 bit is 1.
Writing 0 to the DVBSE0 bit while the DVBINT0 flag is 1 sets the DVBINT0 flag to 0.

34.3 Operation

34.3.1 System Control

This section describes the register settings that are necessary for initialization of this module and power consumption control.

34.3.1.1 Setting Data to the USB Related Register

Setting the SYSCFG.USBE bit to 1 after starting the clock supply to the USB (SYSCFG.SCKE bit = 1) enables and starts USB operation.

34.3.1.2 Controller Function Selection

For the USB, the host or function controller can be selected using the SYSCFG.DCFM bit. Note that the DCFM bit should be modified in the initial settings immediately after a reset is released or when pull-up resistor of the D+line and pull-down resistors of the D+ and D– lines are disabled (the SYSCFG.DPRPU bit = 0 and DRPD bit = 0).

34.3.1.3 Controlling USB Data Bus Resistors

The USB has pull-up and pull-down resistors for the D+ and D– lines. Pull up or pull down these lines by setting the SYSCFG.DPRPU and DRPD bits.
When the function controller is selected, confirm that connection to the USB host is made, then set the DPRPU bit to 1 and pull up the D+ line (during full-speed).
When the DPRPU bit is set to 0 during communication with the PC, the USB module disables the pull-up resistor of the USB data line, thus notifying the USB host of disconnection.
When the host controller is selected, set the DRPD bit and pull down the D+ and D– lines.

Table 34.11 USB Data Bus Resistor Control

SYSCFG register				
DRPD bit	**DPRPU bit**	**D–**	**D+**	**Function**
0	0	Open	Open	Not in use
0	1	Open	Pull-up	When operating as the function controller (in full-speed)
1	0	Pull-down	Pull-down	When operating as the host controller
1	1	—	—	Setting prohibited

34.3.1.4 Example of USB External Connection Circuit

Figure 34.2 shows an example of OTG connection of the USB connector in the self-powered state.
The USB controls the signals for enabling a pull-up resistor for the D+ signal and pull-down resistors for the D+ and D– signals. These signals can be pulled up or down using the SYSCFG.DPRPU and DRPD bits.
When the function controller is selected and the DPRPU bit is set to 0 during communication with the host controller, the pull-up resistor of the USB data line is disabled, making it possible to notify the USB host of the device disconnection.

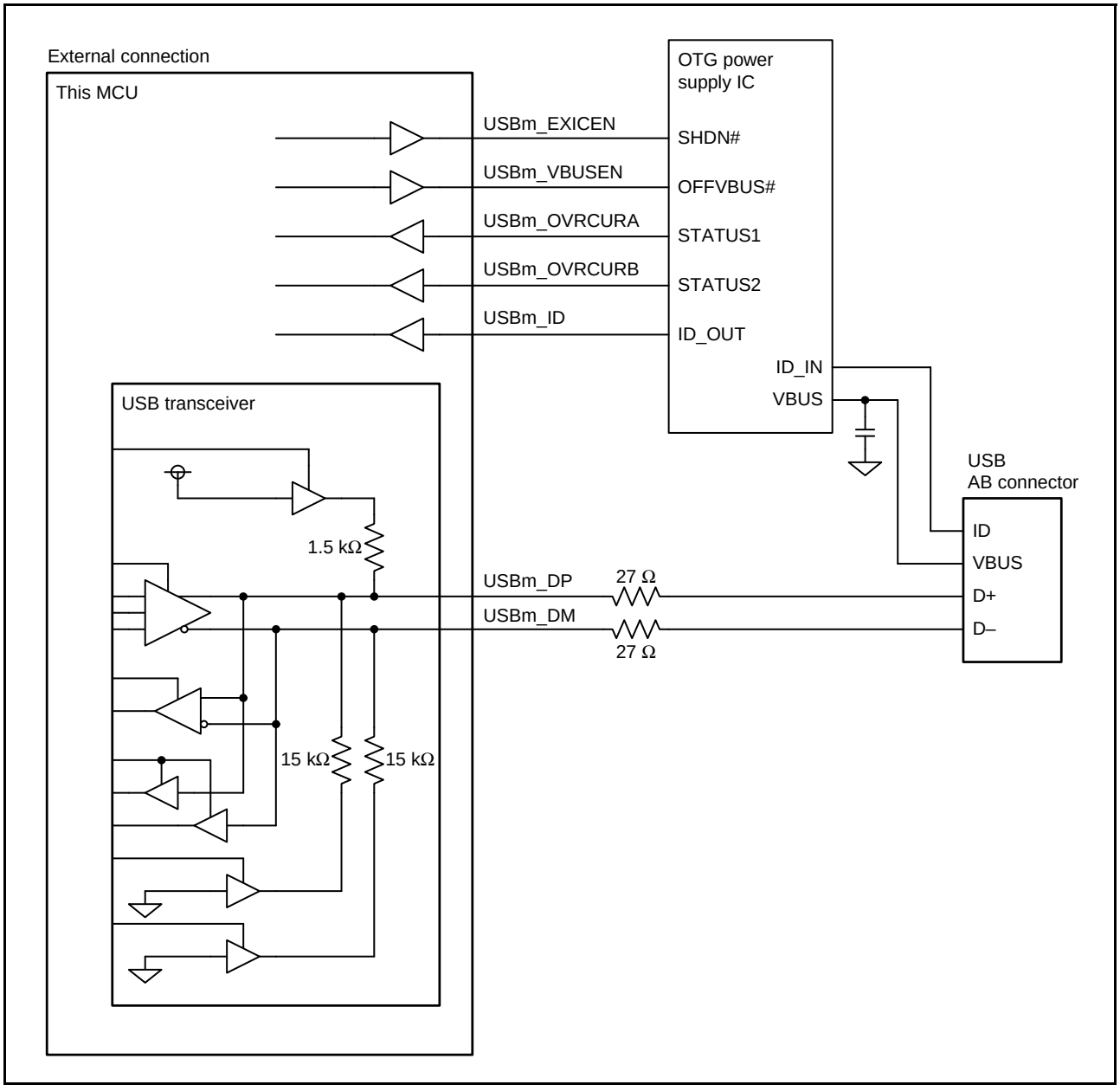


Figure 34.2 Sample OTG Connection of USB Connector in Self-Powered State (m = 0, 1)

Figure 34.3 shows an example of functional connection of the USB connector in the self-powered state.

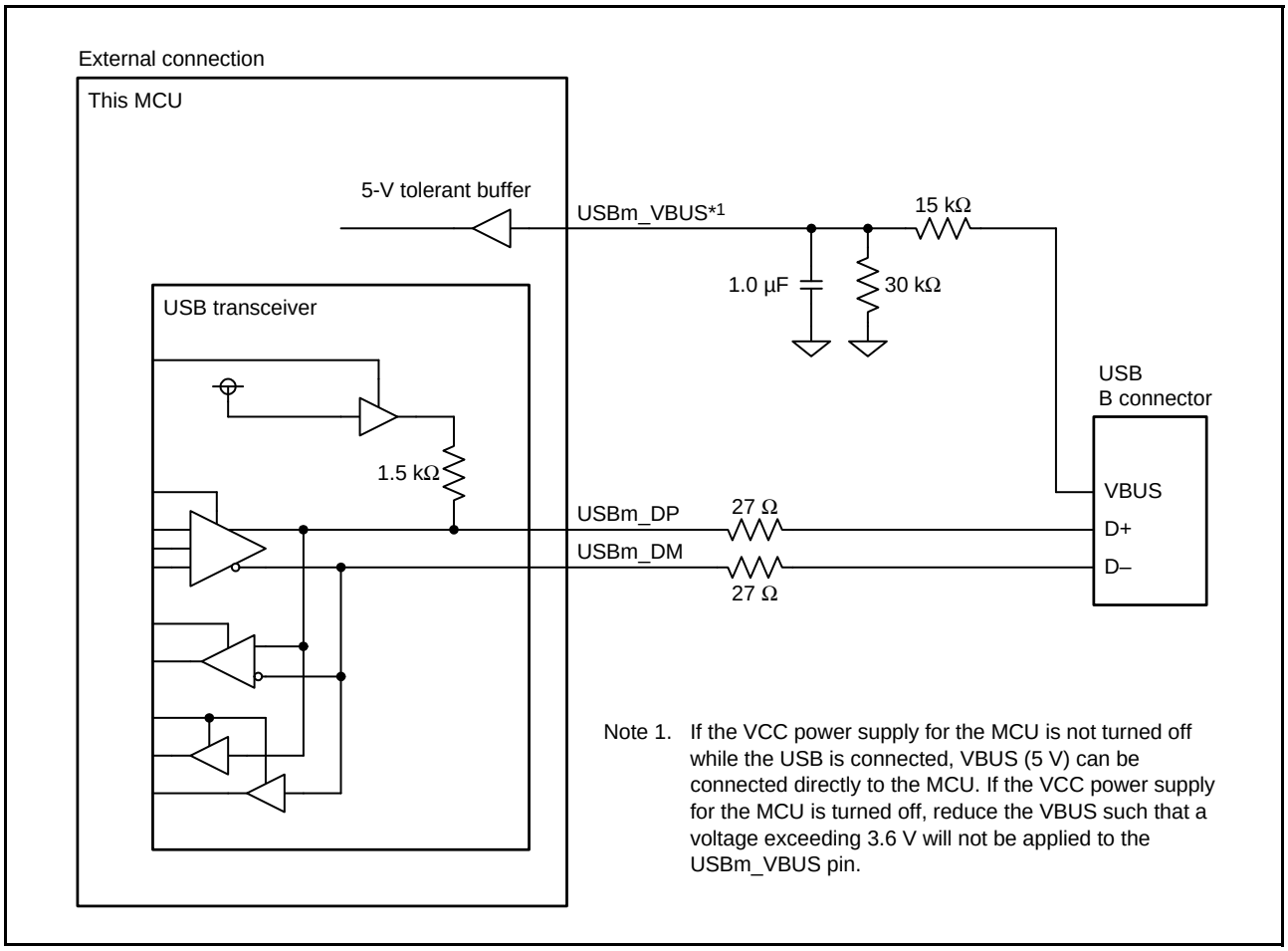


Figure 34.3 Functional Connection of USB Connector in Self-Powered State (m = 0, 1)

Figure 34.4 shows an example of host connection of the USB connector.

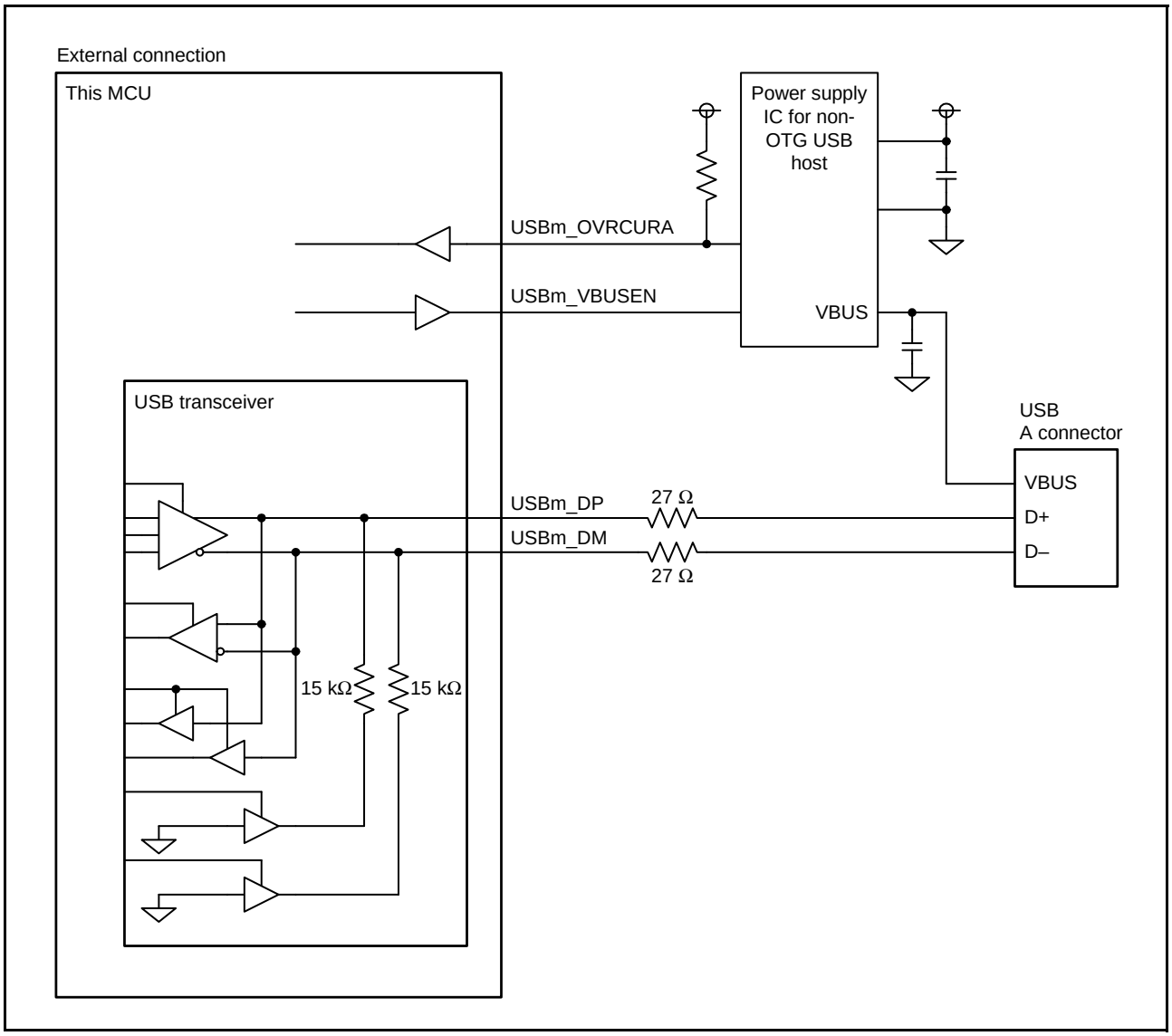


Figure 34.4 Sample Host Connection of USB Connector (m = 0, 1)

Figure 34.5 shows an example of functional connection of the USB connector in bus powered state.

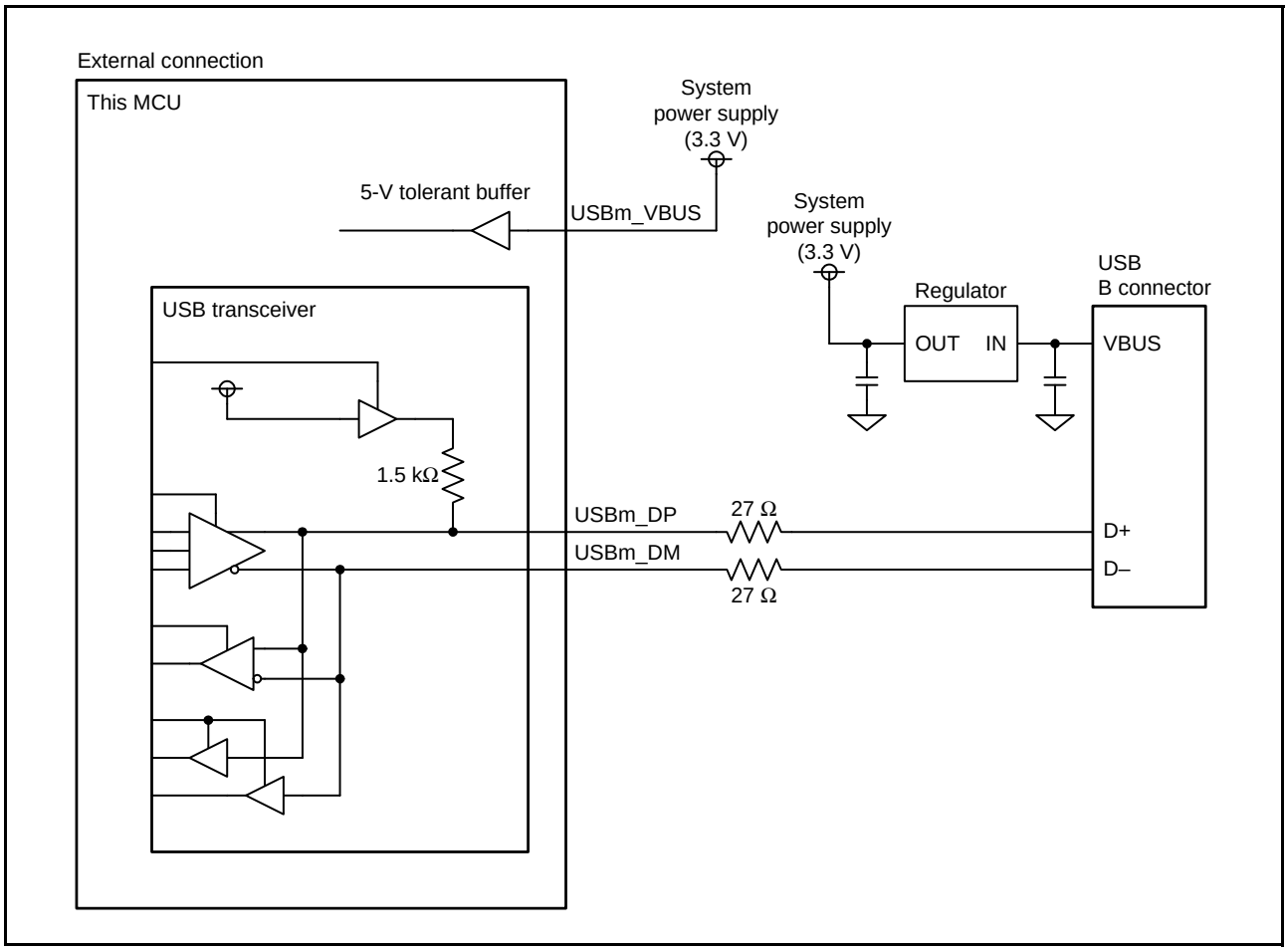


Figure 34.5 Functional Connection Sample of USB Connector in Bus Powered State (m = 0, 1)

The examples of external circuits given in this section are simplified circuits, and their operation in every system is not guaranteed.

34.3.1.5 Release from Deep Software Standby Mode Due to USB Suspend/Resume Interrupts

Deep software standby mode can be canceled by a USB0 suspend/resume interrupt.
A USB0 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 to detect USB0 suspend/resume interrupts.
Figure 34.6 shows a schematic diagram of connection between the USB resume detecting unit and the I/O pins for USB0.

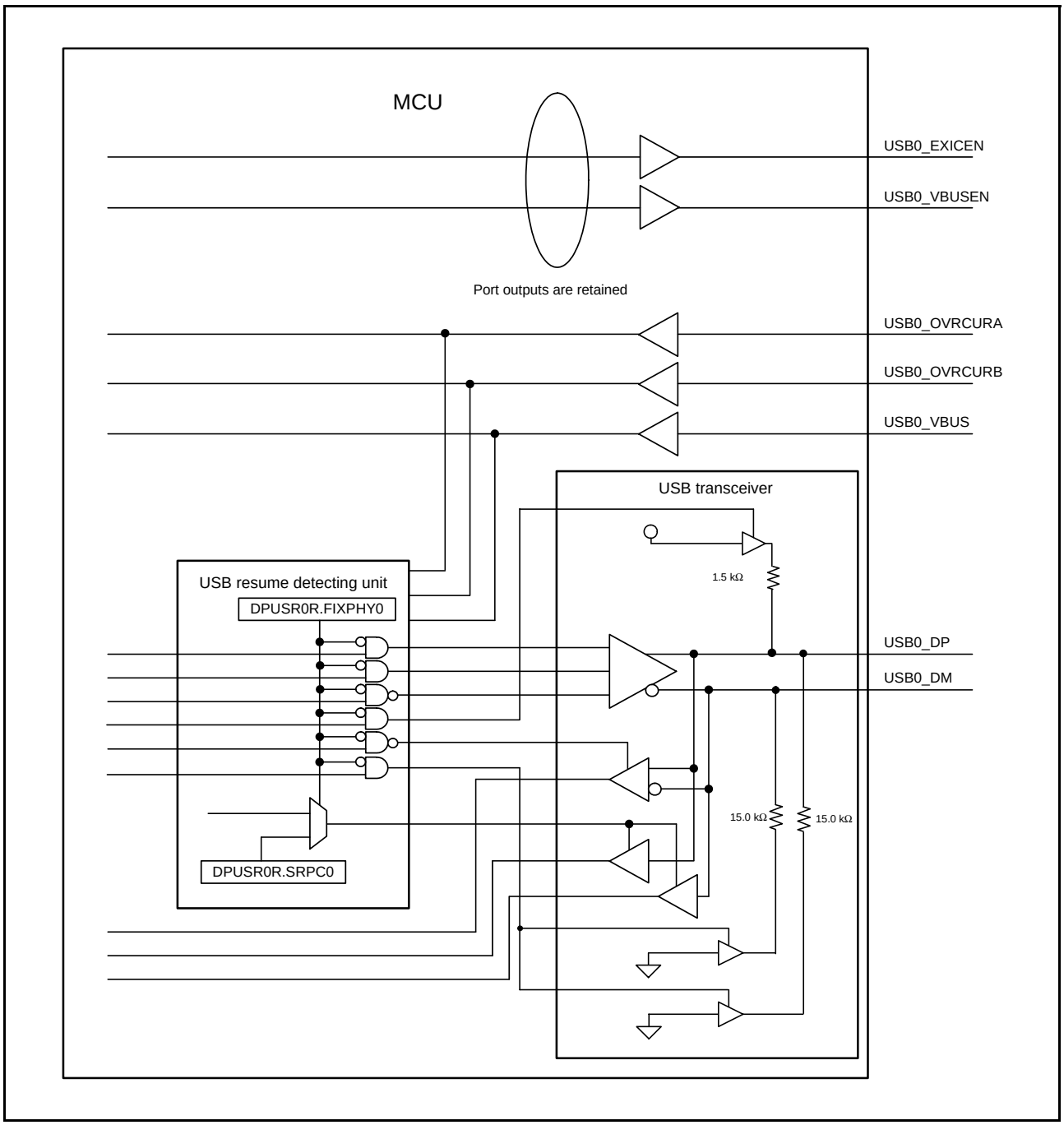


Figure 34.6 Overview of Connection between USB Resume Detecting Unit and USB0 I/O Pins

Table 34.12 shows the USB0 suspend/resume interrupt sources and their corresponding I/O pins.

Table 34.12 USB0 Suspend/Resume Interrupt Sources and Corresponding I/O Pins

USB Operating Mode	Source	Pin Name
Function/OTG	Resume	USB0_DP
Host/OTG	Connection/Disconnection	USB0_DP/USB0_DM
Function	Connection/Disconnection	USB0_VBUS
Host	Overcurrent detection	USB0_OVRCURA
OTG	Overcurrent detection	USB0_OVRCURA/USB0_OVRCURB

Figure 34.7 shows a flowchart for setting the USB when entering deep software standby mode while the host controller or function controller is selected. Figure 34.8 shows a flowchart for setting the USB0 when canceling deep software standby mode while the host controller is selected. Figure 34.9 shows a flowchart for setting the USB0 when canceling deep software standby mode while the function controller is selected.

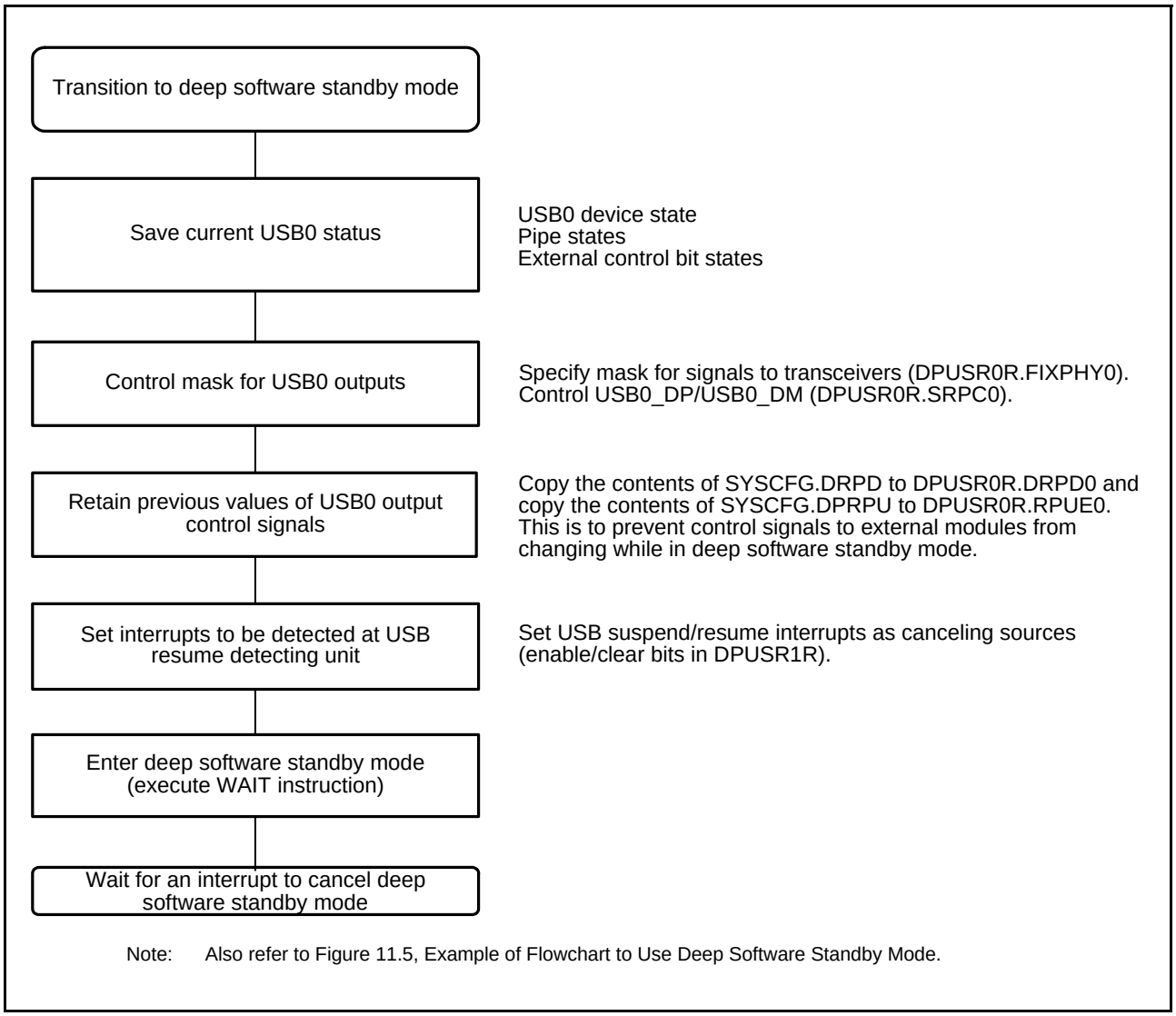
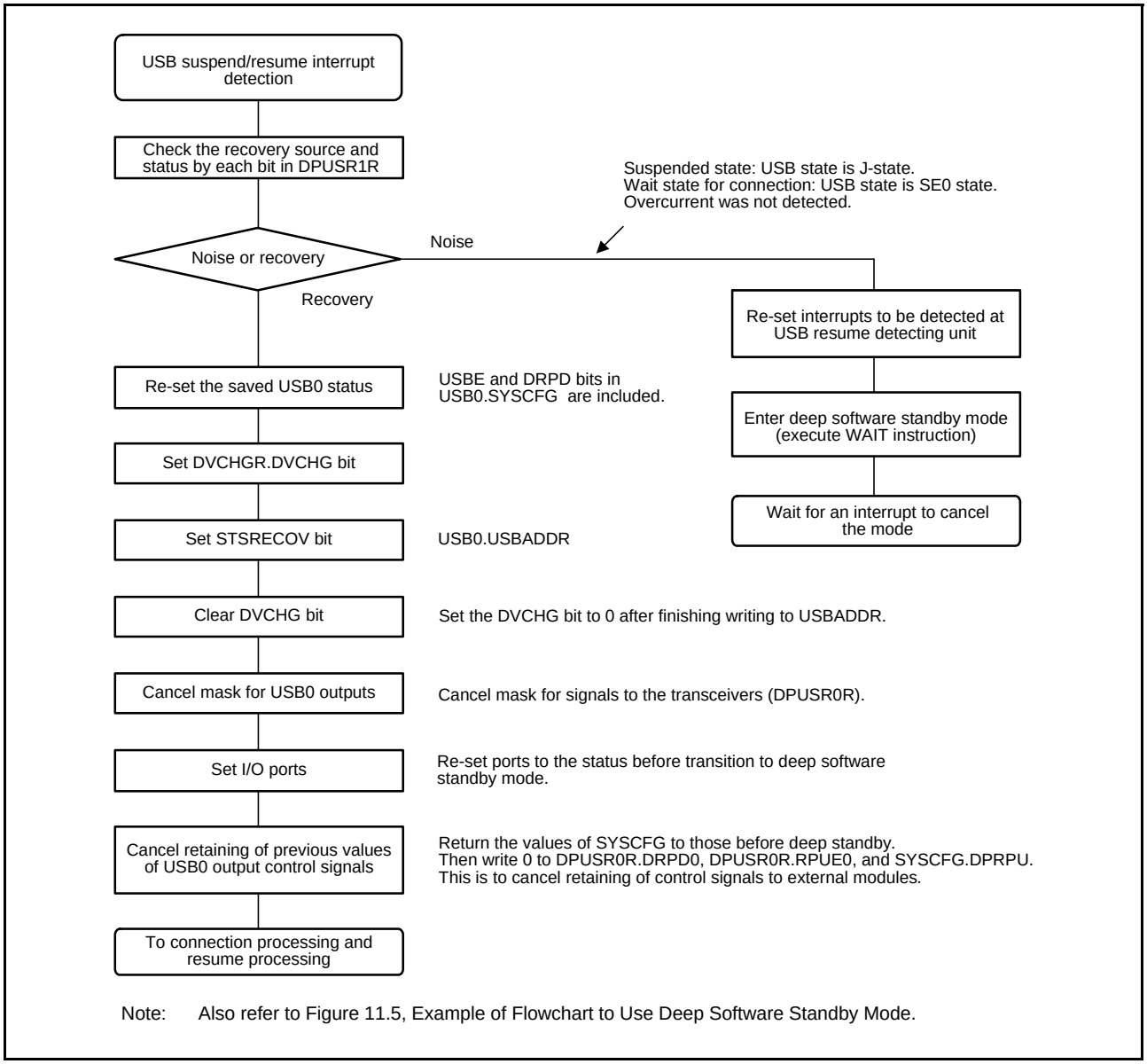
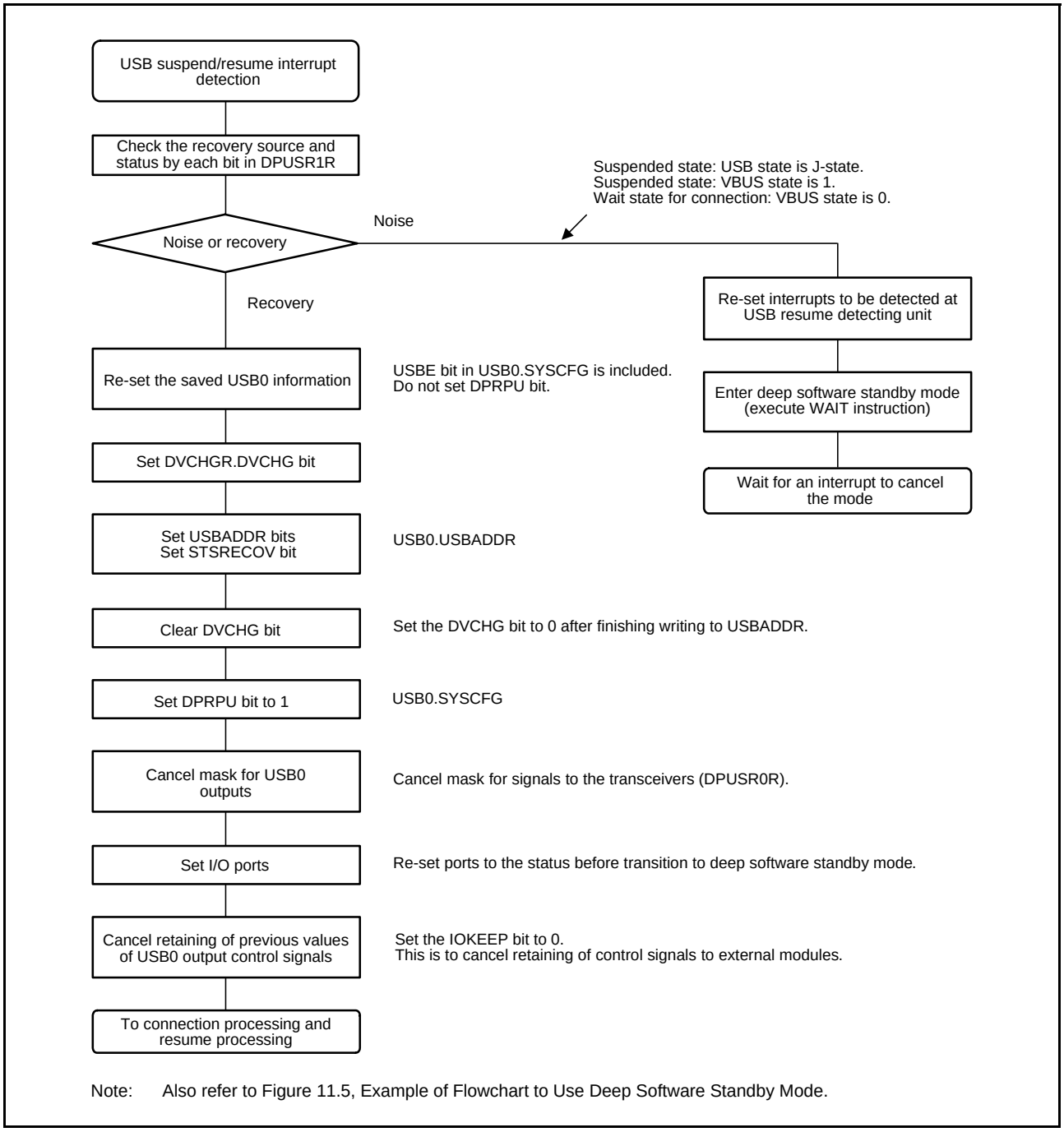


Figure 34.7 USB Setting Flowchart for Transition to Deep Software Standby Mode as Host/Function Controller



Note: Also refer to Figure 11.5, Example of Flowchart to Use Deep Software Standby Mode.

Figure 34.8 USB Setting Flowchart for Canceling Deep Software Standby Mode as Host Controller



Note: Also refer to Figure 11.5, Example of Flowchart to Use Deep Software Standby Mode.

Figure 34.9 USB Setting Flowchart for Canceling Deep Software Standby Mode as Function Controller

34.3.2 Interrupt Sources

Table 34.13 lists the interrupt sources in the USB.

When an interrupt generation condition is satisfied and the interrupt output is enabled using the corresponding interrupt enable register, a USB interrupt request is issued the Interrupt Controller (ICU) and an USB interrupt will be generated.

Table 34.13 Interrupt Sources (m = 0, 1)

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 is selected] • When an SOF packet with a different frame number has been transmitted [Function controller is selected] • When an SOF packet with a different frame number has been received	Host/function	—
DVST	Device state transition interrupt	• When a device state transition has been detected (any of the following conditions) A USB bus reset detected Suspended state detected SET_ADDRESS request received SET_CONFIGURATION request received	Function	INTSTS0. DVSQ[2:0]
CTRT	Control transfer stage transition interrupt	• When a stage transition has been detected in control transfer (any of the following conditions) Setup stage completed Control write transfer status stage transition Control read transfer status stage transition Control transfer completed A control transfer sequence error occurred	Function	INTSTS0. CTSQ[2:0]
BEMP	Buffer empty interrupt	• When transmission of all data in the buffer memory has been completed and the buffer has become empty • When a packet larger than the maximum packet size has been received	Host/function	BEMPSTS. PIPEnBEMP
NRDY	Buffer not ready interrupt	[Host controller is selected] • When STALL has been received from the peripheral device for the issued token • When a response has not been received correctly from the peripheral device for the issued token (no response was returned three consecutive times or a packet reception error occurred three consecutive times) • When an overrun/underrun occurred during isochronous transfer [Function controller is selected] • When NAK has been returned for an IN or OUT token while the PID[1:0] bits are 01b (BUF) • When a CRC error or a bit stuffing error occurred during data reception in isochronous transfer • When an overrun/underrun occurred during data reception in isochronous transfer	Host/function	NRDYSTS. PIPEnNRDY
BRDY	Buffer ready interrupt	• When the buffer has become ready (reading or writing is enabled)	Host/function	BRDYSTS. PIPEnBRDY
OVRCR	Overcurrent input change interrupt	• When a change in the state of the USBm_OVRCURA or USBm_OVRCURB input pin has been detected (low to high or high to low)	Host	INTSTS1. OVRCR
BCHG	Bus change interrupt	• When a change of USB bus state has been detected	Host/function	SYSSTS0. LNST[1:0]
DTCH	Disconnection detection during full-speed operation	• When disconnection of a peripheral device has been detected in full-speed operation	Host	DVSTCTR0. RHST[2:0]
ATTCH	Device connection detection	• When J-state or K-state is detected on the USB port for 2.5 µs. Used for checking whether a peripheral device is connected.	Host	—
EOFERR	EOF error detection	• When an EOF error of a peripheral device has been detected	Host	—
SACK	Normal setup operation	• When the normal response (ACK) for the setup transaction has been received	Host	—
SIGN	Setup error	• When a setup transaction error (no response or ACK packet corruption) was detected three consecutive times	Host	—

Note 1. Though this interrupt can be generated while the host function is selected, it is not usually used with the host function.

Figure 34.10 and Figure 34.11 show the circuits related to the interrupts in the USB0 and USB1, respectively.

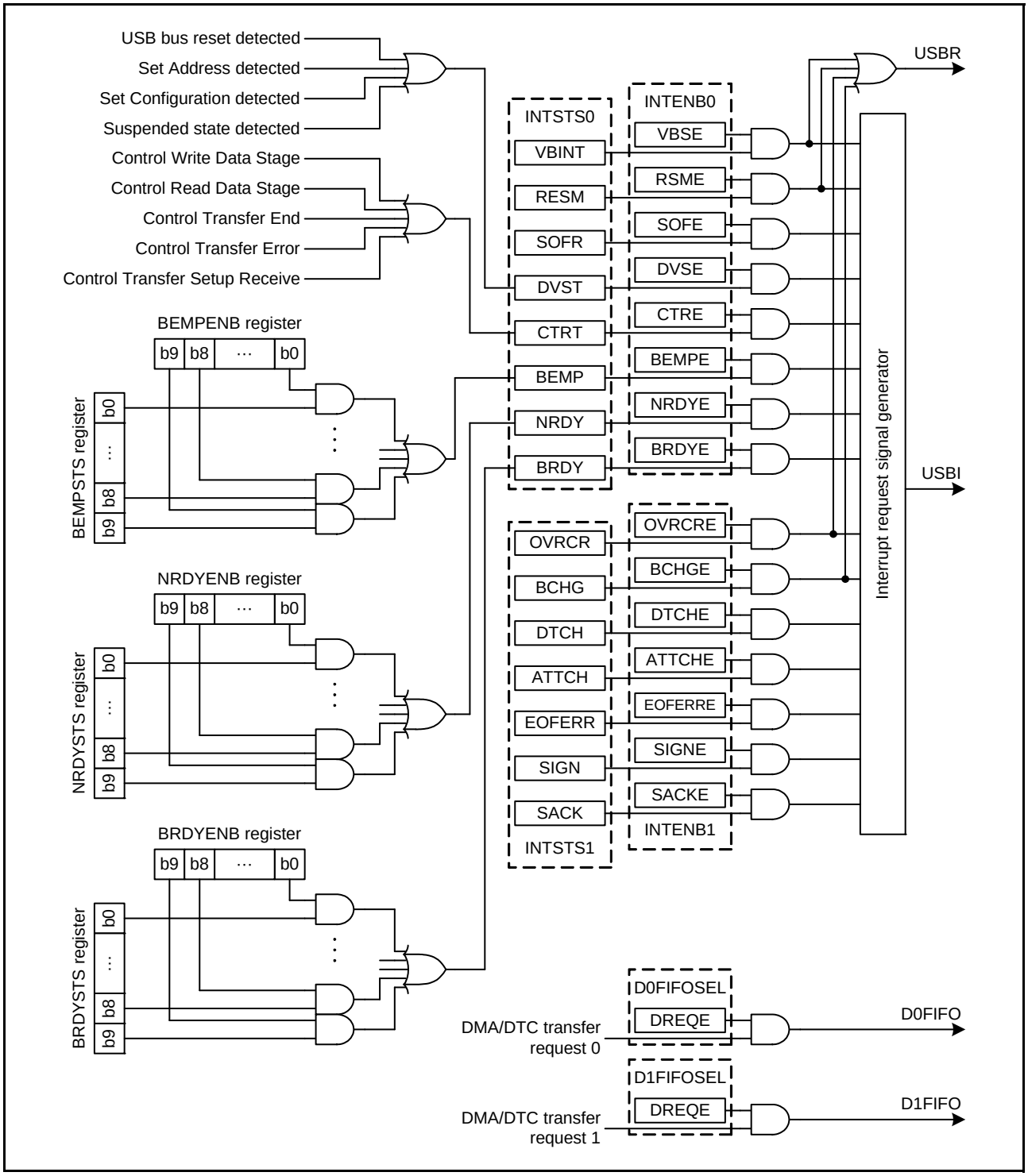


Figure 34.10 Circuits Related to Interrupts in USB0

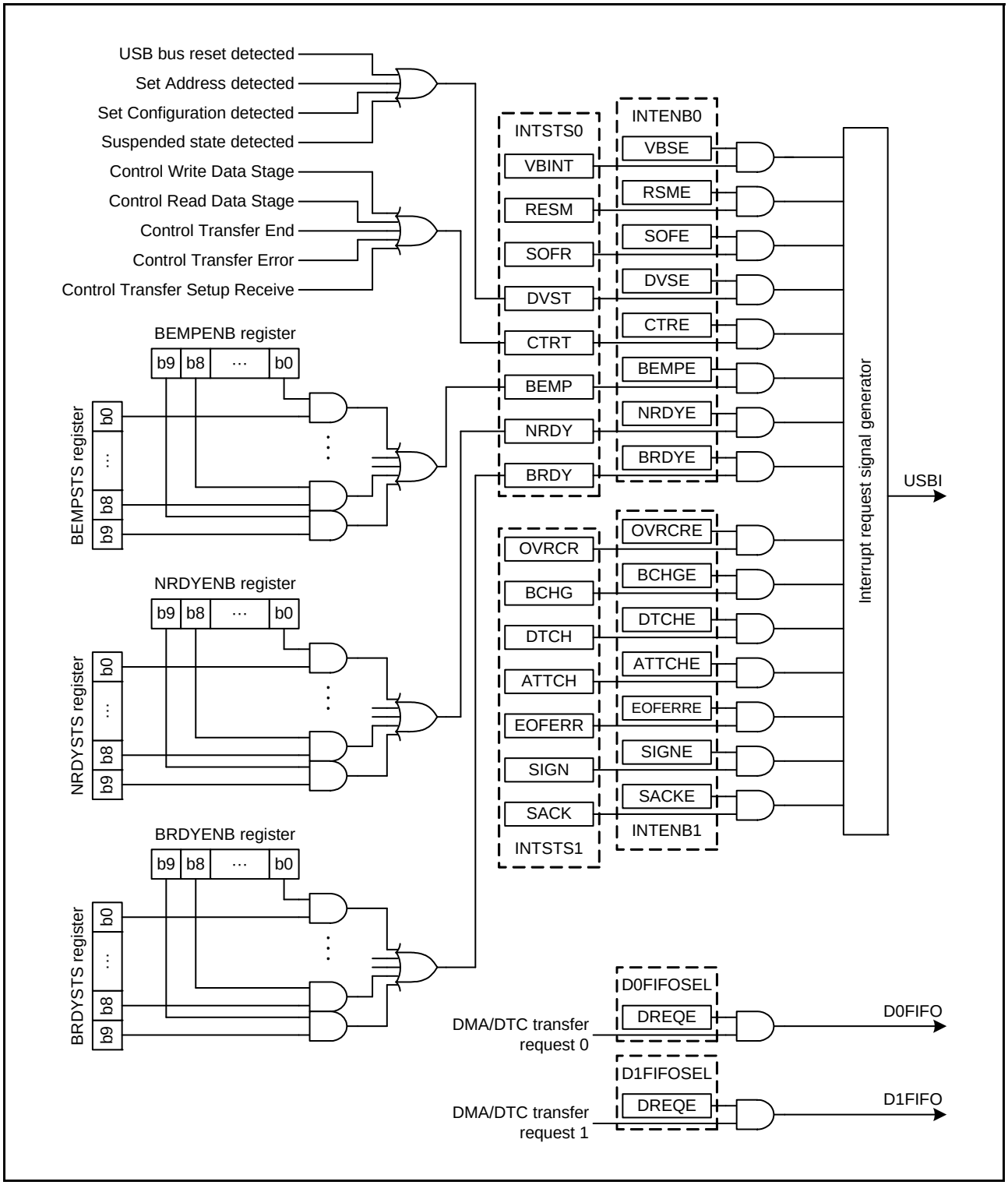


Figure 34.11 Circuits Related to Interrupts in USB1

Table 34.14 shows the interrupts generated in the USB.

Table 34.14 USB Interrupts

Interrupt Name	Interrupt Status Flag	DTC Activation	DMAC Activation
D0FIFO	DMA/DTC transfer request 0	Possible	Possible
D1FIFO	DMA/DTC transfer request 1	Possible	Possible
USBI	VBUS interrupt, resume interrupt, frame number update interrupt, device state transition interrupt, control transfer stage transition interrupt, buffer empty interrupt, buffer not ready interrupt, buffer ready interrupt, overcurrent input change interrupt, bus change interrupt, disconnection detection during full-speed operation, device connection detection, EOF error detection, normal setup operation, and setup error	Not possible	Not possible
USBR*1	VBUS interrupt, resume interrupt, overcurrent input change interrupt, and bus change interrupt	Not possible	Not possible

Note 1. This is not supported in USB1.

34.3.3 Interrupt Descriptions

34.3.3.1 BRDY Interrupt

The BRDY interrupt is generated when either of the host controller or function controller is selected. The following shows the conditions under which the USB sets 1 to a corresponding bit in the BRDYSTS register. Under this condition, the USB generates a BRDY interrupt if software has set 1 to the BRDYENB.PIPEnBRDYE bit that corresponds to the pipe and 1 to the INTENB0.BRDYE bit.
The conditions for generating and clearing the BRDY interrupt depend on the settings of the SOFCFG.BRDYM bit and PIPECFG.BFRE bit for each pipe as described below.

(1) When the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 0

With these settings, the BRDY interrupt indicates that the FIFO port is accessible.
On any of the following conditions, the USB generates an internal BRDY interrupt request trigger and sets 1 to the BRDYSTS.PIPEnBRDY flag corresponding to the pertinent pipe.

(a) For the pipe in the transmitting direction:

- When the DIR bit is changed from 0 to 1 by software.
- When packet transmission is completed using the pertinent pipe while write-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS flag is read as 0).
- When one FIFO buffer is empty on completion of writing data to the other FIFO buffer in double buffer mode.
- No request trigger is generated until completion of writing data to the currently-written FIFO buffer even if transmission to the other FIFO buffer is completed.
- When the hardware flushes the buffer of the pipe for isochronous transfers.
- When 1 is written to the PIPEnCTR.ACLRM bit, which causes the FIFO buffer to make transition from the write-disabled to write-enabled state.

No request trigger is generated for the DCP (that is, during data transmission for control transfers).

(b) For the pipe in the receiving direction:

- When packet reception is completed successfully thus enabling the FIFO buffer to be read while read-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the BSTS flag is read as 0).
  No request trigger is generated for the transaction in which data PID mismatch has occurred.
- When one FIFO buffer is read-enabled on completion of reading data from the other FIFO buffer in double buffer mode.
  No request trigger is generated until completion of reading data from the currently-read FIFO buffer even if reception by the other FIFO buffer is completed.

When the function controller is selected, the BRDY interrupt is not generated in the status stage of control transfers.
The BRDY interrupt status of the pertinent pipe can be set to 0 by writing 0 to the corresponding PIPEnBRDY flag through software. In this case, 1s should be written to the PIPEnBRDY flags for the other pipes.
Clear the BRDY status before accessing the FIFO buffer.

(2) When the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 1

With these settings, the USB generates a BRDY interrupt on completion of reading all data for a single transfer using the pipe in the receiving direction, and sets 1 to the bit in the BRDYSTS register corresponding to the pertinent pipe.
On any of the following conditions, the USB determines that the last data for a single transfer has been received.

- When a short packet including a zero-length packet is received.
- When the pipe n transaction counter register (PIPEnTRN) is used and the number of packets specified by the PIPEnTRN register are completely received.

When the pertinent data is completely read after any of the above conditions has been satisfied, the USB determines that all data for a single transfer has been completely read.
When a zero-length packet is received while the FIFO buffer is empty, the USB module determines that all data for a single transfer has been completely read when the FRDY flag in the FIFO port control register is 1 and the DTLN[8:0] flags are 0. In this case, to start the next transfer, write 1 to the BCLR bit in the corresponding port control register through software.
With these settings, the USB does not detect a BRDY interrupt for the pipe in the transmitting direction.
The BRDY interrupt status of the pertinent pipe can be set to 0 by writing 0 to the corresponding BRDYSTS.PIPEnBRDY flag through software. In this case, 1s should be written to the PIPEnBRDY flags for the other pipes.
In this mode, the PIPECFG.BFRE bit setting should not be modified until all data for a single transfer has been processed. When it is necessary to modify the PIPECFG.BFRE bit before completion of processing, all FIFO buffers for the pertinent pipe should be cleared using the PIPEnCTR.ACLRM bit.

(3) When the SOFCFG.BRDYM Bit = 1 and the PIPECFG.BFRE Bit = 0

With these settings, the BRDYSTS.PIPEnBRDY values are linked to the BSTS flag setting for each pipe. In other words, the BRDY interrupt status flags (PIPEnBRDY) are set to 1 or 0 by the USB depending on the FIFO buffer status.

(a) For the pipe in the transmitting direction:

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for write access, and are set to 0 when it is not ready.
However, the BRDY interrupt is not generated even if the DCP in the transmitting direction is ready for write access.

(b) For the pipe in the receiving direction:

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for read access, and are set to 0 when all data have been read (not ready for read access).
When a zero-length packet is received while the FIFO buffer is empty, the pertinent bit is set to 1 and the BRDY interrupt is continuously generated until BCLR = 1 is written through software.
With this setting, the PIPEnBRDY flag cannot be set to 0 through software.
When the SOFCFG.BRDYM bit is set to 1, all PIPECFG.BFRE bits (for all pipes) should be set to 0.

Figure 34.12 shows the timing of BRDY interrupt generation.

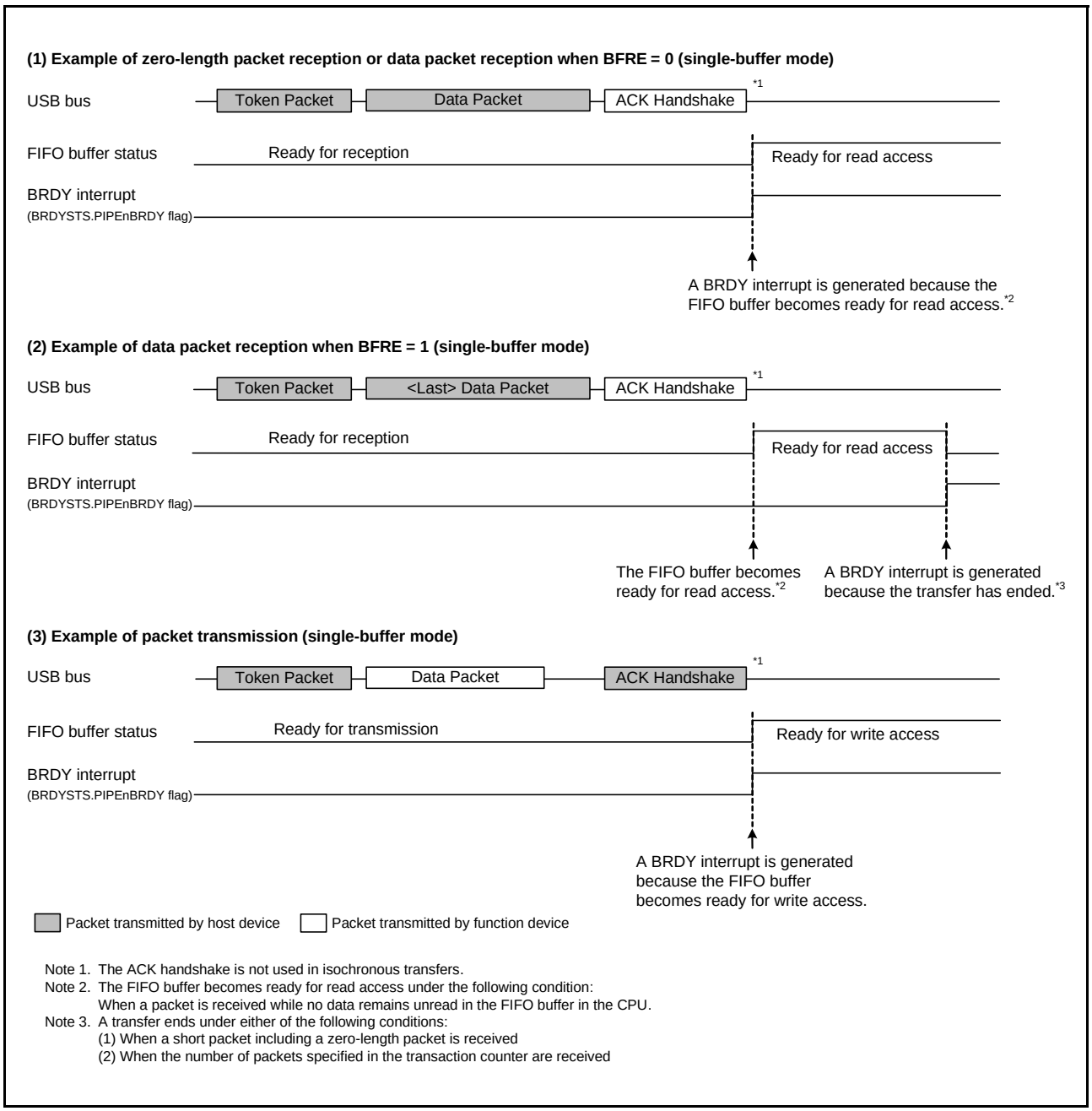


Figure 34.12 Timing of BRDY Interrupt Generation

The condition that USB clears the INTSTS0.BRDY flag depends on the SOFCFG.BRDYM bit setting. Table 34.15 shows the condition for clearing the BRDY flag.

Table 34.15 Condition for Clearing BRDY Flag

BRDYM Bit	Condition for Clearing BRDY Flag
0	The USB sets the BRDY flag to 0 when all bits in the BRDYSTS register have been set to 0 by software.
1	The USB sets the BRDY flag to 0 when the BSTS flags for all pipes have become 0.

34.3.3.2 NRDY Interrupt

On generating an internal NRDY interrupt request for the pipe whose PID[1:0] bits are set to 01b (BUF) by software, the USB sets the corresponding NRDYSTS.PIPEnNRDY flag to 1. If the corresponding bit in the NRDYENB register has been set to 1 by software, the USB sets the INTSTS0.NRDY flag to 1 and generates a USB interrupt.
The following describes the conditions on which the USB generates the internal NRDY interrupt request for a given pipe.
Note that the internal NRDY interrupt request is not generated during setup transaction execution when the host controller is selected. During setup transactions when the host controller is selected, the SACK or SIGN interrupt is detected.
The internal NRDY interrupt request is not generated during status stage execution of the control transfer when the function controller is selected.

(1) When Host Controller is Selected

(a) For the pipe in the transmitting direction:

On any of the following conditions, the USB detects an NRDY interrupt.

- For the pipe for isochronous transfers, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer.
  In this case, the USB transmits a zero-length packet following the OUT token and sets the bit corresponding to the NRDYSTS.PIPEnNRDY flag and the FRMNUM.OVRN flag to 1.
- During communications other than setup transactions using the pipe for the transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device (when timeout is detected before detection of the handshake packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.
  In this case, the USB sets the bit corresponding to the PIPEnNRDY flag to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 00b (NAK).
- During communications other than setup transactions, when the STALL handshake is received from the peripheral device.
  In this case, the USB sets the bit corresponding to the PIPEnNRDY flag to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 11b (STALL).

(b) For the pipe in the receiving direction:

- For the pipe for isochronous transfers, when the time to issue an IN token comes while there is no space available in the FIFO buffer.
  In this case, the USB discards the received data for the IN token and sets the PIPEnNRDY flag corresponding to the pipe and the OVRN flag to 1.
  When a packet error is detected in the received data for the IN token, the USB also sets the FRMNUM.CRCE flag to 1.
- For the pipe for the transfers other than isochronous transfers, when any combination of the following two cases occur three consecutive times: 1) no response is returned from the peripheral device for the IN token issued by the USB (when timeout is detected before detection of the DATA packet from the peripheral device) and 2) an error is detected in the packet from the peripheral device.
  In this case, the USB sets the PIPEnNRDY flag corresponding to the pipe to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 00b (NAK).
- For the pipe for isochronous transfers, when no response is returned from the peripheral device for the IN token (when timeout is detected before detection of the DATA packet from the peripheral device) or an error is detected in the packet from the peripheral device.
  In this case, the USB sets the PIPEnNRDY flag corresponding to the pipe to 1. (The setting of the PID[1:0] bits of the pipe is not modified.)

- For the pipe for isochronous transfers, when a CRC error or a bit stuffing error is detected in the received data packet.
  In this case, the USB sets the PIPEnNRDY flag corresponding to the pipe and the CRCE flag to 1.
- When the STALL handshake is received.
  In this case, the USB sets the PIPEnNRDY flag corresponding to the pipe to 1 and modifies the setting of the PID[1:0] bits of the corresponding pipe to 11b (STALL).

(2) When Function Controller is Selected

(a) For the pipe in the transmitting direction:

- When an IN token is received while there is no data to be transmitted in the FIFO buffer.
  In this case, the USB generates a NRDY interrupt request at the reception of the IN token and sets the NRDYSTS.PIPEnNRDY flag to 1.
  For the pipe for the isochronous transfers in which an interrupt is generated, the USB transmits a zero-length packet and sets the FRMNUM.OVRN flag to 1.

(b) For the pipe in the receiving direction:

- When an OUT token is received while there is no space available in the FIFO buffer.
  For the pipe for the isochronous transfers in which an interrupt is generated, the USB generates a NRDY interrupt request at the reception of the OUT token and sets the PIPEnNRDY flag to 1 and OVRN flag to 1.
  For the pipe for the transfers other than isochronous transfers in which an interrupt is generated, the USB generates a NRDY interrupt request when a NAK handshake is transferred after the data following the OUT token is received, and sets the PIPEnNRDY flag to 1.
  However, during re-transmission (due to data PID mismatch), the NRDY interrupt request is not generated. In addition, if an error occurs in the DATA packet, the NRDY interrupt request is not generated.
- For the pipe for isochronous transfers, when a token is not received successfully within an interval frame.
  In this case, the USB generates a NRDY interrupt request when SOF is received, and sets the PIPEnNRDY flag to 1.

Figure 34.13 shows the timing of NRDY interrupt generation when the function controller is selected.

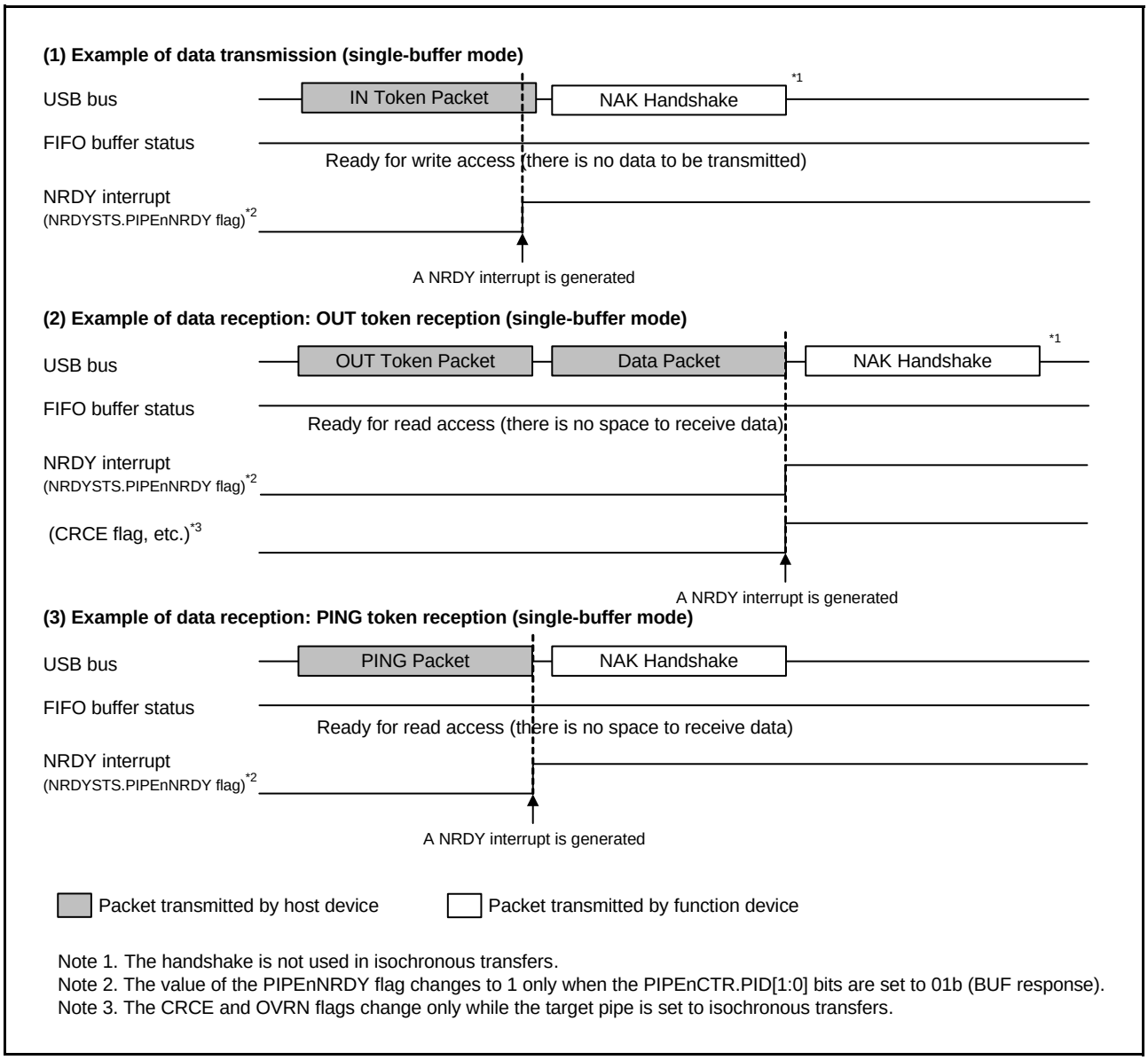


Figure 34.13 Timing of NRDY Interrupt Generation (When Function Controller is Selected)

34.3.3.3 BEMP Interrupt

On detecting a BEMP interrupt for the pipe whose PID[1:0] bits are set to 01b (BUF) by software, the USB sets the corresponding BEMPSTS.PIPEnBEMP flag to 1. If the corresponding bit in the BEMPENB register has been set to 1 by software, the USB sets the INTSTS0.BEMP flag to 1 and generates a USB interrupt.

The following describes the conditions on which the USB generates an internal BEMP interrupt request.

(1) For the pipe in the transmitting direction:

When the FIFO buffer of the corresponding pipe is empty on completion of transmission (including zero-length packet transmission).
In single buffer mode, an internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for the pipe other than DCP. However, the internal BEMP interrupt request is not generated on any of the following conditions.

- When the CPU or DMAC/DTC has already started writing data to the FIFO buffer of the CPU on completion of transmitting data from one FIFO buffer in double buffer mode.
- When the buffer is cleared (emptied) by setting the PIPEnCTR.ACLRM or the BCLR bit in the port control register to 1.
- When IN transfer (zero-length packet transmission) is performed during the control transfer status stage while the function controller is selected.

(2) For the pipe in the receiving direction:

When the successfully-received data packet size exceeds the specified maximum packet size.
In this case, the USB generates a BEMP interrupt request, sets the corresponding BEMPSTS.PIPEnBEMP flag to 1, discards the received data, and modifies the setting of the PID[1:0] bits of the corresponding pipe to 11b (STALL). Here, the USB returns no response when used as the host controller, and returns STALL response when used as the function controller.
However, the internal BEMP interrupt request is not generated on any of the following conditions.

- When a CRC error or a bit stuffing error is detected in the received data.
- When a setup transaction is being performed,
  Writing 0 to the BEMPSTS.PIPEnBEMP flag clears the status.
  Writing 1 to the BEMPSTS.PIPEnBEMP flag has no effect.

Figure 34.14 shows the timing of BEMP interrupt generation when the function controller is selected.

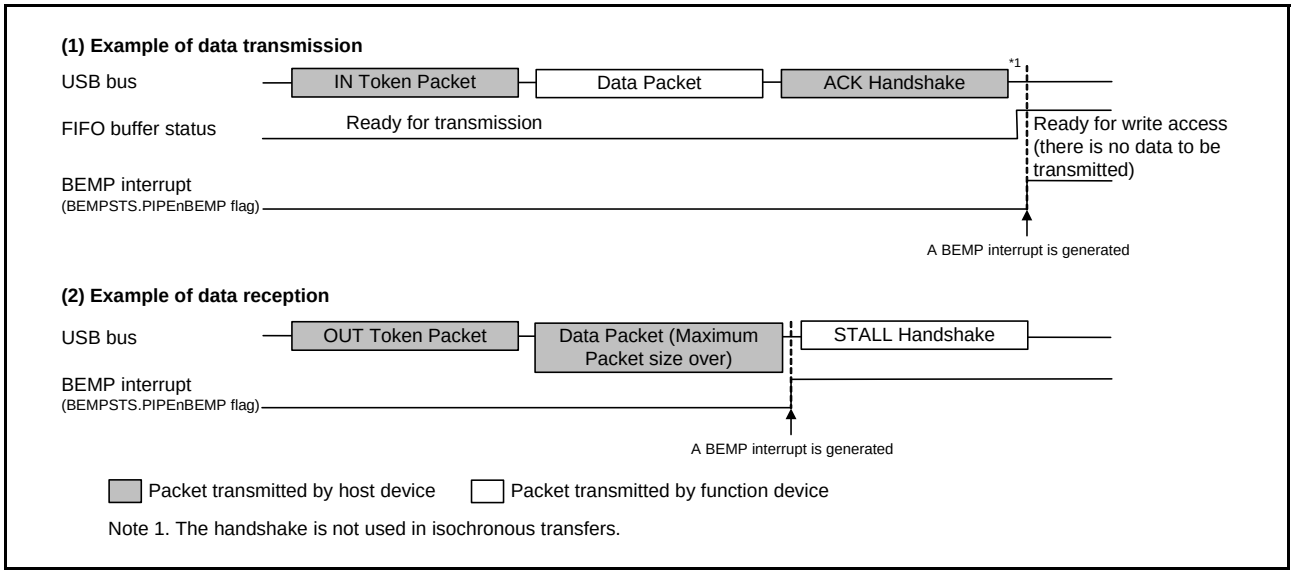


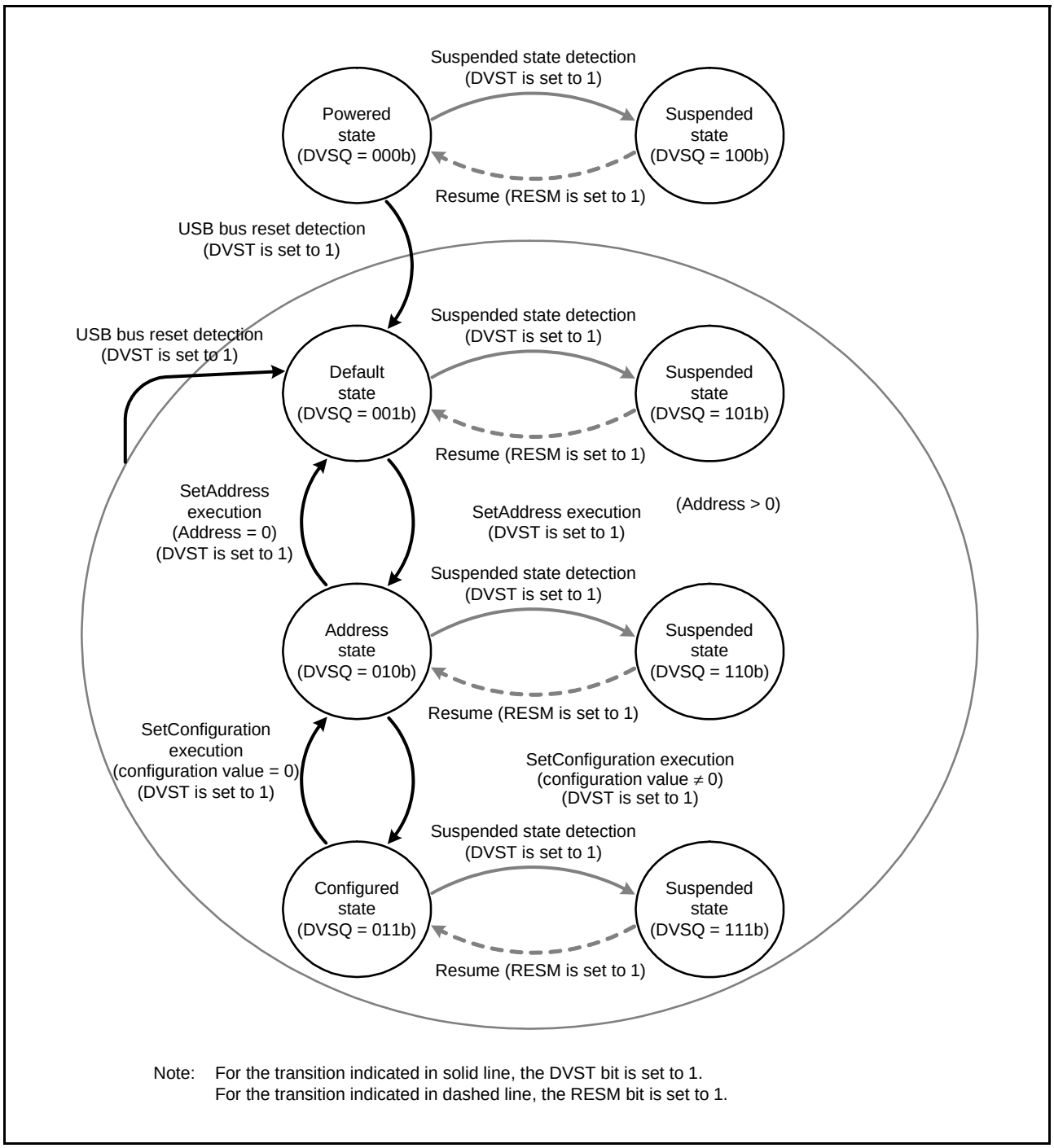
Figure 34.14 Timing of BEMP Interrupt Generation (When Function Controller is Selected)

34.3.3.4 Device State Transition Interrupt

Figure 34.15 is a diagram of device state transitions in the USB. The USB controls device state and generates device state transition interrupts. However, recovery from the suspended state (resume signal detection) is detected by means of the resume interrupt. The device state transition interrupts can be enabled or disabled individually using INTENB0. The device state to which a transition was made can be confirmed using the INTSTS0.DVSQ[2:0] flags.

When a transition is made to the default state, a device state transition interrupt is generated after a USB bus reset is detected.

Device state can be controlled only when the function controller is selected. The device state transition interrupts can also be generated only when the function controller is selected.



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 34.15 Device State Transitions

34.3.3.5 Control Transfer Stage Transition Interrupt

Figure 34.16 is a diagram of control transfer stage transitions in the USB. The USB controls the control transfer sequence and generates control transfer stage transition interrupts. The control transfer stage transition interrupts can be enabled or disabled individually using INTENB0. The transfer stage to which a transition was made can be confirmed using the INTSTS0.CTSQ[2:0] flags.

Control transfer stage transition interrupts are generated only when the function controller is selected.

The control transfer sequence errors are listed below. If an error occurs, the DCPCTR.PID[1:0] bits are set to 1xb (STALL response).

During control read transfer:

- An OUT token is received while no data has been transferred for the IN token at the data stage.
- An IN token is received at the status stage.
- A data packet with PID = DATA0 is received at the status stage.

During control write transfer:

- An IN token is received while no ACK response has been returned for the OUT token at the data stage.
- A data packet with PID = DATA0 is received for the first data packet at the data stage.
- An OUT token is received at the status stage

During no-data control transfers:

- An OUT token is received at the status stage.

At the control write transfer data stage, if the number of receive data exceeds the wLength value of the USB request, it cannot be recognized as a control transfer sequence error. At the control read transfer status stage, packets other than zero-length packets are received by an ACK response and the transfer ends normally.

When a CTRT interrupt occurs in response to a sequence error (INTSTS0.CTRT = 1), CTSQ[2:0] = 110b value is retained until the CTRT flag = 0 is written from the system (the interrupt status is cleared). Therefore, while CTSQ[2:0] = 110b is being held, the CTRT interrupt that ends the setup stage will not be generated even if a new USB request is received. (The USB retains the setup stage end, and after the interrupt status has been cleared by software, a setup stage end interrupt is generated.)

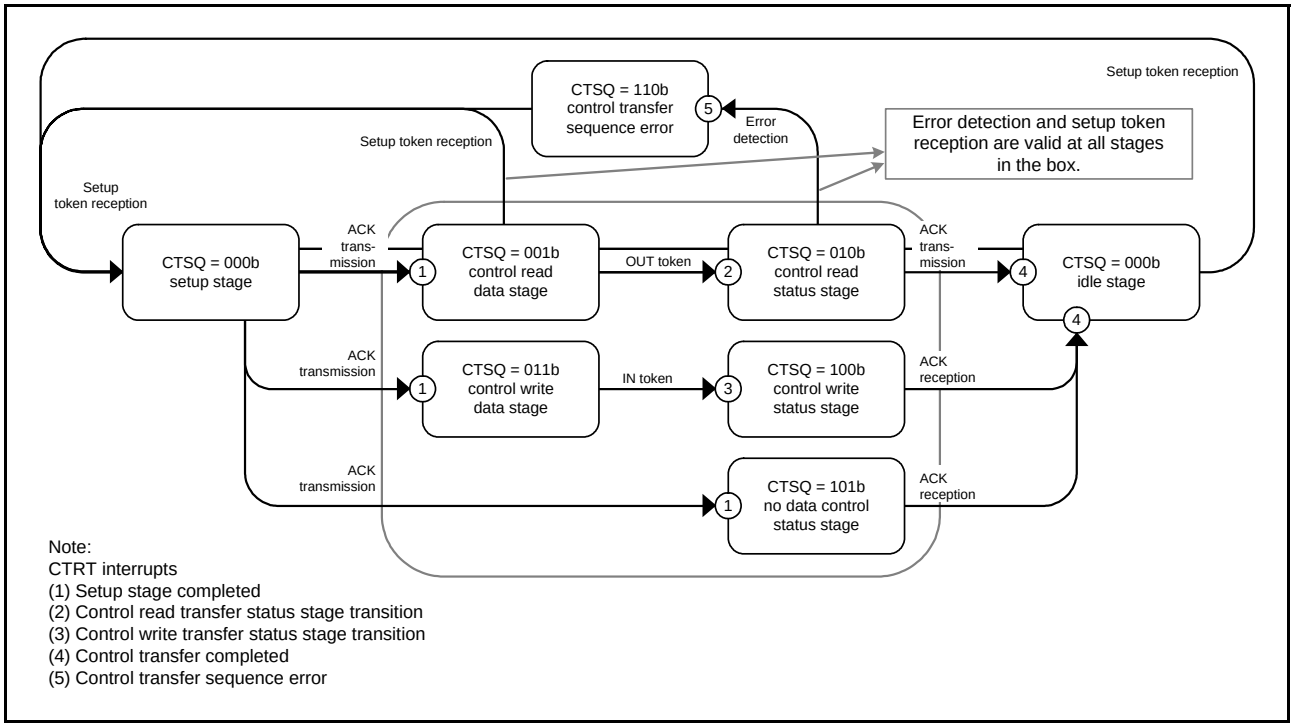


Figure 34.16 Control Transfer Stage Transitions

34.3.3.6 Frame Update Interrupt

With the host controller selected, an interrupt is generated at the timing when the frame number is updated. With the function controller selected, an SOFR interrupt is generated when the frame number is updated.
When the function controller is selected, the USB updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation.

34.3.3.7 VBUS Interrupt

When the USBm_VBUS pin level changes (m = 0, 1), a VBUS interrupt is generated. The level of the USBm_VBUS pin can be checked with the INTSTS0.VBSTS flag. Whether the host controller is connected or disconnected can be confirmed using the VBUS interrupt. However, if the system is activated with the host controller connected, the first VBUS interrupt is not generated because there is no change in the USBm_VBUS pin level.

34.3.3.8 Resume Interrupt

When the function controller is selected, a resume interrupt is generated when the device state is the suspended state and the USB bus state has changed (from J-state to K-state, or from J-state to SE0). Recovery from the suspended state is detected by means of the resume interrupt.
When the host controller is selected, no resume interrupt is generated. Use the BCHG interrupt to detect a change in the USB bus state.

34.3.3.9 OVRCR Interrupt

An OVRCR interrupt is generated when the USBm_OVRCURA or USBm_OVRCURB pin level has changed. The levels of the USBm_OVRCURA and USBm_OVRCURB pins can be checked with the SYSSTS0.OVCMON[1:0] bits. The external power supply IC can check whether overcurrent has been detected using the OVRCR interrupt.
For OTG connection, whether a change has been detected in the VBUS comparator can be checked using the OVRCR interrupt.

34.3.3.10 BCHG Interrupt

A BCHG interrupt is generated when the USB bus state has changed. The BCHG interrupt can be used to detect whether the peripheral device is connected and can also be used to detect a remote wakeup when the host controller is selected. The BCHG interrupt is generated regardless of whether the host controller or function controller is selected.

34.3.3.11 DTCH Interrupt

A DTCH interrupt is generated when disconnection of the USB bus is detected while the host controller is selected. The USB detects bus disconnection based on USB Specification 2.0.
After detecting a DTCH interrupt, the USB controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the pertinent port should be terminated by software and make a transition to the wait state for bus connection to the pertinent port (wait state for ATTCH interrupt generation).

- Modifies the DVSTCTR0.UACT bit for the port in which a DTCH interrupt has been detected to 0.
- Puts the port in which a DTCH interrupt has been generated into the idle state.

34.3.3.12 SACK Interrupt

A SACK interrupt is generated when an ACK response for the transmitted setup packet has been received from the peripheral device with the host controller selected. The SACK interrupt can be used to confirm that the setup transaction has been completed successfully.

34.3.3.13 SIGN Interrupt

A SIGN interrupt is generated when an ACK response for the transmitted setup packet has not been correctly received from the peripheral device three consecutive times with the host controller selected. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral device or corruption of an ACK packet.

34.3.3.14 ATTCH Interrupt

An ATTCH interrupt is generated when J-state or K-state of the full-speed signal level is detected on the USB port for 2.5 µs with the host controller selected. To be more specific, an ATTCH interrupt is detected on any of the following conditions.

- When K-state, SE0, or SE1 changes to J-state, and J-state continues 2.5 µs.
- When J-state, SE0, or SE1 changes to K-state, and K-state continues 2.5 µs.

34.3.3.15 EOFERR Interrupt

An EOFERR interrupt is generated when it is detected that communication is not completed at the EOF2 timing prescribed in USB Specification 2.0.
After detecting an EOFERR interrupt, the USB controls hardware as described below (irrespective of the value set in the corresponding interrupt enable bit). All pipes in which communications are currently carried out for the pertinent port should be terminated by software and perform re-enumeration of the pertinent port.

- Modifies the DVSTCTR0.UACT bit for the port in which an EOFERR interrupt has been detected to 0.
- Puts the port in which an EOFERR interrupt has been generated into the idle state.

34.3.4 Pipe Control

Table 34.16 lists the pipe settings for the USB. With USB data transfer, data transfer is carried out using the pipe that the software has associated with the endpoint. The USB has ten pipes that are used for data transfer.
Appropriate settings should be made for each of the pipes according to the specifications of the system.

Table 34.16 Pipe Settings

Register Name	Bit Name	Setting	Remarks
DCPCFG PIPECFG	TYPE[1:0]	Specifies the transfer type	Pipes 1 to 9: Can be set
	BFRE	Selects the BRDY interrupt mode	Pipes 1 to 5: Can be set
	DBLB	Selects double buffer mode	Pipes 1 to 5: Can be set
	DIR	Selects transfer direction	IN or OUT can be set
	EPNUM[3:0]	Endpoint number	Pipes 1 to 9: Can be set A value other than 0000b should be set when the pipe is used.
	SHTNAK	Selects disabled state for pipe when transfer ends	Pipes 1 and 2: Can be set (only when bulk transfer has been selected) Pipes 3 to 5: Can be set
DCPMAXP PIPEMAXP	DEVSEL[3:0]	Selects a device	Referenced only when the host controller is selected.
	MXPS[8:0]	Maximum packet size	Compliant with USB Specification 2.0.
PIPEPERI	IFIS	Buffer flush	Pipes 1 and 2: Can be set (only when isochronous transfer has been selected) Pipes 3 to 9: Cannot be set
	IITV[2:0]	Interval counter	Pipes 1 and 2: Can be set (only when isochronous transfer has been selected) Pipes 3 to 5: Cannot be set Pipes 6 to 9: Can be set (only when the host controller has been selected)
DCPCTR PIPEnCTR	BSTS	Buffer status	For the DCP, receive buffer status and transmit buffer status are switched with the ISEL bit.
	INBUFM	IN buffer monitor	Available only for pipes 1 to 5.
	SUREQ	SETUP request	Can be set only for the DCP. Can be controlled only when the host controller has been selected.
	SUREQCLR	SUREQ clear	Can be set only for the DCP. Can be controlled only when the host controller has been selected.
	ATREPM	Auto response mode	Pipes 1 to 5: Can be set Can be set only when the function controller has been selected.
	ACLRM	Auto buffer clear	Pipes 1 to 9: Can be set
	SQCLR	Sequence clear	Clears the data toggle bit.
	SQSET	Sequence set	Sets the data toggle bit.
	SQMON	Sequence monitor	Monitors the data toggle bit.
	PBUSY	Pipe busy status	
	PID[1:0]	Response PID	Refer to section 34.3.4.6, Response PID.
PIPEnTRE	TRENB	Transaction counter enable	Pipes 1 to 5: Can be set
	TRCLR	Current transaction counter clear	Pipes 1 to 5: Can be set
PIPEnTRN	—	Transaction counter	Pipes 1 to 5: Can be set

34.3.4.1 Pipe Control Register Switching Procedures

The following bits in the pipe control registers can be modified only when USB communication is prohibited (PID[1:0] = 00b (NAK)).

The following shows the registers and bits that should not be modified when USB communication is enabled (PID[1:0] = 01b (BUF)).

- Bits in the DCPCFG and DCPMAXP registers
- The SQCLR and SQSET bits in the DCPCTR register
- Bits in registers PIPECFG, PIPEMAXP, and PIPEPERI
- The ATREPM, ACLRM, SQCLR, and SQSET bits in the PIPEnCTR register
- Bits in the PIPEnTRE and PIPEnTRN registers

In order to modify the above bits in the USB communication enabled (PID[1:0] = 01b (BUF)) state, follow the procedure shown below:

1. A request to modify bits in the pipe control register occurs.
2. Modify the PID[1:0] bits corresponding to the pipe to 00b (NAK).
3. Wait until the corresponding PBUSY flag is set to 0.
4. Modify the bits in the pipe control register.

The following bits in the pipe control registers can be modified only when the pertinent pipe information has not been set by the CURPIPE[3:0] bits in registers CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

Registers that should not be set when the CURPIPE[3:0] bits are set:

- Bits in the DCPCFG and DCPMAXP register
- Bits in registers PIPECFG, PIPEMAXP and PIPEPERI

In order to modify pipe information, the CURPIPE[3:0] bits in the port select registers should be set to a pipe other than the pipe to be modified. For the DCP, the buffer should be cleared using the BCLR bit in the port control register after the pipe information is modified.

34.3.4.2 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).
- Pipes 1 and 2: These should be set to bulk transfer or isochronous transfer.
- Pipes 3 to 5: These should be set to bulk transfer.
- Pipes 6 to 9: These should be set to interrupt transfer.

34.3.4.3 Endpoint Number

The PIPECFG.EPNUM[3:0] bits are used to set the endpoint number for each pipe. The DCP is fixed at endpoint 0. The other pipes can be set from endpoint 1 to endpoint 15.

- DCP: No setting is necessary (fixed at endpoint 0).
- Pipes 1 to 9: The endpoint numbers from 1 to 15 should be selected and set.
  These should be set so that the combination of the PIPECFG.DIR bit and EPNUM[3:0] bits is unique.

34.3.4.4 Maximum Packet Size Setting

The DCPMAXP.MXPS[6:0] bits and the PIPEMAXP.MXPS[8:0] bits are used to specify the maximum packet size for each pipe. DCP and pipes 1 to 5 can be set to any of the maximum pipe sizes defined by USB Specification 2.0. For pipes 6 to 9, 64 bytes are the upper limit of the maximum packet size. The maximum packet size should be set before beginning the transfer (PID[1:0] = 01b (BUF)).

- DCP: Set 8, 16, 32, or 64.
- Pipes 1 to 5: Set 8, 16, 32, or 64 when using bulk transfer.
- Pipes 1 and 2: Set a value between 1 and 256 when using isochronous transfer.
- Pipes 6 to 9: Set a value between 1 and 64.

34.3.4.5 Transaction Counter (For Pipes 1 to 5 in Reading Direction)

When the specified number of transactions has been completed in the data packet receiving direction, the USB recognizes that the transfer has ended. Two transaction counters are provided: one is the PIPEnTRN register that specifies the number of transactions to be executed and the other is the current counter that internally counts the number of executed transactions. With the PIPECFG.SHTNAK bit set to 1, when the current counter value matches the specified number of transactions, the corresponding PIPEnCTR.PID[1:0] bits are set to 00b (NAK) and the subsequent transfer is disabled. The transactions can be counted again from the beginning by initializing the current counter of the transaction counter function through the PIPEnTRE.TRCLR bit. The information read from PIPEnTRN differs depending on the setting of the PIPEnTRE.TRENB bit.

- The TRENB bit = 0: The specified transaction counter value can be read.
- The TRENB bit = 1: The current counter value indicating the internally counted number of executed transactions can be read.

When operating the TRCLR bit, the following should be noted.

- If the transactions are being counted and PID[1:0] = 01b (BUF), the current counter cannot be cleared.
- If there is any data left in the buffer, the current counter cannot be cleared.

34.3.4.6 Response PID

The PID[1:0] bits in the DCPCTR and PIPEnCTR registers are used to set the response PID for each pipe.
The following shows the USB operation with various response PID settings:

(1) Response PID settings when the host controller is selected:

The response PID is used to specify the execution of transactions.

- NAK setting: Using pipes is disabled. No transaction is executed.
- BUF setting: Transactions are executed based on the status of the buffer memory.
  For OUT direction: If there are transmit data in the buffer memory, an OUT token is issued.
  For IN direction: If there is an area to receive data in the buffer memory, an IN token is issued.
- STALL setting: Using pipes is disabled. No transaction is executed.

Note: Setup transactions for the DCP are set with the DCPCTR.SUREQ bit.

(2) Response PID settings when the function controller is selected:

The response PID is used to specify the response to transactions from the host.

- NAK setting: The NAK response is returned in response to the generated transaction.
- BUF setting: Responses are made to transactions according to the status of the buffer memory.
- STALL setting: The STALL response is returned in response to the generated transaction.

Note: For setup transactions, an ACK response is returned regardless of the PID[1:0] bits setting, and the USB request is stored in the register.

The USB may write to the PID[1:0] bits, depending on the results of the transaction as described below.

(3) When the host controller has been selected and the response PID is set by hardware:

- NAK setting: In the following cases, PID[1:0] = 00b (NAK) is set and issuing of tokens is automatically stopped:
  When a transfer other than isochronous transfer has been performed and an NRDY interrupt is generated.
  (For details, refer to section 34.3.3.2, NRDY Interrupt.)
  - If a short packet is received when the PIPECFG.SHTNAK bit has been set to 1 for bulk transfer.
  - If the transaction counting ends when the SHTNAK bit has been set to 1 for bulk transfer.
- BUF setting: There is no BUF writing by the USB.
- STALL setting: In the following cases, PID[1:0] = 1xb (STALL) is set and issuing of tokens is automatically stopped:
  When STALL is received in response to the transmitted token.
  When the size of the receive data packet exceeds the maximum packet size.

(4) When the function controller has been selected and the response PID is set by hardware:

- NAK setting: In the following cases, PID[1:0] = 00b (NAK) is set and NAK is returned in response to transactions:
  When the SETUP token is received normally (DCP only).
  If the transaction counting ends or a short packet is received when the PIPECFG.SHTNAK bit has been set to 1 for bulk transfer.
- BUF setting: There is no BUF writing by the USB.
- STALL setting: In the following cases, PID[1:0] = 1xb (STALL) is set and STALL is returned in response to transactions:
  When a maximum packet size exceeded error is detected in the received data packet.
  When a control transfer sequence error has been detected (DCP only).

34.3.4.7 Data PID Sequence Bit

The USB automatically toggles the sequence bit in the data PID when data is transferred successfully in the control transfer data stage, bulk transfer, and interrupt transfer. The sequence bit of the next data PID to be transmitted can be confirmed with the SQMON flag in the DCPCTR and PIPEnCTR registers. When data is transmitted, the sequence bit switches at the timing of ACK handshake reception. When data is received, the sequence bit switches at the timing of ACK handshake transmission. The DCPCTR.SQCLR bit and the PIPEnCTR.SQSET bit can be used to change the data PID sequence bit.

When the function controller has been selected and control transfer is used, the USB automatically sets the sequence bit when a stage transition is made. DATA1 is returned when the setup stage is ended. The sequence bit is not referenced and PID = DATA1 is returned in a status stage. Therefore, software settings are not required. However, when the host controller has been selected and control transfer is used, the sequence bit should be set by software at a stage transition. For the ClearFeature request transmission or reception, the data PID sequence bit should be set by software regardless of whether the host controller or function controller is selected.

34.3.4.8 Response PID = NAK Function

The USB has a function that disables pipe operation (response PID = NAK) at the timing at which the final data packet of a transaction is received (the USB automatically distinguishes this based on reception of a short packet or the transaction counter) by setting the PIPECFG.SHTNAK bit to 1.

When the double buffer mode is being used for the buffer memory, using this function enables reception of data packets in transfer units. If pipe operation has been disabled, software should set the pipe to the enabled state again (response PID = BUF).

The response PID = NAK function can be used only when bulk transfers are used.

34.3.4.9 Auto Response Mode

With the pipes for bulk transfer (pipe 1 to pipe 5), when the PIPEnCTR.ATREPM bit is set to 1, a transition is made to auto response mode. During an OUT transfer (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.

34.3.4.10 OUT-NAK Mode

With the pipes for bulk OUT transfer, NAK is returned in response to an OUT token and an NRDY interrupt is output when the PIPEnCTR.ATREPM bit is set to 1. To make a transition from normal mode to OUT-NAK mode, OUT-NAK mode should be specified in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, OUT-NAK mode becomes valid. However, if an OUT token is received immediately before pipe operation is disabled, the token data is normally received, and an ACK is returned to the host.

To make a transition from OUT-NAK mode to normal mode, OUT-NAK mode should be canceled in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). In normal mode, reception of OUT data is enabled.

34.3.4.11 Null Auto Response Mode

With the pipes for bulk IN transfer, zero-length packets are continuously transmitted when the PIPEnCTR.ATREPM bit is set to 1.

To make a transition from normal mode to null auto response mode, null auto response mode should be set in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, null auto response mode becomes valid. Before setting null auto response mode, the PIPEnCTR.INBUFM = 0 should be confirmed because the mode can be set only when the buffer is empty. If the INBUFM flag is 1, the buffer should be emptied with the PIPEnCTR.ACLRM bit. While a transition to null auto response mode is being made, data should not be written from the FIFO port.

To make a transition from null auto response mode to normal mode, pipe operation disabled state (response PID = NAK) should be retained for the period of zero-length packet transmission (about 10 µs) before canceling null auto response mode. In normal mode, data can be written from the FIFO port; therefore, packet transmission to the host is enabled by enabling pipe operation (response PID = BUF).

34.3.5 FIFO Buffer Memory

34.3.5.1 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 34.17 and Table 34.18 show the buffer status in the USB. The buffer memory status can be confirmed using the DCPCTR.BSTS flag and the PIPEnCTR.INBUFM flag. The transfer direction for the buffer memory can be specified using either the PIPECFG.DIR bit or the CFIFOSEL.ISEL bit (when DCP is selected).

The INBUFM flag is valid for pipe 0 to pipe 5 in the transmitting direction.

When a transmitting pipe uses the double buffer configuration, software can read the BSTS flag to monitor the buffer memory status on the CPU side and the INBUFM flag to monitor the buffer memory status on the SIE side. When the BEMP interrupt may not show the buffer empty status because the write access to the FIFO port by the CPU or DMAC/DTC is slow, software can use the INBUFM flag to confirm the end of transmission.

Table 34.17 Buffer Status Indicated by the BSTS Flag

ISEL or DIR	BSTS	Buffer Memory Status
0 (receiving direction)	0	There is no received data, or data is being received. Reading from the FIFO port is disabled.
0 (receiving direction)	1	There is received data, or a zero-length packet has been received. Reading from the FIFO port is allowed. Note that when a zero-length packet is received, reading is not possible and the buffer must be cleared.
1 (transmitting direction)	0	The transmission has not been completed. Writing to the FIFO port is disabled.
1 (transmitting direction)	1	The transmission has been completed. CPU write is allowed.

Table 34.18 Buffer Status Indicated by the INBUFM Flag

DIR	INBUFM	Buffer Memory Status
0 (receiving direction)	Invalid	Invalid
1 (transmitting direction)	0	The transmission has been completed. There is no waiting data to be transmitted.
1 (transmitting direction)	1	The FIFO port has written data to the buffer. There is data to be transmitted.

34.3.5.2 FIFO Buffer Clearing

Table 34.19 shows the clearing of the FIFO buffer memory by the USB. The buffer memory can be cleared using the BCLR, DnFIFOSEL.DCLRM, and PIPEnCTR.ACLRM bit in the port control register.
Either a single or double buffer configuration can be selected for pipes 1 to 5, using the PIPECFG.DBLB bit.

Table 34.19 List of Buffer Clearing Methods

FIFO Buffer Clearing Mode	Clearing Buffer Memory on CPU Side	Mode for Automatically Clearing Buffer Memory after Reading Specified Pipe Data	Auto Buffer Clear Mode for Discarding All Received Packets
Register used	CFIFOCTR DnFIFOCTR	DnFIFOSEL	PIPEnCTR
Bit used	BCLR	DCLRM	ACLRM
Clearing condition	Cleared by writing 1	1: Mode valid 0: Mode invalid	1: Mode valid 0: Mode invalid

(1) Auto Buffer Clear Mode Function

With the USB, all received data packets are discarded if the PIPEnCTR.ACLRM bit is set to 1. If a correct data packet has been received, the ACK response is returned to the host controller. The auto buffer clear mode function can be set only in the buffer memory reading direction.
If the ACLRM bit is set to 1 and then to 0, the buffer memory of the selected pipe can be cleared regardless of the access direction.
An access cycle of at least 100 ns is required for the internal hardware sequence processing time between ACLRM = 1 and ACLRM = 0.

34.3.5.3 FIFO Port Functions

Table 34.20 shows the settings for the FIFO port functions of the USB. In write access, writing data until the maximum packet size is reached automatically enables transmission of the data. To enable transmission before the maximum packet size is reached, the BVAL bit in the port control register should be set to end writing. To send a zero-length packet, the BCLR bit in the register should be used to clear the buffer and then the BVAL bit set in order to end writing.
In reading, reception of new packets is automatically enabled when all data has been read. Data cannot be read when a zero-length packet has been received (the DTLN[8:0] flags = 0), so the BCLR bit in the register should be used to clear the buffer. The length of the receive data can be confirmed using the DTLN[8:0] flags in the port control register.

Table 34.20 FIFO Port Function Settings

Register Name	Bit Name	Description
CFIFOSEL, DnFIFOSEL (n = 0, 1)	RCNT	Selects DTLN read mode.
	REW	Buffer memory rewind (re-read, rewrite).
	DCLRM	Automatically clears receive data for a specified pipe after the data has been read (only for DnFIFO).
	DREQE	Enables DMA/DTC transfers (only for DnFIFO).
	MBW	FIFO port access bit width.
	BIGEND	Selects FIFO port endian.
	ISEL	FIFO port access direction (only for DCP).
	CURPIPE	Selects the current pipe.
CFIFOCTR, DnFIFOCTR (n = 0, 1)	BVAL	Ends writing to the buffer memory.
	BCLR	Clears the buffer memory on the CPU side.
	DTLN	Checks the length of receive data.

(1) FIFO Port Selection

Table 34.21 shows the pipes that can be selected with the various FIFO ports. The pipe to be accessed should be selected using the CURPIPE[3:0] bits in the port select register. After the pipe is selected, whether the written value can be correctly read from the CURPIPE[3:0] bits should be checked. (If the previous pipe number is read, it indicates that the pipe modification is being executed by the USB controller.) Then, the FRDY flag in a port control register = 1 is checked.
In addition, the bus width to be accessed should be selected using the MBW bit in the port select register. The buffer memory access direction conforms to the PIPECFG.DIR bit. Only for the DCP, the ISEL bit in the port select register determines the direction.

Table 34.21 FIFO Port Access Categorized by Pipe

Pipe	Access Method	Port that can be Used
DCP	CPU access	CFIFO port register
Pipe 1 to pipe 9	CPU access	CFIFO port register D0FIFO/D1FIFO port register
	DMAC/DTC access	D0FIFO/D1FIFO port register

(2) REW Bit

It is possible to temporarily stop access to the pipe currently being accessed, access a different pipe, and then continue processing for the current pipe again. The REW bit in the port select register is used for this processing.
If a pipe is selected through the CURPIPE[3:0] bits in the port select register with the REW bit set to 1, the pointer used for reading from and writing to the buffer memory is reset, and reading or writing can be carried out from the first byte.
If a pipe is selected with 0 set for the REW bit, data can be read and written in continuation from the previous selection, without the pointer being reset.
To access the FIFO port, the FRDY flag in the port control register = 1 should be checked after selecting a pipe.

34.3.5.4 DMA Transfers (D0FIFO and D1FIFO Ports)

(1) Overview of DMA Transfers

For pipes 1 to 9, 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 34.22 shows the packet reception and buffer memory clearing processing by software for each of the various settings. As shown in Table 34.22, the buffer clearing conditions depend on the value set in the PIPECFG.BFRE bit. Using the DnFIFOSEL.DCLRM bit eliminates the need for the buffer to be cleared by software in any situation that requires buffer clearing. This enables DMA transfers without involving software.
The DnFIFO auto clear mode can be set only in the buffer memory reading direction.

Table 34.22 Packet Reception and Buffer Memory Clearing Processing by Software

Buffer Status When Packet is Received	Register Setting			
	DCLRM = 0		DCLRM = 1	
	BFRE = 0	BFRE = 1	BFRE = 0	BFRE = 1
Buffer full	Clearing is not necessary	Clearing is not necessary	Clearing is not necessary	Clearing is not necessary
Zero-length packet reception	Clearing is necessary	Clearing is necessary	Clearing is not necessary	Clearing is not necessary
Normal short packet reception	Clearing is not necessary	Clearing is necessary	Clearing is not necessary	Clearing is not necessary
Transaction count end	Clearing is not necessary	Clearing is necessary	Clearing is not necessary	Clearing is not necessary

34.3.6 Control Transfers Using DCP

In the data stage of control transfers, data is transferred using the default control pipe (DCP).
The DCP buffer memory is a 64-byte single buffer and is a fixed area that is shared for both control reading and control writing. The buffer memory can be accessed only through the CFIFO port.

34.3.6.1 Control Transfers When the Host Controller is Selected

(1) Setup Stage

Registers USBREQ, USBVAL, USBINDX, and USBLENG are the registers that are used to transmit a USB request for setup transactions. Writing setup packet data to the registers and writing 1 to the DCPCTR.SUREQ bit transmits the specified data for setup transactions. Upon completion of the transaction, the SUREQ bit is set to 0. The above USB request registers should not be modified while SUREQ = 1.
After the attached state of the connected function device is detected, the first setup transaction for the device should be issued by using the sequence described above with the DCPMAXP.DEVSEL[3:0] bits set to 0 and the DEVADD0.USBSPD[1:0] bits set appropriately.
After the connected function device is shifted to the Address state, setup transactions should be issued by using the sequence described above with the assigned USB address set in the DEVSEL[3:0] bits and the bits in the DEVADDn register corresponding to the specified USB address set appropriately. For example, when PIPEMAXP.DEVSEL[3:0] = 0010b, make appropriate settings in the DEVADD2 register; when PIPEMAXP.DEVSEL[3:0] = 0101b, make appropriate settings in the DEVADD5 register.
When the setup transaction data has been sent, an interrupt request is generated according to the response received from the peripheral device (SIGN or SACK flag in the INTSTS1 register), by means of which the result of the setup transactions can be confirmed.
A data packet of DATA0 (USB request) is transmitted as the data packet for a setup transaction regardless of the setting of the DCPCTR.SQMON flag.

(2) Data Stage

Data is transferred using the DCP buffer memory.
The access direction of the DCP buffer memory should be specified using the CFIFOSEL.ISEL bit. The transfer direction should be specified using the DCPCFG.DIR bit.
For the first data packet of the data stage, the data PID should be transferred as DATA1. Set data PID = DATA1 in the DCPCTR.SQSET bit and the PID[1:0] bits = 01b (BUF). Completion of data transfer is detected using the BRDY or BEMP interrupt.
For control write transfers, when the number of data bytes to be sent is an integer multiple of the maximum packet size, software should control so as to send a zero-length packet at the end.

(3) Status Stage

Zero-length packet data is transferred in the direction opposite to that in the data stage. As in the data stage, data is transferred using the DCP buffer memory. Transactions are done in the same manner as the data stage.
For the data packets of the status stage, the data PID should be set to DATA1 using the DCPCTR.SQSET bit.
For reception of a zero-length packet, the received data length should be confirmed using the CFIFOCTR.DTLN[8:0] flags after a BRDY interrupt is generated, and the buffer memory should then be cleared using the CFIFOCTR.BCLR bit.

34.3.6.2 Control Transfers When the Function Controller is Selected

(1) Setup Stage

The USB sends an ACK response for a correct setup packet targeted to the USB. The operation of the USB in the setup stage is described below.
When receiving a new setup packet, the USB sets the following bits.

- Set the INTSTS0.VALID flag to 1.
- Set the DCPCTR.PID[1:0] bits to 00b (NAK).
- Set the DCPCTR.CCPL bit to 0.

When receiving a data packet right after the setup packet, the USB stores the USB request parameters in registers USBREQ, USBVAL, USBINDX, and USBLENG.

Response processing with respect to the control transfer should be carried out after setting the VALID flag = 0. In the VALID flag = 1 state, PID[1:0] = 01b (BUF) cannot be set, and the data stage cannot be terminated.
Using the function of the VALID flag, the USB can suspend the current request processing when receiving a new USB request during a control transfer, and can send a response to the newest request.
In addition, the USB automatically detects the direction bit (bit 8 of bmRequestType) and the request data length (wLength) of the received USB request, distinguishes between control read transfer, control write transfer, and no-data control transfer, and controls stage transitions. For a wrong sequence, the sequence error of the control transfer stage transition interrupt is generated, and the software is notified of occurrence of the error. For the stage control of the USB, refer to Figure 34.16.

(2) Data Stage

Data transfers corresponding to received USB requests should be done using the DCP. Before accessing the DCP buffer memory, the access direction should be specified using the CFIFOSEL.ISEL bit.
If the transfer data is larger than the size of the DCP buffer memory, the data transfer should be carried out using the BRDY interrupt for control write transfers and the BEMP interrupt for control read transfers.

(3) Status Stage

Control transfers are terminated by setting the DCPCTR.CCPL bit to 1 while the DCPCTR.PID[1:0] bits are set to 01b (BUF).
After the above settings have been made, the USB automatically executes the status stage in accordance with the data transfer direction determined at the setup stage. The specific procedure is as follows.

- For control read transfers
  A zero-length packet is received from the USB host and an ACK response is sent.
- For control write transfers and no-data control transfers
  A zero-length packet is transmitted and an ACK response is received from the USB host.

(4) Control Transfer Auto Response Function

The USB automatically responds to a correct SET_ADDRESS request. If any of the following errors occurs in the SET_ADDRESS request, a response from the software is necessary.

- bmRequestType is not 00h: Any transfer other than a control write transfer
- wIndex is not 00h: Request error
- wIndex is not 00h: Any transfer other than a no-data control transfer
- wValue is larger than 7Fh: Request error
- INTSTS0.DVSQ[2:0] are 011b (Configured state): Control transfer of a device state error

For all requests other than the SET_ADDRESS request, a response is required from the corresponding software.

34.3.7 Bulk Transfers (Pipes 1 to 5)

The buffer memory usage (single/double buffer setting) can be selected for bulk transfers. The USB provides the following functions for bulk transfers.

- BRDY interrupt function (PIPECFG.BFRE bit: refer to section 34.3.3.1, (2) When the SOFCFG.BRDYM Bit = 0 and the PIPECFG.BFRE Bit = 1
- Transaction count function
  (PIPEnTRE.TRENB and TRCLR bits and PIPEnTRN register: refer to section 34.3.4.5, Transaction Counter (For Pipes 1 to 5 in Reading Direction)
- Response PID = NAK function (PIPECFG.SHTNAK bit: refer to section 34.3.4.8, Response PID = NAK Function
- Auto response mode (PIPEnCTR.ATREPM bit: refer to section 34.3.4.9, Auto Response Mode)

34.3.8 Interrupt Transfers (Pipes 6 to 9)

When the function controller is selected, the USB carries out interrupt transfers in accordance with the timing controlled by the host controller.
When the host controller is selected, the timing of issuing a token can be specified using the interval counter.

34.3.8.1 Interval Counter during Interrupt Transfers When the Host Controller is Selected

For interrupt transfers, intervals between transactions are set in the PIPEPERI.IITV[2:0] bits. The USB controller issues interrupt transfer tokens based on the specified intervals.

(1) Counter Initialization

The interval counter is initialized when the MCU is reset or when the PIPEnCTR.ACLRM bit is set to 1. Note that the PIPEPERI.IITV[2:0] bits are not initialized when the ACLRM bit is used for initialization.

Note that the interval counter is not initialized in the following case.

- USB bus reset or USB suspended
  The IITV[2:0] bits are not initialized. Setting 1 to the DVSTCTR0.UACT bit starts counting from the value before entering the USB bus reset state or USB suspended state.

(2) Operation When Transmission/Reception is Impossible at Token Issuance Timing

The USB cannot issue tokens even at token issuance timing in the following cases. In such a case, the USB attempts transactions at the subsequent interval.

- When the PID[1:0] bits are set to 00b (NAK) or 1xb (STALL).
- When the buffer memory is full at the token sending timing in the receiving (IN) direction.
- When there is no data to be sent in the buffer memory at the token sending timing in the transmitting (OUT) direction.

34.3.9 Isochronous Transfers (Pipes 1 and 2)

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)

34.3.9.1 Error Detection in Isochronous Transfers

The USB has a function for detecting the error information described below, so that when errors occur in isochronous transfers, they can be controlled by software. Table 34.23 and Table 34.24 show the priority in which errors are confirmed and the interrupts generated corresponding to errors.

(a) PID errors

- If the PID of the received packet is illegal.

(b) CRC errors and bit stuffing errors

- If an error occurs in the CRC of the received packet or the bit stuffing is illegal.

(c) Maximum packet size exceeded

- The data of the received packet is larger than the specified maximum packet size.

(d) Overrun and underrun errors

- When the host controller is selected
  When the buffer memory is full at the token sending timing in the IN (receiving) direction.
  When there is no data to be sent in the buffer memory at the token sending timing in the OUT (transmitting) direction.
- When the function controller is selected
  When there is no data to be sent in the buffer memory at the token receiving timing in the IN (transmitting) direction.
  When the buffer memory is full at the token receiving timing in the OUT (receiving) direction.

(e) Interval errors

An interval error is generated on any of the following conditions when the function controller is selected.

- During an isochronous IN transfer, an IN token could not be received in the interval frame.
- During an isochronous OUT transfer, an OUT token could not be received in the interval frame.

Table 34.23 Error Detection When a Token is Received

Detection Priority	Error	Generated Interrupt and Status
1	PID errors	No interrupts are generated in both cases when the host controller is selected and the function controller is selected (ignored as a corrupted packet).
2	CRC errors and bit stuffing errors	No interrupts generated in both cases when the host controller is selected and the function controller is selected (ignored as a corrupted packet).
3	Overrun and underrun errors	An NRDY interrupt is generated to set the FRMNUM.OVRN flag to 1 in both cases when the host controller is selected and function controller is selected. When the function controller is selected, a zero-length packet is transmitted in response to IN token. However, no data packets are received in response to OUT token.
4	Interval errors	An NRDY interrupt is generated when the function controller is selected. It is not generated when the host controller is selected.

Table 34.24 Error Detection When a Data Packet is Received

Detection Priority	Error	Generated Interrupt and Status
1	PID errors	No interrupts are generated (ignored as a corrupted packet).
2	CRC errors and bit stuffing errors	An NRDY interrupt is generated to set the FRMNUM.CRCE to 1 bit in both cases when the host controller is selected and the function controller is selected.
3	Maximum packet size exceeded errors	A BEMP interrupt is generated to set the PID[1:0] bits to 1xb (STALL) in both cases when the host controller is selected and the function controller is selected.

34.3.9.2 Data PID

When the function controller is selected, the USB operates as follows in response to the received PID.
IN direction

- DATA0: Sent as data packet PID
- DATA1: Not sent
- DATA2: Not sent
- mDATA: Not sent

OUT direction

- DATA0: Received normally as data packet PID
- DATA1: Received normally as data packet PID
- DATA2: Packets are ignored
- mDATA: Packets are ignored

34.3.9.3 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 34.25 when the function controller is selected. When the host controller is selected, the USB generates the token issuance timing. When the host controller is selected, the interval counter operation is the same as that in the interrupt transfer.

Table 34.25 Interval Counter Function When the Function Controller is Selected

Transfer Direction	Function	Conditions for Detection
IN	Flushes transmit buffer	When an IN token cannot be successfully received in the interval frame during an isochronous IN transfer
OUT	Notifies that a token not being received	When an OUT token cannot be successfully received in the interval frame during an isochronous OUT transfer

The interval count is carried out when an SOF is received or for interpolated SOFs, so the isochronism can be maintained even if an SOF is damaged. The frame interval that can be set is the $2^{IITV[2:0]}$ frames.

(1) Counter Initialization When the Function Controller is Selected

The interval counter is initialized when the MCU is reset or when the PIPEnCTR.ACLRM bit is set to 1. Note that the PIPEPERI.IITV[2:0] bits are not initialized when the ACLRM bit is used for initialization.

After the interval counter has been initialized, counting is started under either of the following conditions 1 and 2 when a packet has been transferred successfully.

1. An SOF is received after transmission of data in response to an IN token while the PID[1:0] bits are 01b (BUF).
2. An SOF is received after reception of data of an OUT token while the PID[1:0] bits are 01b (BUF).

Note that the interval counter is not initialized under the following conditions.

- When the PID[1:0] bits are set to 00b (NAK) or 1xb (STALL)
  The interval timer does not stop. The USB attempts transactions at the subsequent interval.
- When the USB bus is reset or USB is suspended
  The IITV[2:0] bits are not initialized. When an SOF has been received, counting is restarted from the value prior to the reception of the SOF.

(2) Interval Counting and Transfer Control When the Host Controller is Selected

The USB controls the interval between token issuance operations based on the PIPEPERI.IITV[2:0] bit settings. Specifically, the USB issues a token for a selected pipe once every $2^{IITV[2:0]}$ frames.
The USB starts counting the token issuance interval at the frame following the frame in which the PID[1:0] bits have been set to 01b (BUF) by software.

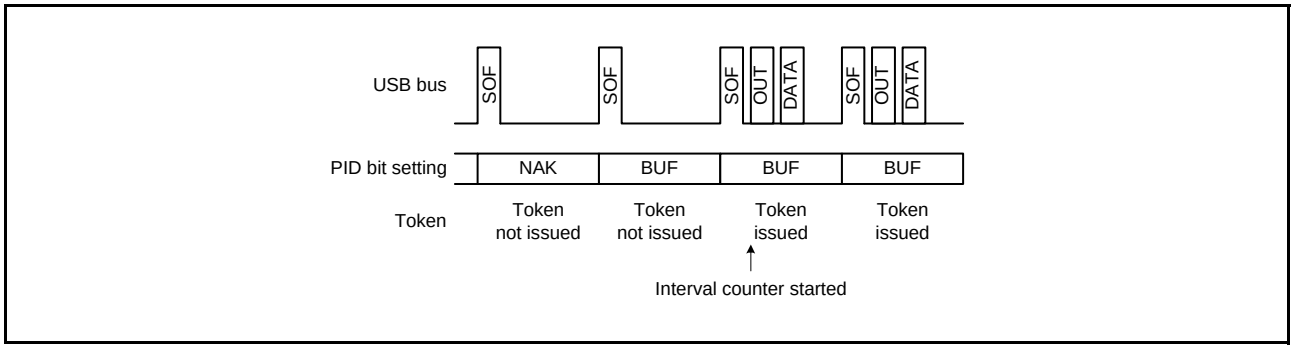


Figure 34.17 Token Issuance When IITV[2:0] = 000b

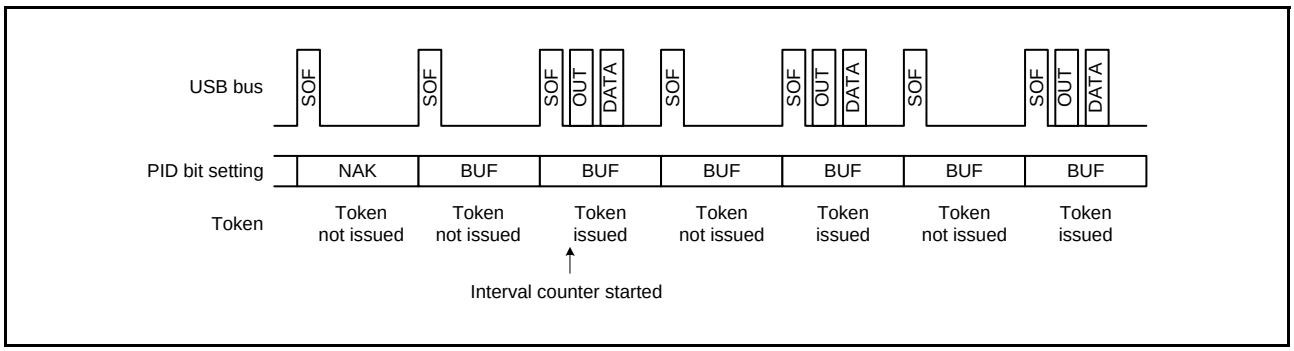


Figure 34.18 Token Issuance When IITV[2:0] = 001b

When the selected pipe is set for isochronous transfers, the USB carries out the following operation in addition to controlling the token issuance interval. The USB issues a token even when the NRDY interrupt generation condition is satisfied.

(a) When the selected pipe is for isochronous IN transfers

The USB generates an NRDY interrupt when the USB issues an IN token but does not receive a packet successfully from a peripheral device (no response or packet error).
The USB sets the FRMNUM.OVRN flag to 1 generating an NRDY interrupt when the time to issue an IN token comes while the USB cannot receive data because the FIFO buffer is full (due to the fact that the CPU or DMAC/DTC is too slow to read data from the FIFO buffer).

(b) When the selected pipe is for isochronous OUT transfers

The USB sets the OVRN flag to 1 generating an NRDY interrupt and transmitting a zero-length packet when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer (because the CPU or DMAC/DTC is too slow to write data to the FIFO buffer).
The token issuance interval is reset on any of the following conditions.

- When the USB is reset through a reset pin
  (The IITV[2:0] bits are also set to 0).
- When the PIPEnCTR.ACLRM bit has been set to 1 by software.

(3) Interval Counting and Transfer Control When the Function Controller is Selected

(a) When the selected pipe is for isochronous OUT transfers

The USB generates an NRDY interrupt when the USB fails to receive a data packet within the interval set by the PIPEPERI.IITV[2:0] bits.
The USB also generates an NRDY interrupt when the USB fails to receive data because of a CRC error or other errors contained in the data packet or because of the FIFO buffer being full.
The NRDY interrupt is generated at the timing of SOF packet reception. Even if the SOF packet is corrupted, the internal interpolation allows the interrupt to be generated at the timing to receive the SOF packet.
However, when the IITV[2:0] bits are set to a value other than 0, the USB generates an NRDY interrupt on receiving an SOF packet for every interval after starting interval counting operation.
When the PID[1:0] bits are set to 00b (NAK) by software after starting the interval timer, the USB does not generate an NRDY interrupt on receiving an SOF packet.
The timing to start interval counting depends on the setting of IITV[2:0] bits as shown below.

- When the IITV[2:0] = 000b: The interval counting starts when software has set the PID[1:0] bits for the selected pipe to 01b (BUF).

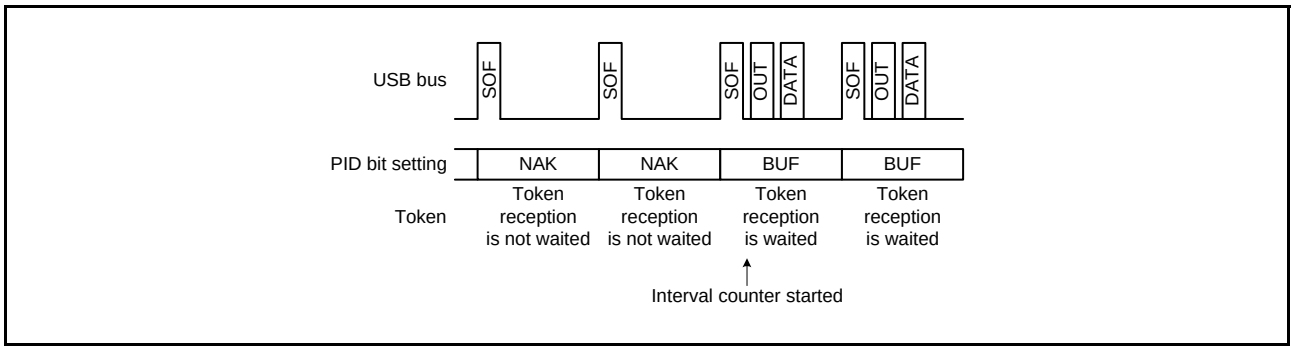


Figure 34.19 Relationship between Frames and Expected Token Reception When IITV[2:0] = 000b

- When the IITV[2:0] ≠ 000b: The interval counting starts on completion of successful reception of the first data packet after the PID[1:0] bits for the selected pipe have been modified to 01b (BUF).

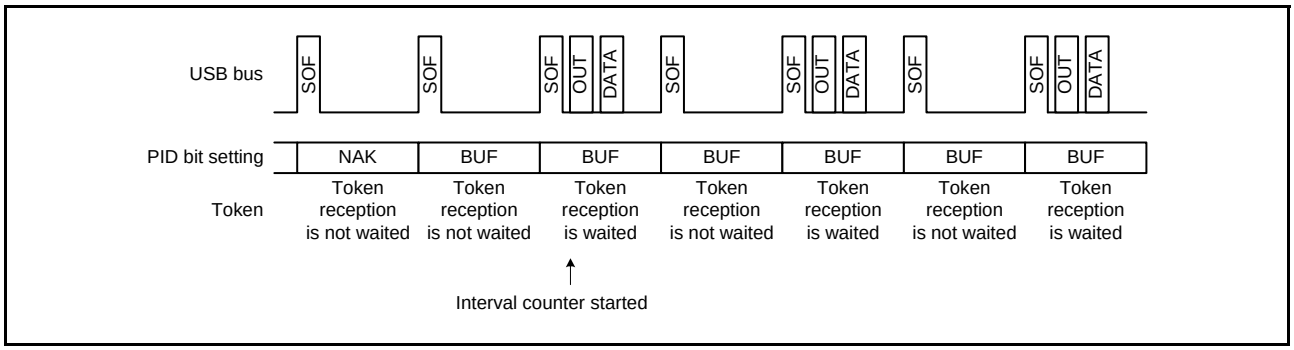


Figure 34.20 Relationship between Frames and Expected Token Reception When IITV[2:0] ≠ 000b

(b) When the selected pipe is for isochronous IN transfers

The PIPEPERI.IFIS bit should be 1 for this use. When IFIS = 0, the USB transmits a data packet in response to the received IN token irrespective of the setting of the PIPEPERI.IITV[2:0] bits.
When IFIS = 1, the USB clears the FIFO buffer when the USB fails to receive an IN token in the frame at the interval set by the IITV[2:0] bits while there is data to be transmitted in the FIFO buffer.
The USB also clears the FIFO buffer when the USB fails to receive an IN token successfully because of a bus error such as a CRC error contained in the IN token.
The FIFO buffer is cleared at the timing of SOF packet reception. Even if the SOF packet is corrupted, the internal interpolation allows the FIFO buffer to be cleared at the timing to receive the SOF packet.

The timing to start interval counting depends on the setting of the IITV[2:0] bits (similar to the timing during OUT transfers).
The interval is counted on any of the following conditions in function controller mode.

- When a hardware-reset is applied to the USB (here, the IITV[2:0] bits are also set to 000b).
- When the PIPEnCTR.ACLRM bit is set to 1 by software.
- When the USB detects a USB bus reset.

(4) Setup of Data to be Transmitted Using Isochronous Transfer When the Function Controller is Selected

With isochronous data transmission using the USB in the function controller, after data has been written to the buffer memory, a data packet can be transmitted with the next frame after the frame in which an SOF packet is detected. This function is called the isochronous transfer transmission data setup function, and it makes it possible to designate the frame from which transmission began.

In a double buffer configuration, even after the writing of data to both buffers has been completed, transmission will be enabled for only one buffer to which data writing was completed first. Accordingly, even if multiple IN tokens are received, only one packet of data is transmitted from a single buffer.

When an IN token is received, if the buffer memory is in the transmission enabled state, the USB transmits data as a normal response. If the buffer memory is not in the transmission enabled state, however, a zero-length packet is sent and an underrun error occurs.

Figure 34.21 shows an example of transmission using the isochronous transfer transmission data setup function with the USB when IITV[2:0] = 000b (every frame) has been set.

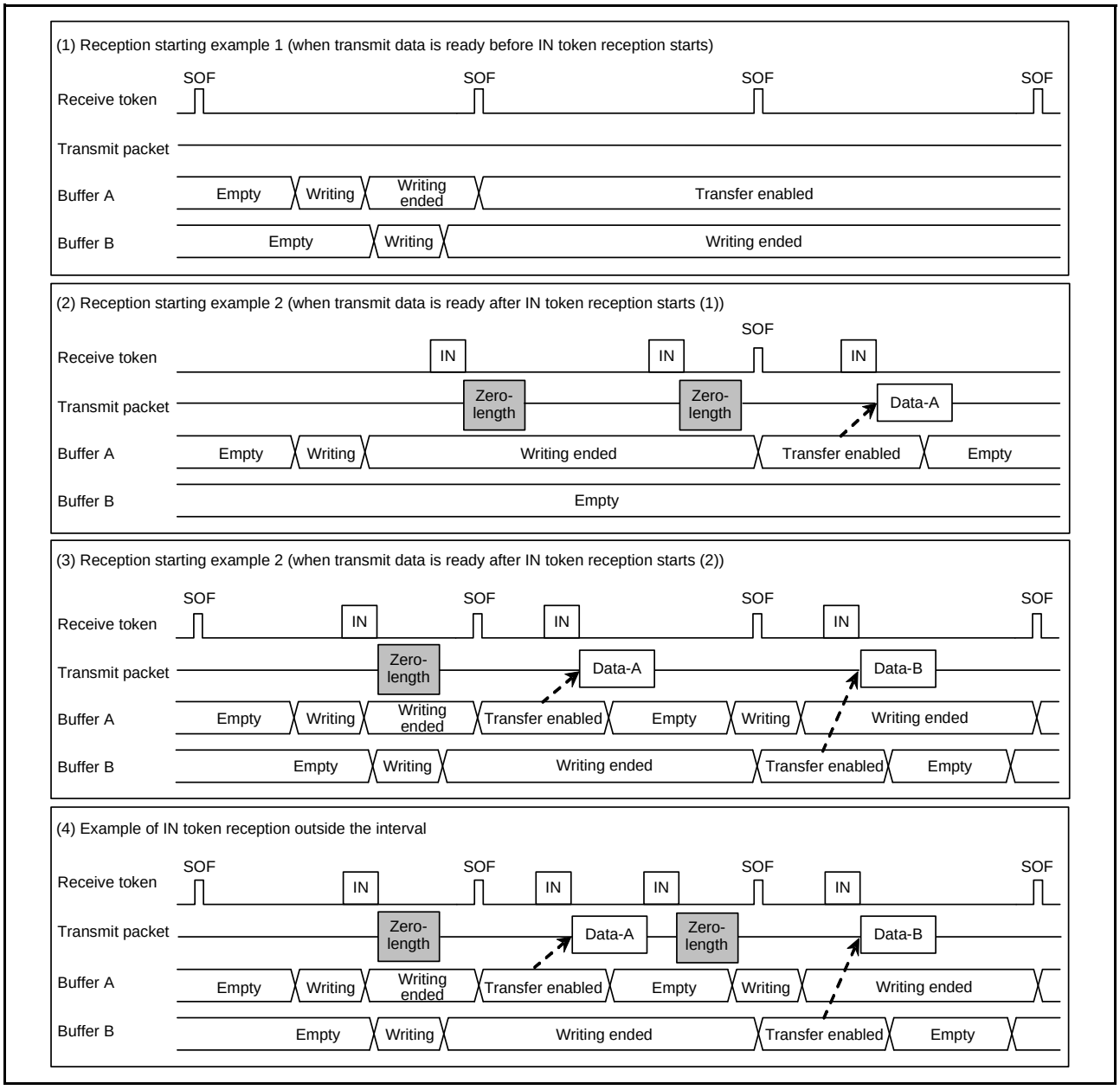


Figure 34.21 Example of Data Setup Function Operation

(5) Isochronous Transfer Transmission Buffer Flush When the Function Controller is Selected

If an SOF packet of the next frame is received without receiving an IN token in an interval frame during isochronous data transmission, the USB operates as if an IN token had been corrupted, and clears the buffer for which transmission is enabled, putting that buffer in the writing enabled state.
If a double buffer configuration is used and writing to both buffers has been completed, the buffer memory that was cleared is assumed as the data having been sent in the interval frame, and transmission is enabled for the buffer memory that is not cleared with SOF packet reception.
The timing of the buffer flush function depends on the setting of the PIPEPERI.IITV[2:0] bits.

- When the IITV[2:0] = 000b
  The buffer flush operation starts from the next frame after the pipe becomes valid.
- When the IITV[2:0] ≠ 000b
  The buffer flush operation is carried out after the first successful transaction.

Figure 34.22 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.

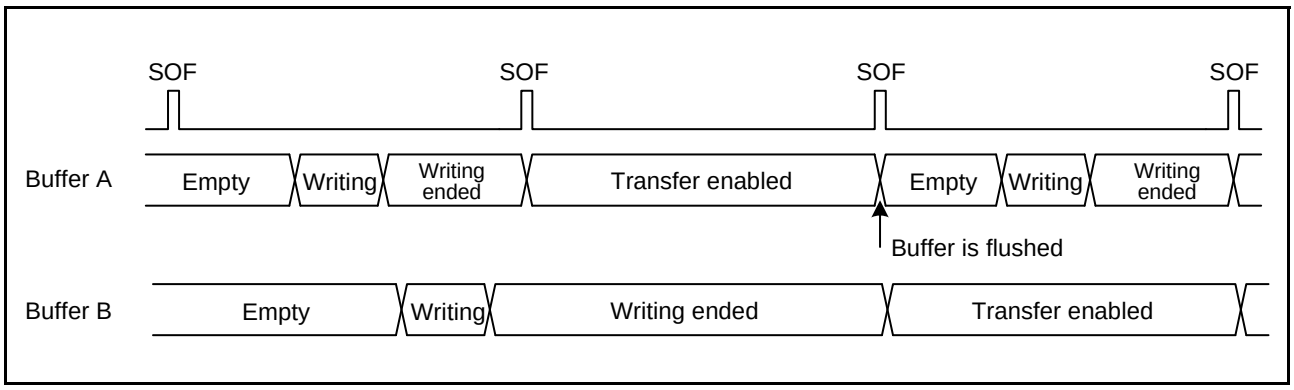


Figure 34.22 Example of Buffer Flush Operation

Figure 34.23 shows an example of interval error occurrence in the USB. There are five types of interval errors, as shown below. The interval error is generated at the timing indicated by ① in the figure, and the buffer flush function is activated.

If an interval error occurs during an IN transfers, the buffer flush function is activated; if it occurs during an OUT transfer, an NRDY interrupt is generated.

The FRMNUM.OVRN flag should be used to distinguish between NRDY interrupts such as received packet errors and overrun errors.

In response to tokens that are shaded in the figure, responses are sent according to the buffer memory status.

IN direction

- If the buffer is in the transmission enabled state, the data is transferred as a normal response.
- If the buffer is in the transmission disabled state, a zero-length packet is sent and an underrun error occurs.

OUT direction

- If the buffer is in the reception enabled state, the data is received as a normal response.
- If the buffer is in the reception disabled state, the data is discarded and an overrun error occurs.

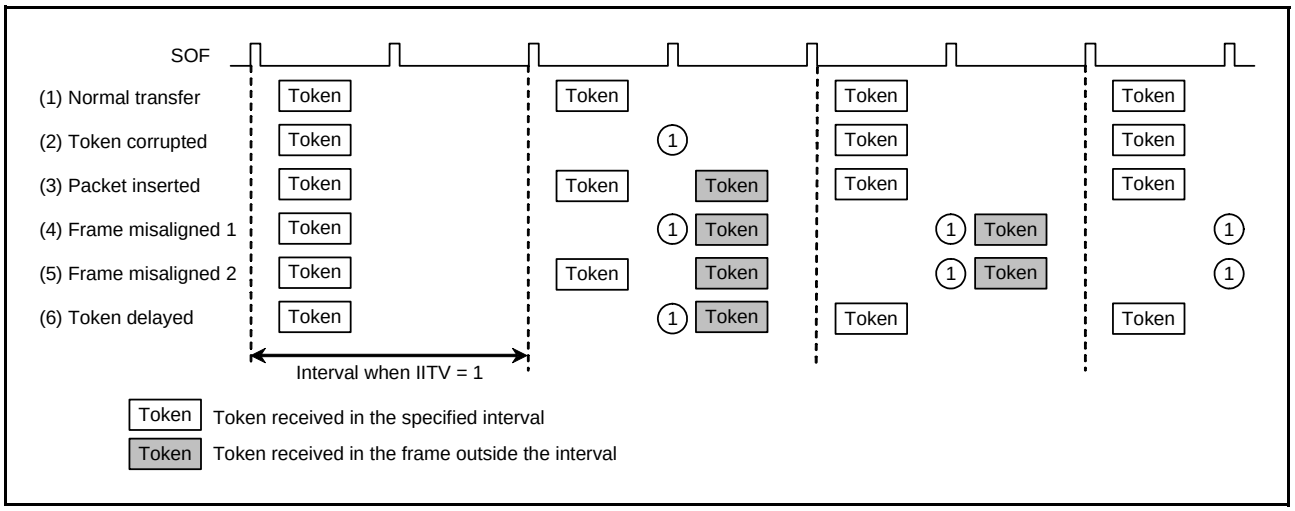


Figure 34.23 Example of Interval Error Occurrence When IITV[2:0] = 001b

34.3.10 SOF Interpolation Function

When the function controller is selected and if data could not be received at intervals of 1 ms because an SOF packet was corrupted or missing, the USB interpolates the SOF. The SOF interpolation operation begins when the USBE and SCKE bits in the SYSCFG register have been set to 1 and an SOF packet is received. The interpolation function is initialized under the following conditions.

- MCU reset
- USB bus reset
- Suspended state detected

The SOF interpolation operates as follows.

- The interpolation function is not activated until an SOF packet is received.
- After the first SOF packet is received, interpolation is carried out by counting 1 ms with an internal clock of 48 MHz.
- After the second and subsequent SOF packets are received, interpolation is carried out at the previous reception interval.
- Interpolation is not carried out in the suspended state or while a USB bus reset is being received.

The USB supports the following functions based on the SOF packet reception. These functions also operate normally with SOF interpolation, if the SOF packet was missing.

- Updating of the frame number
- SOFR interrupt timing
- Isochronous transfer interval count

If an SOF packet is missing during full-speed operation, the FRMNUM.FRNM[10:0] flags are not updated.

34.3.11 Pipe Schedule

34.3.11.1 Conditions for Generating a Transaction

When the host controller is selected and the DVSTCTR0.UACT bit has been set to 1, the USB generates a transaction under the conditions shown in Table 34.26.

Table 34.26 Conditions for Generating a Transaction

	Conditions for Generation				
Transaction	**DIR**	**PID[1:0]**	**IITV[0]**	**Buffer State**	**SUREQ**
Setup	—*1	—*1	—*1	—*1	1 setting
Control transfer data stage, status stage, bulk transfer	IN	BUF	Invalid	Receive area exists	—*1
	OUT	BUF	Invalid	Transmit data exists	—*1
Interrupt transfer	IN	BUF	Valid	Receive area exists	—*1
	OUT	BUF	Valid	Transmit data exists	—*1
Isochronous transfer	IN	BUF	Valid	*2	—*1
	OUT	BUF	Valid	*3	—*1

Note 1. Symbols (—) in the table indicate that the condition is unrelated to the generating of tokens. "Valid" indicates that, for interrupt transfers and isochronous transfers, a transaction is generated only in transfer frames that are based on the interval counter. "Invalid" indicates that a transaction is generated regardless of the interval counter.
Note 2. This indicates that a transaction is generated regardless of whether there is a receive area. If there is no receive area, however, the received data is discarded.
Note 3. This indicates that a transaction is generated regardless of whether there is any data to be transmitted. If there is no data to be transmitted, however, a zero-length packet is transmitted.

34.3.11.2 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 Pipe 1 → Pipe 2 → Pipe 6 → Pipe 7 → Pipe 8 → Pipe 9, and then, if there is a pipe for which an isochronous or interrupt transfer transaction can be generated, the transaction is generated.
2. Setup transactions for control transfers
   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 → Pipe 1 → Pipe 2 → Pipe 3 → Pipe 4 → Pipe 5, and then, if there is a pipe for which a transaction for a bulk transfer, a control transfer data stage, or a control transfer status stage can be generated, the transaction is generated.
   When a transaction is generated, processing moves to the next pipe transaction regardless of whether the response from the peripheral device is ACK or NAK. If there is time for transfer within the frame, step 3 is repeated.

34.3.11.3 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.

34.4 Usage Notes

34.4.1 Setting the Module-Stop Function

Operation of the USB module can be disabled or enabled by setting a bit in the module stop control register B (MSTPCRB). The USB is initially disabled after a reset. Registers in the USB only become accessible after it has been released from the module-stop state. For details, refer to section 11, Low Power Consumption.

35. Serial Communications Interface (SCIk, SCIm, SCIh)

This MCU has 13 independent serial communications interface (SCI) channels. The SCI consists of the SCIk module (SCI0 to SCI9), the SCIm module (SCI10 and SCI11), and the SCIh module (SCI12).
The SCIk module (SCI0 to SCI9) and SCIm module (SCI10 and 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). The SCI is also supports simple SPI interfaces, and simple I^2C-bus interfaces when configured for single-master systems.
The SCIh module includes the functions of the SCIg module, and supports an extended serial communication protocol formed of Start Frames and Information Frames.
In this section, “PCLK” for SCI0 to SCI9 and SCI12 is used to refer to PCLKB and “PCLK” for SCI10 and SCI11 is used to refer to PCLKA.

35.1 Overview

Table 35.1 lists the specifications of the SCIk module, Table 35.2 lists the specifications of the SCIm module, Table 35.3 lists the specifications of the SCIh module, and Table 35.4 lists the specifications of the individual SCI channels. Figure 35.1 shows the block diagram of SCI0 to SCI4 and SCI7 to SCI9, Figure 35.2 shows the block diagram of SCI5 and SCI6, Figure 35.3 shows the block diagram of SCI10 and SCI11, and Figure 35.4 shows the block diagram of SCI12 (SCIh).

Table 35.1 SCIk Specifications (1/2)

Item	Description
Serial communication modes	• Asynchronous • Clock synchronous • Smart card interface • Simple I^2C-bus • Simple SPI bus
Transfer speed	Bit rate specifiable with the on-chip baud rate generator.
Full-duplex communications	Transmitter: Continuous transmission possible using double-buffer structure. Receiver: Continuous reception possible using double-buffer structure.
I/O pins	Refer to Table 35.5 to Table 35.7.
Data transfer	Selectable as LSB first or MSB first transfer*1
I/O signal level inverting function	Input signal and output signal can be inverted independently.
Interrupt sources	Transmit end, transmit data empty, receive data full, receive error, and data match Completion of generation of a start condition, restart condition, or stop condition (for simple I^2C mode)
Low power consumption function	Module stop state can be set for each channel.

Table 35.1 SCIk Specifications (2/2)

Item		Description
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
	Data match detection	Compares receive data and comparison data, and generates interrupt when they are matched.
	Start-bit detection	Low level or falling edge is selectable.
	Receive data sampling timing adjustment	Sampling point of receive data can be changed to front or rear of center of bit.
	Transmit signal transition timing adjustment	Falling or rising edge of the transmit data can be delayed.
	Break detection	When a framing error occurs, a break can be detected by reading the RXDn pin level directly or reading the SPTR.RXDMON flag.
	Clock source	An internal or external clock can be selected. Transfer rate clock input from the TMR can be used. (SCI5 and SCI6)
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Multi-processor communications function	Serial communication among multiple processors
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters.
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun error
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
Smart card interface mode	Error processing	An error signal can be automatically transmitted when detecting a parity error during reception
		Data can be automatically retransmitted when receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are supported.
Simple I^2C mode	Transfer format	I^2C-bus format
	Operating mode	Master (single-master operation only)
	Transfer rate	Fast mode is supported (refer to section 35.2.13, Bit Rate Register (BRR) to set the transfer rate).
	Noise cancellation	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters, and the interval for noise cancellation is adjustable.
Simple SPI bus	Data length	8 bits
	Detection of errors	Overrun error
	SS input pin function	Applying the high level to the SSn# pin can cause the output pins to enter the high-impedance state.
	Clock settings	Four kinds of settings for clock phase and clock polarity are selectable.
Bit rate modulation function		Correction of outputs from the on-chip baud rate generator can reduce errors.
Event link function (supported by SCI5 only)		Error (receive error or error signal detection) event output
		Receive data full event output
		Transmit data empty event output
		Transmit end event output

Note 1. In simple I^2C mode, only MSB first is available.

Table 35.2 SCIm Specifications (1/2)

Item		Description
Serial communication modes		• Asynchronous • Clock synchronous • Smart card interface • Simple I²C-bus • Simple SPI bus
Transfer speed		Bit rate specifiable with the on-chip baud rate generator.
Full-duplex communications		Transmitter: Continuous transmission possible using double-buffer structure. Receiver: Continuous reception possible using double-buffer structure.
I/O pins		Refer to Table 35.5 to Table 35.7.
Data transfer		Selectable as LSB first or MSB first transfer*1
I/O signal level inverting function		Input signal and output signal can be inverted independently.
Interrupt sources		Transmit end, transmit data empty, receive data full, receive error, receive data ready, and data match Completion of generation of a start condition, restart condition, or stop condition (for simple I²C mode)
Low power consumption function		Module stop state can be set for each channel.
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
	Transmit/receive FIFO	16-stage FIFOs for transmit and receive buffers
	Data match detection	Compares receive data and comparison data, and generates interrupt when they are matched
	Start-bit detection	Low level or falling edge is selectable.
	Receive data sampling timing adjustment	Sampling point of receive data can be changed to front or rear of center of bit.
	Transmit signal transition timing adjustment	Falling or rising edge of the transmit data can be delayed.
	Break detection	When a framing error occurs, a break can be detected by reading the RXDn pin level directly or reading the SPTR.RXDMON flag.
	Clock source	An internal or external clock can be selected.
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Multi-processor communications function	Serial communication among multiple processors
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters.
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun error
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
	Transmit/receive FIFO	16-stage FIFOs for transmit and receive buffers
Smart card interface mode	Error processing	An error signal can be automatically transmitted when detecting a parity error during reception
		Data can be automatically retransmitted when receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are supported.

Table 35.2 SCIm Specifications (2/2)

Item		Description
Simple I2C mode	Transfer format	I2C-bus format
	Operating mode	Master (single-master operation only)
	Transfer rate	Fast mode is supported (refer to section 35.2.13, Bit Rate Register (BRR) to set the transfer rate).
	Noise cancellation	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters, and the interval for noise cancellation is adjustable.
Simple SPI bus	Data length	8 bits
	Detection of errors	Overrun error
	SS input pin function	Applying the high level to the SSn# pin can cause the output pins to enter the high-impedance state.
	Clock settings	Four kinds of settings for clock phase and clock polarity are selectable.
Bit rate modulation function		Correction of outputs from the on-chip baud rate generator can reduce errors.

Note 1. In simple I2C mode, only MSB first is available.

Table 35.3 SCIh Specifications (1/2)

Item		Description
Serial communication modes		• Asynchronous • Clock synchronous • Smart card interface • Simple I2C-bus • Simple SPI bus
Transfer speed		Bit rate specifiable with the on-chip baud rate generator.
Full-duplex communications		Transmitter: Continuous transmission possible using double-buffer structure. Receiver: Continuous reception possible using double-buffer structure.
I/O pins		Refer to Table 35.5 to Table 35.8.
Data transfer		Selectable as LSB first or MSB first transfer*1
Interrupt sources		Transmit end, transmit data empty, receive data full, and receive error Completion of generation of a start condition, restart condition, or stop condition (for simple I2C mode)
Low power consumption function		Module stop state can be set.
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
	Start-bit detection	Low level or falling edge is selectable.
	Break detection	When a framing error occurs, a break can be detected by reading the RXDn pin level directly.
	Clock source	An internal or external clock can be selected. Transfer rate clock input from the TMR can be used.
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Multi-processor communications function	Serial communication among multiple processors
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters.
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun error
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.

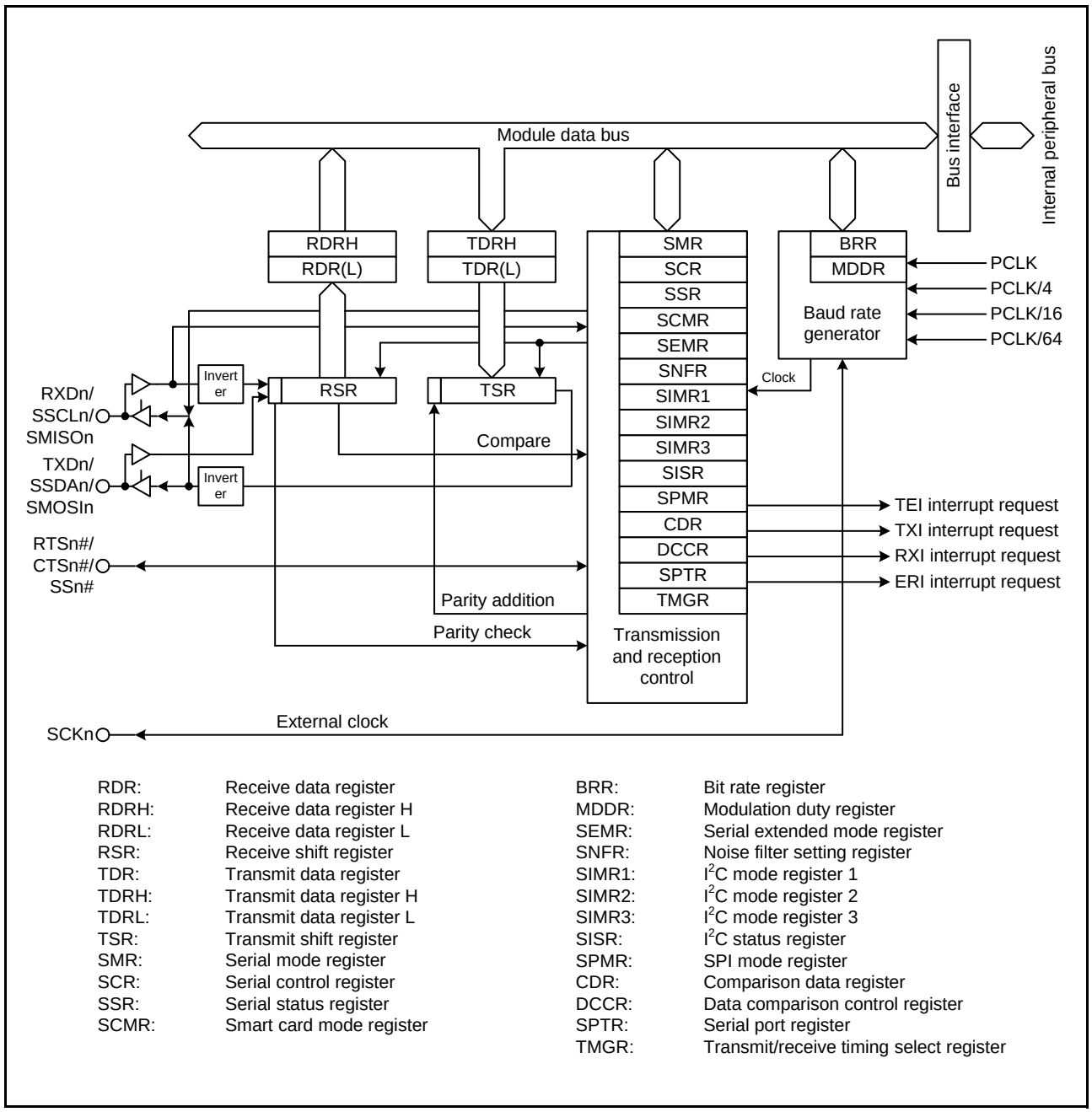
Table 35.3 SCIh Specifications (2/2)

Item		Description
Smart card interface mode	Error processing	An error signal can be automatically transmitted when detecting a parity error during reception
		Data can be automatically retransmitted when receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are supported.
Simple I^2C mode	Transfer format	I^2C-bus format
	Operating mode	Master (single-master operation only)
	Transfer rate	Fast mode is supported (refer to section 35.2.13, Bit Rate Register (BRR) to set the transfer rate).
	Noise cancellation	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters, and the interval for noise cancellation is adjustable.
Simple SPI bus	Data length	8 bits
	Detection of errors	Overrun error
	SS input pin function	Applying the high level to the SSn# pin can cause the output pins to enter the high-impedance state.
	Clock settings	Four kinds of settings for clock phase and clock polarity are selectable.
Extended serial mode	Start Frame transmission	• Output of a low level as the Break Field over a specified width and generation of interrupts on completion • Detection of bus collisions and the generation of interrupts on detection
	Start Frame reception	• Detection of the Break Field low width and generation of an interrupt on detection • Comparison of Control Fields 0 and 1 and generation of an interrupt when the two match • Two kinds of data for comparison (primary and secondary) can be set in Control Field 1. • A priority interrupt bit can be set in Control Field 1. • Handling of Start Frames that do not include a Break Field • Handling of Start Frames that do not include a Control Field 0 • Function for measuring bit rates
	I/O control function	• Selectable polarity for TXDX12 and RXDX12 signals • Selection of a digital filter for the RXDX12 signal • Half-duplex operation employing RXDX12 and TXDX12 signals multiplexed on the same pin • Selectable timing for the sampling of data received through RXDX12
	Timer function	• Usable as a reload timer
Bit rate modulation function		Correction of outputs from the on-chip baud rate generator can reduce errors.

Note 1. In simple I^2C mode, only MSB first is available.

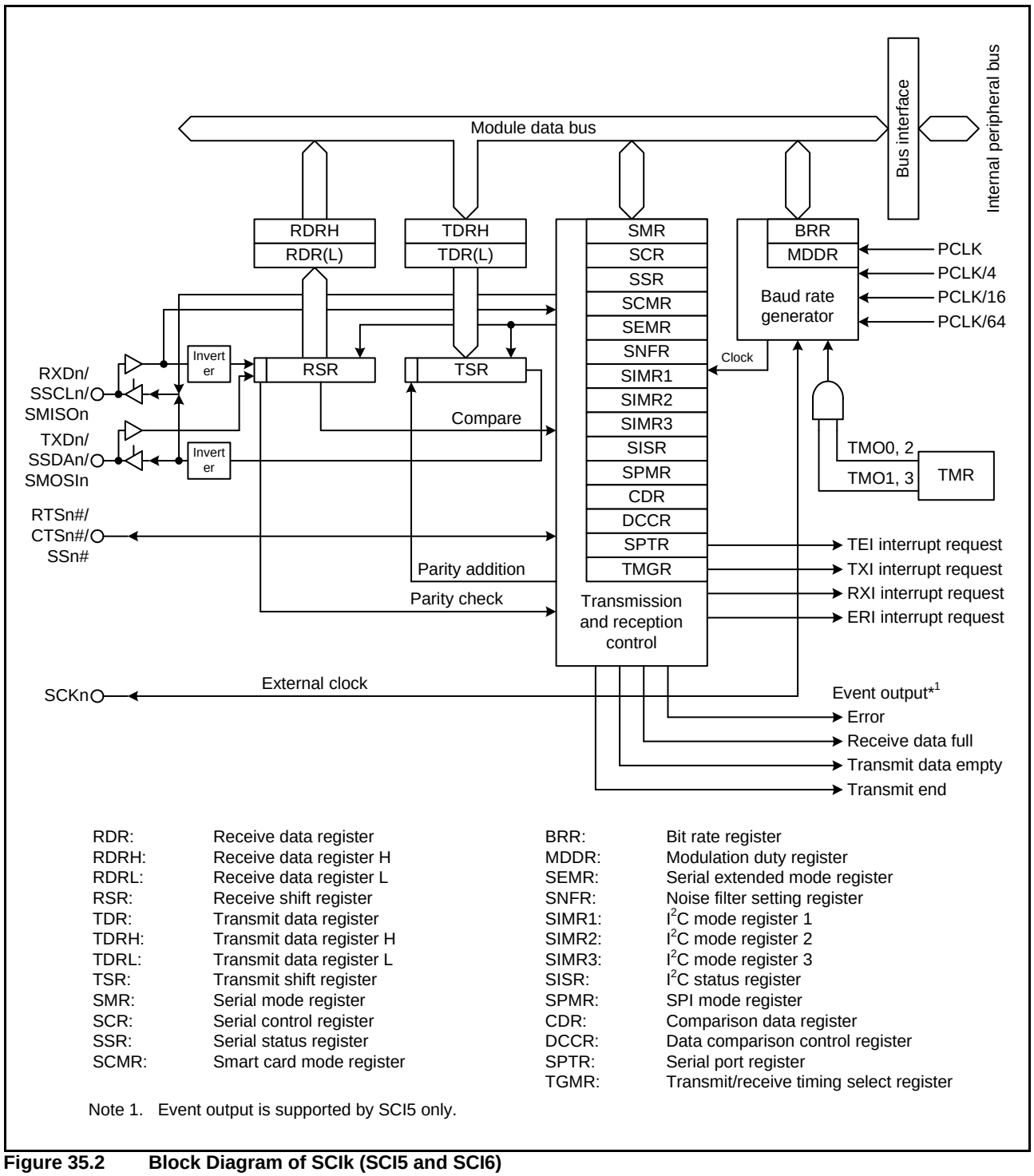
Table 35.4 Functions of SCI Channels

Item	SCI0 to SCI4, SCI7 to SCI9	SCI5	SCI6	SCI10, SCI11	SCI12
Asynchronous mode	Available	Available	Available	Available	Available
Clock synchronous mode	Available	Available	Available	Available	Available
Smart card interface mode	Available	Available	Available	Available	Available
Simple I^2C mode	Available	Available	Available	Available	Available
Simple SPI mode	Available	Available	Available	Available	Available
FIFO mode	Not available	Not available	Not available	Available	Not available
Data match detection	Available	Available	Available	Available	Not available
Extended serial mode	Not available	Not available	Not available	Not available	Available
TMR clock input	Not available	Available	Available	Not available	Available
Event link function	Not available	Available	Not available	Not available	Not available
Peripheral module clock	PCLKB	PCLKB	PCLKB	PCLKA	PCLKB



RDR:	Receive data register	BRR:	Bit rate register
RDRH:	Receive data register H	MDDR:	Modulation duty register
RDRL:	Receive data register L	SEMR:	Serial extended mode register
RSR:	Receive shift register	SNFR:	Noise filter setting register
TDR:	Transmit data register	SIMR1:	I^2C mode register 1
TDRH:	Transmit data register H	SIMR2:	I^2C mode register 2
TDRL:	Transmit data register L	SIMR3:	I^2C mode register 3
TSR:	Transmit shift register	SISR:	I^2C status register
SMR:	Serial mode register	SPMR:	SPI mode register
SCR:	Serial control register	CDR:	Comparison data register
SSR:	Serial status register	DCCR:	Data comparison control register
SCMR:	Smart card mode register	SPTR:	Serial port register
		TMGR:	Transmit/receive timing select register

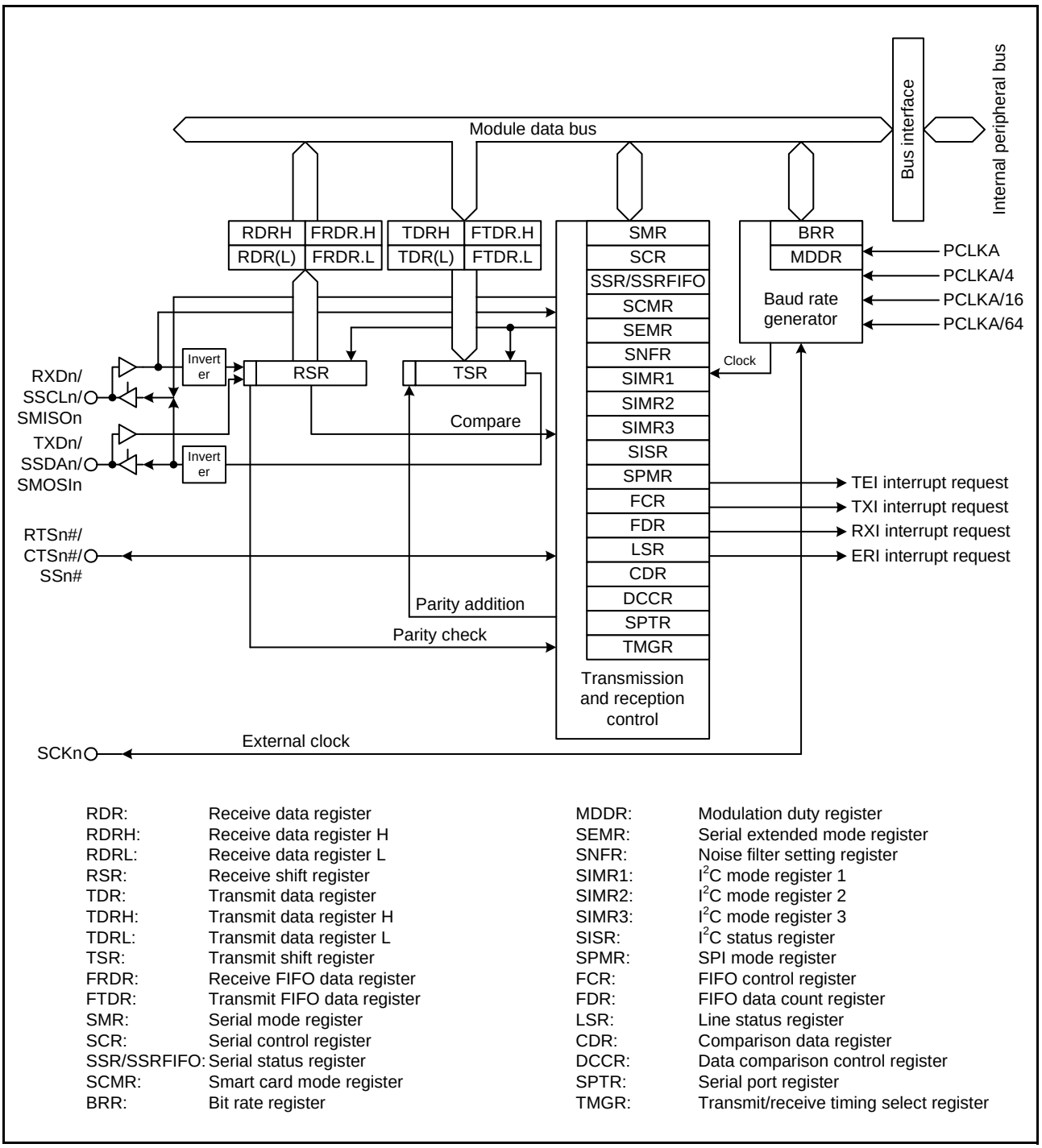
Figure 35.1 Block Diagram of SCIk (SCI0 to SCI4 and SCI7 to SCI9)



RDR:	Receive data register	BRR:	Bit rate register
RDRH:	Receive data register H	MDDR:	Modulation duty register
RDRL:	Receive data register L	SEMR:	Serial extended mode register
RSR:	Receive shift register	SNFR:	Noise filter setting register
TDR:	Transmit data register	SIMR1:	I^2C mode register 1
TDRH:	Transmit data register H	SIMR2:	I^2C mode register 2
TDRL:	Transmit data register L	SIMR3:	I^2C mode register 3
TSR:	Transmit shift register	SISR:	I^2C status register
SMR:	Serial mode register	SPMR:	SPI mode register
SCR:	Serial control register	CDR:	Comparison data register
SSR:	Serial status register	DCCR:	Data comparison control register
SCMR:	Smart card mode register	SPTR:	Serial port register
		TGMR:	Transmit/receive timing select register

Note 1. Event output is supported by SCI5 only.

Figure 35.2 Block Diagram of SCIk (SCI5 and SCI6)



RDR:	Receive data register	MDDR:	Modulation duty register
RDRH:	Receive data register H	SEMR:	Serial extended mode register
RDRL:	Receive data register L	SNFR:	Noise filter setting register
RSR:	Receive shift register	SIMR1:	I^2C mode register 1
TDR:	Transmit data register	SIMR2:	I^2C mode register 2
TDRH:	Transmit data register H	SIMR3:	I^2C mode register 3
TDRL:	Transmit data register L	SISR:	I^2C status register
TSR:	Transmit shift register	SPMR:	SPI mode register
FRDR:	Receive FIFO data register	FCR:	FIFO control register
FTDR:	Transmit FIFO data register	FDR:	FIFO data count register
SMR:	Serial mode register	LSR:	Line status register
SCR:	Serial control register	CDR:	Comparison data register
SSR/SSRFIFO:	Serial status register	DCCR:	Data comparison control register
SCMR:	Smart card mode register	SPTR:	Serial port register
BRR:	Bit rate register	TMGR:	Transmit/receive timing select register

Figure 35.3 Block Diagram of SCIm (SCI10 and SCI11)

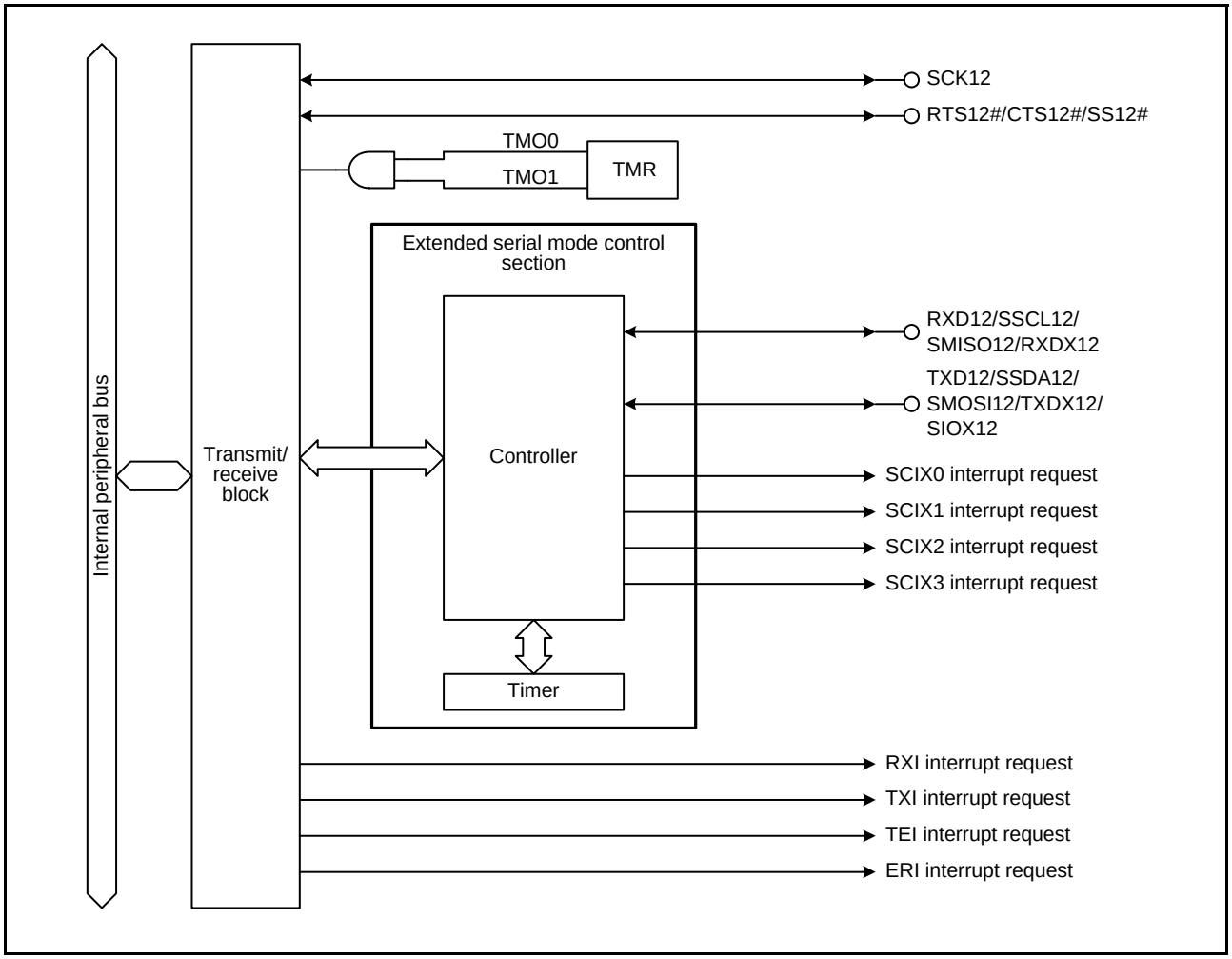


Figure 35.4 Block Diagram of SCIh (SCI12)

Table 35.5 to Table 35.8 list the pin configuration of the SCIs for the individual modes.

Table 35.5 SCI Pin Configuration in Asynchronous Mode and Clock Synchronous Mode

Channel	Pin Name	I/O	Function
SCI0	SCK0	I/O	SCI0 clock input/output
	RXD0	Input	SCI0 receive data input
	TXD0	Output	SCI0 transmit data output
	CTS0#/RTS0#	I/O	SCI0 transfer start control input/output
SCI1	SCK1	I/O	SCI1 clock input/output
	RXD1	Input	SCI1 receive data input
	TXD1	Output	SCI1 transmit data output
	CTS1#/RTS1#	I/O	SCI1 transfer start control input/output
SCI2	SCK2	I/O	SCI2 clock input/output
	RXD2	Input	SCI2 receive data input
	TXD2	Output	SCI2 transmit data output
	CTS2#/RTS2#	I/O	SCI2 transfer start control input/output
SCI3	SCK3	I/O	SCI3 clock input/output
	RXD3	Input	SCI3 receive data input
	TXD3	Output	SCI3 transmit data output
	CTS3#/RTS3#	I/O	SCI3 transfer start control input/output
SCI4	SCK4	I/O	SCI4 clock input/output
	RXD4	Input	SCI4 receive data input
	TXD4	Output	SCI4 transmit data output
	CTS4#/RTS4#	I/O	SCI4 transfer start control input/output
SCI5	SCK5	I/O	SCI5 clock input/output
	RXD5	Input	SCI5 receive data input
	TXD5	Output	SCI5 transmit data output
	CTS5#/RTS5#	I/O	SCI5 transfer start control input/output
SCI6	SCK6	I/O	SCI6 clock input/output
	RXD6	Input	SCI6 receive data input
	TXD6	Output	SCI6 transmit data output
	CTS6#/RTS6#	I/O	SCI6 transfer start control input/output
SCI7	SCK7	I/O	SCI7 clock input/output
	RXD7	Input	SCI7 receive data input
	TXD7	Output	SCI7 transmit data output
	CTS7#/RTS7#	I/O	SCI7 transfer start control input/output
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

Table 35.5 SCI Pin Configuration in Asynchronous Mode and Clock Synchronous Mode

Channel	Pin Name	I/O	Function
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 35.6 SCI Pin Configuration in Simple I^2C Mode

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
SCI3	SSCL3	I/O	SCI3 I^2C clock input/output
	SSDA3	I/O	SCI3 I^2C data input/output
SCI4	SSCL4	I/O	SCI4 I^2C clock input/output
	SSDA4	I/O	SCI4 I^2C data input/output
SCI5	SSCL5	I/O	SCI5 I^2C clock input/output
	SSDA5	I/O	SCI5 I^2C data input/output
SCI6	SSCL6	I/O	SCI6 I^2C clock input/output
	SSDA6	I/O	SCI6 I^2C data input/output
SCI7	SSCL7	I/O	SCI7 I^2C clock input/output
	SSDA7	I/O	SCI7 I^2C data input/output
SCI8	SSCL8	I/O	SCI8 I^2C clock input/output
	SSDA8	I/O	SCI8 I^2C data input/output
SCI9	SSCL9	I/O	SCI9 I^2C clock input/output
	SSDA9	I/O	SCI9 I^2C data input/output
SCI10	SSCL10	I/O	SCI10 I^2C clock input/output
	SSDA10	I/O	SCI10 I^2C data input/output
SCI11	SSCL11	I/O	SCI11 I^2C clock input/output
	SSDA11	I/O	SCI11 I^2C data input/output
SCI12	SSCL12	I/O	SCI12 I^2C clock input/output
	SSDA12	I/O	SCI12 I^2C data input/output

Table 35.7 SCI Pin Configuration in Simple SPI Mode (1/2)

Channel	Pin Name	I/O	Function
SCI0	SCK0	I/O	SCI0 clock input/output
	SMISO0	I/O	SCI0 slave transmit data input/output
	SMOSI0	I/O	SCI0 master transmit data input/output
	SS0#	Input	SCI0 chip select input
SCI1	SCK1	I/O	SCI1 clock input/output
	SMISO1	I/O	SCI1 slave transmit data input/output
	SMOSI1	I/O	SCI1 master transmit data input/output
	SS1#	Input	SCI1 chip select input
SCI2	SCK2	I/O	SCI2 clock input/output
	SMISO2	I/O	SCI2 slave transmit data input/output
	SMOSI2	I/O	SCI2 master transmit data input/output
	SS2#	Input	SCI2 chip select input
SCI3	SCK3	I/O	SCI3 clock input/output
	SMISO3	I/O	SCI3 slave transmit data input/output
	SMOSI3	I/O	SCI3 master transmit data input/output
	SS3#	Input	SCI3 chip select input
SCI4	SCK4	I/O	SCI4 clock input/output
	SMISO4	I/O	SCI4 slave transmit data input/output
	SMOSI4	I/O	SCI4 master transmit data input/output
	SS4#	Input	SCI4 chip select input
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

Table 35.7 SCI Pin Configuration in Simple SPI Mode (2/2)

Channel	Pin Name	I/O	Function
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 35.8 SCI Pin Configuration in Extended Serial Mode

Channel	Pin Name	I/O	Function
SCI12	RXDX12	Input	SCI12 receive data input
	TXDX12	Output	SCI12 transmit data output
	SIOX12	I/O	SCI12 transfer data input/output

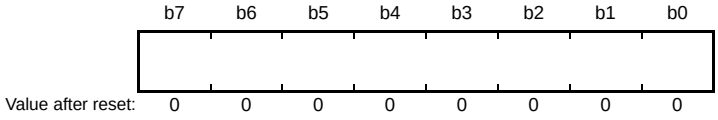
35.2 Register Descriptions

35.2.1 Receive Shift Register (RSR)

The RSR register is a shift register which is used to receive serial data input from the RXDn pin and converts it into parallel data.
When one frame of data has been received, it is automatically transferred to the RDR register.
The RSR register cannot be directly accessed by the CPU.

35.2.2 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 000D 0045h, SCI11.RDR 000D 0065h, SCI12.RDR 0008 B305h



The RDR register is an 8-bit register that stores receive data.
When one frame of serial data has been received, the received serial data is transferred from the RSR register to the RDR register. Then the RSR register can receive the next data.
Since the RSR and RDR registers function as a double buffer in this way, continuous receive operations can be performed.
Read the RDR register 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 the RDR register, an overrun error occurs.
The RDR register cannot be written to by the CPU.

35.2.3 Receive Data Register H, L, HL (RDRH, RDRL, RDRHL)

- Receive Data Register H (RDRH)

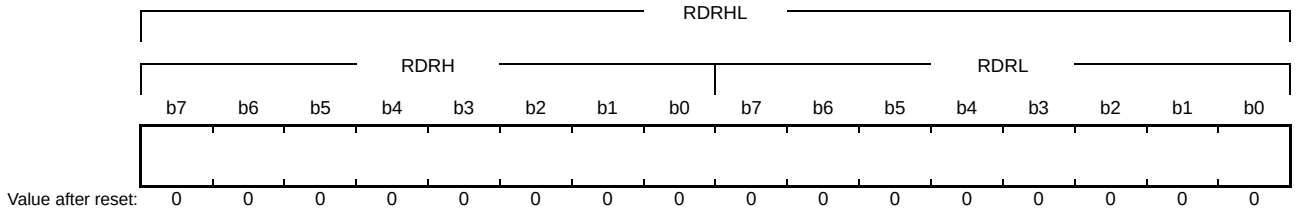
Address(es): SCI0.RDRH 0008 A010h, SCI1.RDRH 0008 A030h, SCI2.RDRH 0008 A050h, SCI3.RDRH 0008 A070h, SCI4.RDRH 0008 A090h, SCI5.RDRH 0008 A0B0h, SCI6.RDRH 0008 A0D0h, SCI7.RDRH 0008 A0F0h, SCI8.RDRH 0008 A110h, SCI9.RDRH 0008 A130h, SCI10.RDRH 000D 0050h, SCI11.RDRH 000D 0070h, SCI12.RDRH 0008 B310h

- Receive Data Register L (RDRL)

Address(es): SCI0.RDRL 0008 A011h, SCI1.RDRL 0008 A031h, SCI2.RDRL 0008 A051h, SCI3.RDRL 0008 A071h, SCI4.RDRL 0008 A091h, SCI5.RDRL 0008 A0B1h, SCI6.RDRL 0008 A0D1h, SCI7.RDRL 0008 A0F1h, SCI8.RDRL 0008 A111h, SCI9.RDRL 0008 A131h, SCI10.RDRL 000D 0051h, SCI11.RDRL 000D 0071h, SCI12.RDRL 0008 B311h

- Receive Data Register HL (RDRHL)

Address(es): SCI0.RDRHL 0008 A010h, SCI1.RDRHL 0008 A030h, SCI2.RDRHL 0008 A050h, SCI3.RDRHL 0008 A070h, SCI4.RDRHL 0008 A090h, SCI5.RDRHL 0008 A0B0h, SCI6.RDRHL 0008 A0D0h, SCI7.RDRHL 0008 A0F0h, SCI8.RDRHL 0008 A110h, SCI9.RDRHL 0008 A130h, SCI10.RDRHL 000D 0050h, SCI11.RDRHL 000D 0070h, SCI12.RDRHL 0008 B310h



The RDRH and RDRL registers are 8-bit registers that store receive data. Use these registers when asynchronous mode and 9-bit data length are selected.

The RDRL register is the shadow register of the RDR register; i.e. access to the RDRL register is equivalent to access to the RDR register.

After one frame of data is received, the received data is transferred from the RSR register to these registers, thus allowing the RSR register to receive the next data.

The RSR, RDRH and RDRL registers have a double-buffered construction to enable continuous reception.

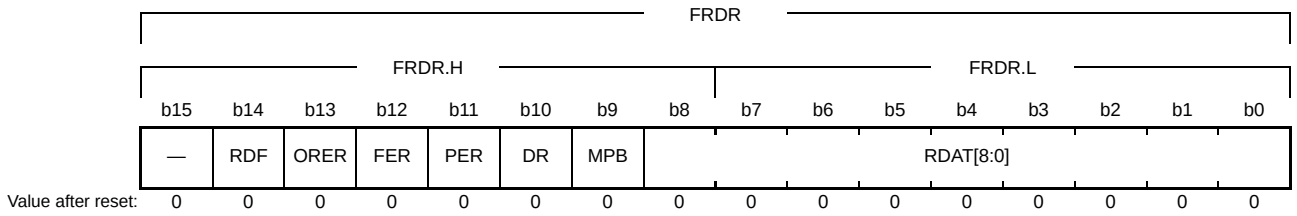
Read the RDRH and RDRL registers should be performed only once in the order from the RDRH register to the RDRL register when a receive data full interrupt (RXI) request is issued. Note that an overrun error occurs when the next frame of data is received before the received data has been read from the RDRL register.

The CPU cannot write to the RDRH and RDRL registers. Bits 0 to 7 in the RDRH register are fixed to 0. These bits are read as 0.

The RDRHL register can be accessed in 16-bit units.

35.2.4 Receive FIFO Data Register (FRDR)

Address(es): SCI10.FRDR 000D 0050h, SCI11.FRDR 000D 0070h,
SCI10.FRDR.H 000D 0050h, SCI11.FRDR.H 000D 0070h,
SCI10.FRDR.L 000D 0051h, SCI11.FRDR.L 000D 0071h



Bit	Symbol	Bit Name	Description	R/W
b8 to b0	RDAT[8:0]	Receive Data	Received data can be read	R
b9	MPB	Multi-Processor Bit Monitor Flag	Value of the multi-processor bit in the reception frame 0: Data transmission cycles 1: ID transmission cycles	R
b10	DR	Receive Data Ready Flag*1	0: The FRDR register does not contain valid data 1: The FRDR register contains valid data	R
b11	PER	Parity Error Flag	0: No parity error occurred 1: A parity error has occurred	R
b12	FER	Framing Error Flag	0: No framing error occurred 1: A framing error has occurred	R
b13	ORER	Overrun Error Flag*1	0: No overrun error occurred 1: An overrun error has occurred	R
b14	RDF	Receive FIFO Full Flag*1	0: The number of data stored in the receive FIFO is less than the threshold value 1: The number of data stored in the receive FIFO is equal to or greater than the threshold value	R
b15	—	Reserved	The read value is undefined	R

Note 1. These flags are the same as the flags of the same name in the SSRFIFO register. To clear these flags, clear the corresponding flag in the SSRFIFO register.

This register is used to read the data at the top of the sixteen-stage receive FIFO. This register is enabled when the FCR.FM bit is 1 (FIFO mode).

Reading the FRDR register when the receive FIFO is empty results in an undefined value.

When reading the lower 8 bits (FRDR.L), the value of the FRDR register is updated with the next data in the receive FIFO. When reading the upper 8 bits (FRDR.H), the FRDR register is not updated. When reading the FRDR register in 8-bit units, read first from the FRDR.H register and then the FRDR.L register.

When the character length is 8 bits 0 is stored in the RDAT[8] bit and when the character length is 7 bits 00b is stored in the RDAT[8:7] bits.

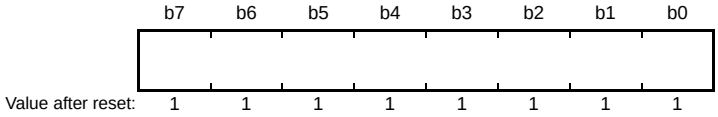
The MPB flag holds the value of the multi-processor bit that was added to the data at the top of the receive FIFO. The SSR.MPB flag is not used when the FCR.FM bit is 1 (FIFO mode).

The FER and PER flags indicate whether the relevant error exists in the data at the top of the receive FIFO. Its value is updated each time the FRDR register is read.

The RDF, ORER, and DR flags are the same as the flags of the same name in the SSRFIFO register. When these flags are read with a value of 1, the flag can be cleared simply by writing 0 to the corresponding flag in the SSRFIFO register.

35.2.5 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 000D 0043h, SCI11.TDR 000D 0063h, SCI12.TDR 0008 B303h



The TDR register is an 8-bit register that stores transmit data.

When the SCI detects that the TSR register is empty, it transfers the transmit data written in the TDR register to the TSR register and starts transmission.

The double-buffered structures of the TDR register and the TSR register enable continuous serial transmission. If the next transmit data has already been written to the TDR register when one frame of data is transmitted, the SCI transfers the written data to the TSR register to continue transmission.

The CPU is able to read from or write to the TDR register at any time. Only write transmit data to the TDR register once after each instance of the transmit data empty interrupt (TXI).

35.2.6 Transmit Data Register H, L, HL (TDRH, TDRL, TDRHL)

- Transmit Data Register H (TDRH)

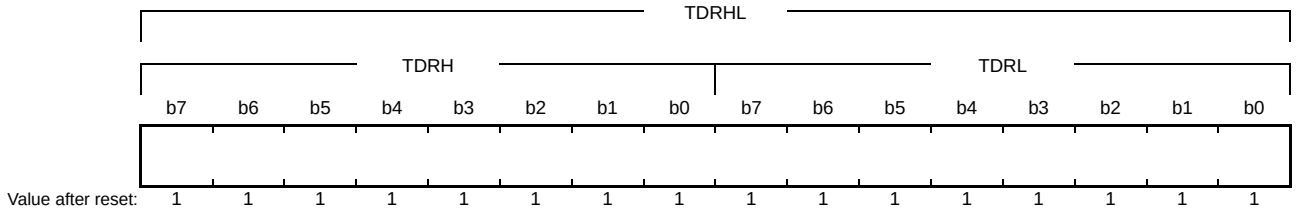
Address(es): SCI0.TDRH 0008 A00Eh, SCI1.TDRH 0008 A02Eh, SCI2.TDRH 0008 A04Eh, SCI3.TDRH 0008 A06Eh, SCI4.TDRH 0008 A08Eh, SCI5.TDRH 0008 A0AEh, SCI6.TDRH 0008 A0CEh, SCI7.TDRH 0008 A0EEh, SCI8.TDRH 0008 A10Eh, SCI9.TDRH 0008 A12Eh, SCI10.TDRH 000D 004Eh, SCI11.TDRH 000D 006Eh, SCI12.TDRH 0008 B30Eh

- Transmit Data Register L (TDRL)

Address(es): SCI0.TDRL 0008 A00Fh, SCI1.TDRL 0008 A02Fh, SCI2.TDRL 0008 A04Fh, SCI3.TDRL 0008 A06Fh, SCI4.TDRL 0008 A08Fh, SCI5.TDRL 0008 A0AFh, SCI6.TDRL 0008 A0CFh, SCI7.TDRL 0008 A0EFh, SCI8.TDRL 0008 A10Fh, SCI9.TDRL 0008 A12Fh, SCI10.TDRL 000D 004Fh, SCI11.TDRL 000D 006Fh, SCI12.TDRL 0008 B30Fh

- Transmit Data Register HL (TDRHL)

Address(es): SCI0.TDRHL 0008 A00Eh, SCI1.TDRHL 0008 A02Eh, SCI2.TDRHL 0008 A04Eh, SCI3.TDRHL 0008 A06Eh, SCI4.TDRHL 0008 A08Eh, SCI5.TDRHL 0008 A0AEh, SCI6.TDRHL 0008 A0CEh, SCI7.TDRHL 0008 A0EEh, SCI8.TDRHL 0008 A10Eh, SCI9.TDRHL 0008 A12Eh, SCI10.TDRHL 000D 004Eh, SCI11.TDRHL 000D 006Eh, SCI12.TDRHL 0008 B30Eh



The TDRH and TDRL registers are 8-bit registers that store transmit data. Use these registers when asynchronous mode and 9-bit data length are selected.

The TDRL register is the shadow register of the TDR register; i.e. access to the TDRL register is equivalent to access to the TDR register.

When empty space is detected in the TSR register, the transmit data stored in the TDRH and TDRL registers is transferred to the TSR register; i.e., transmitting is started.

The TSR, TDRH and TDRL registers have a double-buffered construction to realize continuous reception. When the next data to be transmitted is stored in the TDRL register after one frame of data has been transmitted, the transmitting operation is continued by transfer to the TSR register.

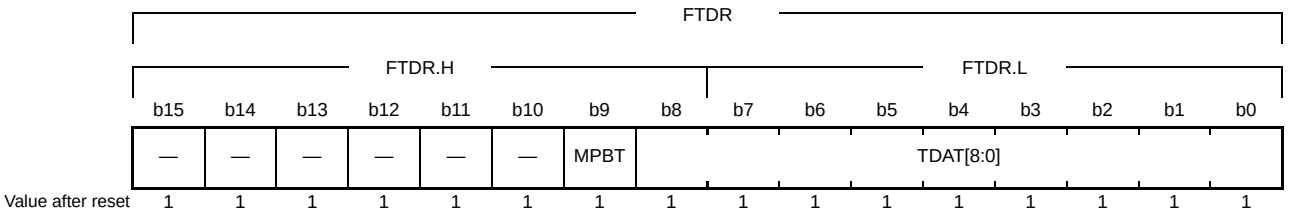
The CPU can read and write to the TDRH and TDRL registers. Bits 0 to 7 in the RDRH register are fixed to 1. These bits are read as 1. The write value should be 1.

Writing transmit data to the TDRH and TDRL registers should be performed only once in the order from the TDRH register to the TDRL register when a transmit data empty interrupt (TXI) request is issued.

The TDRHL register can be accessed in 16-bit units.

35.2.7 Transmit FIFO Data Register (FTDR)

Address(es): SCI10.FTDR 000D 004Eh, SCI11.FTDR 000D 006Eh,
SCI10.FTDR.H 000D 004Eh, SCI11.FTDR.H 000D 006Eh,
SCI10.FTDR.L 000D 004Fh, SCI11.FTDR.L 000D 006Fh



Bit	Symbol	Bit Name	Description	R/W
b8 to b0	TDAT[8:0]	Transmit Data	Specifies the data to be transmitted	W
b9	MPBT	Transmit Multi-Processor	Specifies the value of the multi-processor bit in the transmit frame 0: Data transmission cycles 1: ID transmission cycles	W
b15 to b10	—	Reserved	The write value should be 1	W

This register is used to write data to the 16-stage transmit FIFO. This register is enabled when the FCR.FM bit is 1 (FIFO mode) and the SCR.TE bit is 1.
When the transmit FIFO contains 16 frames of data, no transmit data can be set in the FTDR register.
When writing a value to the lower 8 bits (FTDR.L), the data in the FTDR register is transferred to the transmit FIFO.
When writing a value to the upper 8 bits (FTDR.H), no data is transferred to the transmit FIFO. When writing a 16-bit data to the FTDR register in 8-bit units, write first to the FTDR.H register and then the FTDR.L register.

MPBT Bit (Transmit Multi-Processor)

This bit is used to set the value of the multi-processor bit to add to the transmit frame. The SSR.MPBT bit is not used when the FCR.FM bit is 1 (FIFO mode).

35.2.8 Transmit Shift Register (TSR)

The TSR register is a shift register that transmits serial data.
To perform serial data transmission, the SCI first automatically transfers transmit data from the TDR register to the TSR register, and then sends the data to the TXDn pin.
The TSR register cannot be directly accessed by the CPU.

35.2.9 Serial Mode Register (SMR)

Some bits in the SMR register have different functions in smart card interface mode and non-smart card interface mode.

(1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI0.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 000D 0040h, SCI11.SMR 000D 0060h, SCI12.SMR 0008 B300h

	b7	b6	b5	b4	b3	b2	b1	b0
	CM	CHR	PE	PM	STOP	MP	CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK (n = 0)*1 0 1: PCLK/4 (n = 1)*1 1 0: PCLK/16 (n = 2)*1 1 1: PCLK/64 (n = 3)*1	R/W*4
b2	MP	Multi-Processor Mode	(Valid only in asynchronous mode) 0: Multi-processor communications function is disabled 1: Multi-processor communications function is enabled	R/W*4
b3	STOP	Stop Bit Length	(Valid only in asynchronous mode) 0: 1 stop bit 1: 2 stop bits	R/W*4
b4	PM	Parity Mode	(Valid only when the PE bit is 1) 0: Selects even parity 1: Selects odd parity	R/W*4
b5	PE	Parity Enable	(Valid only in asynchronous mode) • When transmitting 0: Parity bit addition is not performed 1: The parity bit is added • When receiving 0: Parity bit checking is not performed 1: The parity bit is checked	R/W*4
b6	CHR	Character Length	(Valid only in asynchronous mode*2) Selects in combination with the SCMR.CHR1 bit. CHR1 CHR 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length*3	R/W*4
b7	CM	Communications Mode	0: Asynchronous mode or simple I2C mode 1: Clock synchronous mode or simple SPI mode	R/W*4

Note 1. n is the decimal notation of the value of n in the BRR register (refer to section 35.2.13, Bit Rate Register (BRR)).
Note 2. In other than asynchronous mode, this bit setting is invalid and a fixed data length of 8 bits is used.
Note 3. LSB first is fixed and the MSB (bit 7) in the TDR register is not transmitted in transmission.
Note 4. Writable only when the SCR.TE bit is 0 and the SCR.RE bit is 0 (both serial transmission and reception are disabled).

CKS[1:0] Bits (Clock Select)

These bits select the clock source for the on-chip baud rate generator.

For the relation between the settings of these bits and the baud rate, refer to section 35.2.13, Bit Rate Register (BRR).

MP Bit (Multi-Processor Mode)

Disables/enables the multi-processor communications function. The settings of the PE bit and PM bit are invalid in multi-processor mode.

STOP Bit (Stop Bit Length)

Selects the stop bit length in transmission.
In reception, only the first stop bit is checked regardless of this bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.

PM Bit (Parity Mode)

Selects the parity mode (even or odd) for transmission and reception.
The setting of the PM bit is invalid in multi-processor mode.

PE Bit (Parity Enable)

When this bit is set to 1, the parity bit is added to transmit data, and the parity bit is checked in reception.
Irrespective of the setting of the PE bit, the parity bit is not added or checked in multi-processor format.

CHR Bit (Character Length)

Selects the data length for transmission and reception.
Selects in combination with the SCMR.CHR1 bit.
In other than asynchronous mode, a fixed data length of 8 bits is used.

(2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SMCI0.SMR 0008 A000h, SMCI1.SMR 0008 A020h, SMCI2.SMR 0008 A040h, SMCI3.SMR 0008 A060h, SMCI4.SMR 0008 A080h, SMCI5.SMR 0008 A0A0h, SMCI6.SMR 0008 A0C0h, SMCI7.SMR 0008 A0E0h, SMCI8.SMR 0008 A100h, SMCI9.SMR 0008 A120h, SMCI10.SMR 000D 0040h, SMCI11.SMR 000D 0060h, SMCI12.SMR 0008 B300h

	b7	b6	b5	b4	b3	b2	b1	b0
	GM	BLK	PE	PM	BCP[1:0]		CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK (n = 0)*1 0 1: PCLK/4 (n = 1)*1 1 0: PCLK/16 (n = 2)*1 1 1: PCLK/64 (n = 3)*1	R/W*2
b3, b2	BCP[1:0]	Base Clock Pulse	Selects the number of base clock cycles in combination with the SCMR.BCP2 bit. Table 35.9 lists the combinations of the SCMR.BCP2 bit and SMR.BCP[1:0] bits.	R/W*2
b4	PM	Parity Mode	(Valid only when the PE bit is 1) 0: Selects even parity 1: Selects odd parity	R/W*2
b5	PE	Parity Enable	When this bit is set to 1, a parity bit is added to transmit data, and the parity of received data is checked. Set this bit to 1 in smart card interface mode.	R/W*2
b6	BLK	Block Transfer Mode	0: Non-block transfer mode operation 1: Block transfer mode operation	R/W*2
b7	GM	GSM Mode	0: Non-GSM mode operation 1: GSM mode operation	R/W*2

Note 1. n is the decimal notation of the value of n in the BRR register (refer to section 35.2.13, Bit Rate Register (BRR)).
Note 2. Writable only when the SCR.TE bit is 0 and the SCR.RE bit is 0 (both serial transmission and reception are disabled).

CKS[1:0] Bits (Clock Select)

These bits select the clock source for the on-chip baud rate generator.

For the relationship between the settings of these bits and the baud rate, refer to section 35.2.13, Bit Rate Register (BRR).

BCP[1:0] Bits (Base Clock Pulse)

These bits select the number of base clock cycles in a 1-bit data transfer time in smart card interface mode.

Set these bits in combination with the SCMR.BCP2 bit.

For details, refer to section 35.6.4, Receive Data Sampling Timing and Reception Margin.

Table 35.9 Combinations of the SCMR.BCP2 Bit and SMR.BCP[1:0] Bits

SCMR.BCP2 Bit	SMR.BCP[1:0] Bits		Number of Base Clock Cycles for 1-Bit Transfer Period
0	0	0	93 clock cycles (S = 93)*1
0	0	1	128 clock cycles (S = 128)*1
0	1	0	186 clock cycles (S = 186)*1
0	1	1	512 clock cycles (S = 512)*1
1	0	0	32 clock cycles (S = 32)*1 (Initial Value)
1	0	1	64 clock cycles (S = 64)*1
1	1	0	372 clock cycles (S = 372)*1
1	1	1	256 clock cycles (S = 256)*1

Note 1. S is the value of S in the BRR register (refer to section 35.2.13, Bit Rate Register (BRR)).

PM Bit (Parity Mode)

Selects the parity mode for transmission and reception (even or odd).
For details on the usage of this bit in smart card interface mode, refer to section 35.6.2, Data Format (Except in Block Transfer Mode).

PE Bit (Parity Enable)

Set the PE bit to 1.
The parity bit is added to transmit data before transmission, and the parity bit is checked in reception.

BLK Bit (Block Transfer Mode)

Setting this bit to 1 allows block transfer mode operation.
For details, refer to section 35.6.3, Block Transfer Mode.

GM Bit (GSM Mode)

Setting this bit to 1 allows GSM mode operation.
In GSM mode, the SSR.TEND flag set timing is put forward to 11.0 etu (elementary time unit = 1-bit transfer time) from the start and the clock output control function is appended. For details, refer to section 35.6.6, Serial Data Transmission (Except in Block Transfer Mode) and section 35.6.8, Clock Output Control.

35.2.10 Serial Control Register (SCR)

Some bits in the SCR register have different functions in smart card interface mode and non-smart card interface mode.

(1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI0.SCR 0008 A002h, SCI1.SCR 0008 A022h, SCI2.SCR 0008 A042h, SCI3.SCR 0008 A062h, SCI4.SCR 0008 A082h, SCI5.SCR 0008 A0A2h, SCI6.SCR 0008 A0C2h, SCI7.SCR 0008 A0E2h, SCI8.SCR 0008 A102h, SCI9.SCR 0008 A122h, SCI10.SCR 000D 0042h, SCI11.SCR 000D 0062h, 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	(Asynchronous mode) b1 b0 0 0: On-chip baud rate generator The SCKn pin becomes high-impedance. 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*2 • 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 SCKn pin becomes high-impedance when the TMR clock*2 is used. (Clock synchronous mode) b1 b0 0 x: Internal clock The SCKn pin functions as the clock output pin. 1 x: External clock The SCKn pin functions as the clock input pin.	R/W*1
b2	TEIE	Transmit End Interrupt Enable	0: A TEI interrupt request is disabled 1: A TEI interrupt request is enabled	R/W
b3	MPIE	Multi-Processor Interrupt Enable	(Valid in asynchronous mode when the SMR.MP bit is 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 RDRF, ORER, and FER in the SSR register to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and normal reception is resumed.	R/W
b4	RE	Receive Enable	0: Serial reception is disabled 1: Serial reception is enabled	R/W*3
b5	TE	Transmit Enable	0: Serial transmission is disabled 1: Serial transmission is enabled	R/W*3
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. TMR clock is selectable for SCI5, SCI6, and SCI12.

Note 3. 1 can be written only when TE = 0 and RE = 0, while the SMR.CM bit is 1. After setting TE or RE to 1, only 0 can be written to TE and RE. While the SMR.CM bit is 0 and the SIMR1.IICM bit is 0, writing is enabled under any condition.

CKE[1:0] Bits (Clock Enable)

These bits select the clock source and SCKn pin function.
The combination of the settings of these bits and of the SEMR.ACS0 bit sets the internal TMR clock.

TEIE Bit (Transmit End Interrupt Enable)

Enables or disables a TEI interrupt request.
A TEI interrupt request is disabled by setting the TEIE bit to 0.
In simple I^2C mode, the TEI is allocated to the interrupt on completion of issuing a start, restart, or stop condition (STI). In this case, the TEIE bit can be used to enable or disable the STI.

MPIE Bit (Multi-Processor Interrupt Enable)

When this bit is set to 1 and the data with the multi-processor bit set to 0 is received, the data is not read and setting the status flags RDRF, ORER, and FER in the SSR register (and SSRFIFO.DR flag for SCI10 and SCI11) to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and normal reception is resumed. For details, refer to section 35.4, Multi-Processor Communications Function.
When the data with the multi-processor bit set to 0 is received, the receive data is not transferred from the RSR register to the RDR register, a receive error is not detected, and setting the flags RDRF, ORER, and FER (and SSRFIFO.DR flag for SCI10 and SCI11) to 1 is disabled.
When the data with the multi-processor bit set to 1 is received, the MPB bit is set to 1, the MPIE bit is automatically cleared to 0, the RXI and ERI interrupt requests are enabled (if the SCR.RIE bit is set to 1), and setting the flags RDRF, ORER, and FER (and SSRFIFO.DR flag for SCI10 and SCI11) to 1 is enabled.
Set the MPIE bit to 0 if multi-processor communications function is not to be used.

RE Bit (Receive Enable)

Enables or disables serial reception.
When this bit is set to 1, serial reception is started by detecting the start bit in asynchronous mode or the synchronous clock input in clock synchronous mode. Note that the SMR register should be set prior to setting the RE bit to 1 in order to designate the reception format.
Even if reception is halted by setting the RE bit to 0 while the FCR.FM bit is 0 (non-FIFO mode), the ORER, FER, PER, and RDRF flags in the SSR register are not affected and the previous value is retained.
Even if reception is halted by setting the RE bit to 0 while the FCR.FM bit is 1 (FIFO mode), the RDF, ORER, FER, PER, DR flags in the SSRFIFO register are not affected and the previous value is retained.

TE Bit (Transmit Enable)

Enables or disables serial transmission.
When this bit is set to 1, serial transmission is started by writing transmit data to the TDR register. Note that the SMR register should be set prior to setting the TE bit to 1 in order to designate the transmission format.

RIE Bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.
An RXI interrupt request is disabled by setting the RIE bit to 0.
An ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in the SSR register and then setting the flag to 0, or setting the RIE bit to 0.

TIE Bit (Transmit Interrupt Enable)

Enables or disables TXI interrupt request.
A TXI interrupt request is disabled by setting the TIE bit to 0.

(2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SMCI0.SCR 0008 A002h, SMCI1.SCR 0008 A022h, SMCI2.SCR 0008 A042h, SMCI3.SCR 0008 A062h, SMCI4.SCR 0008 A082h, SMCI5.SCR 0008 A0A2h, SMCI6.SCR 0008 A0C2h, SMCI7.SCR 0008 A0E2h, SMCI8.SCR 0008 A102h, SMCI9.SCR 0008 A122h, SMCI10.SCR 000D 0042h, SMCI11.SCR 000D 0062h, SMCI12.SCR 0008 B302h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKE[1:0]	Clock Enable	• When SMR.GM = 0 b1 b0 0 0: Output disabled The SCKn pin becomes high-impedance. 0 1: Clock output 1 x: Setting prohibited • When SMR.GM = 1 b1 b0 0 0: Output fixed low x 1: Clock output 1 0: Output fixed high	R/W*1
b2	TEIE	Transmit End Interrupt Enable	This bit should be 0 in smart card interface mode.	R/W
b3	MPIE	Multi-Processor Interrupt Enable	This bit should be 0 in smart card interface mode.	R/W
b4	RE	Receive Enable	0: Serial reception is disabled 1: Serial reception is enabled	R/W*2
b5	TE	Transmit Enable	0: Serial transmission is disabled 1: Serial transmission is enabled	R/W*2
b6	RIE	Receive Interrupt Enable	0: RXI and ERI interrupt requests are disabled 1: RXI and ERI interrupt requests are enabled	R/W
b7	TIE	Transmit Interrupt Enable	0: A TXI interrupt request is disabled 1: A TXI interrupt request is enabled	R/W

x: Don't care
Note 1. Writable only when TE = 0 and RE = 0.
Note 2. 1 can be written only when TE = 0 and RE = 0. After setting TE or RE to 1, only 0 can be written in TE and RE.

For details on interrupt requests, refer to section 35.12, Interrupt Sources.

CKE[1:0] Bits (Clock Enable)

These bits control the clock output from the SCKn pin.
In GSM mode, clock output can be dynamically switched. For details, refer to section 35.6.8, Clock Output Control.

TEIE Bit (Transmit End Interrupt Enable)

This bit should be 0 in smart card interface mode.

MPIE Bit (Multi-Processor Interrupt Enable)

This bit should be 0 in smart card interface mode.

RE Bit (Receive Enable)

Enables or disables serial reception.

When this bit is set to 1, serial reception is started by detecting the start bit. Note that the SMR register should be set prior to setting the RE bit to 1 in order to designate the reception format.

Even if reception is halted by setting the RE bit to 0, the ORER, FER, and PER flags in the SSR register are not affected and the previous value is retained.

TE Bit (Transmit Enable)

Enables or disables serial transmission.

When this bit is set to 1, serial transmission is started by writing transmit data to the TDR register. Note that the SMR register should be set prior to setting the TE bit to 1 in order to designate the transmission format.

RIE Bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.

An RXI interrupt request is disabled by setting the RIE bit to 0.

An ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in the SSR register and then setting the flag to 0, or setting the RIE bit to 0.

TIE Bit (Transmit Interrupt Enable)

Enables or disables TXI interrupt request.

A TXI interrupt request is disabled by setting the TIE bit to 0.

35.2.11 Serial Status Register (SSR/SSRFIFO)

Some bits in the SSR register have different functions in smart card interface mode and non-smart card interface mode or FIFO mode and non-FIFO mode.

(1) Non-Smart Card Interface Mode and non-FIFO Mode (SCMR.SMIF = 0 and FCR.FM = 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 000D 0044h, SCI11.SSR 000D 0064h, SCI12.SSR 0008 B304h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Value after reset:	1	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MPBT	Multi-Processor Bit Transfer	Sets the multi-processor bit for adding to the transmission frame 0: Data transmission cycles 1: ID transmission cycles	R/W
b1	MPB	Multi-Processor	Value of the multi-processor bit in the reception frame 0: Data transmission cycles 1: ID transmission cycles	R
b2	TEND	Transmit End Flag	0: A character is being transmitted. 1: Character transfer has been completed.	R
b3	PER	Parity Error Flag	0: No parity error occurred 1: A parity error has occurred	R/(W) *1
b4	FER	Framing Error Flag	0: No framing error occurred 1: A framing error has occurred	R/(W) *1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error has occurred	R/(W) *1
b6	RDRF	Receive Data Full Flag	0: No valid data is held in the RDR register 1: Received data is held in the RDR register	R/(W) *2
b7	TDRE	Transmit Data Empty Flag	0: Data to be transmitted is held in the TDR register 1: No data is held in the TDR register	R/(W) *2

Note 1. Only 0 can be written to this bit, to clear the flag. To clear this flag, confirm that the flag is 1 and then set it to 0.
Note 2. Write 1 when writing is necessary.

MPB Bit (Multi-Processor)

Holds the value of the multi-processor bit in the reception frame. This bit does not change when the SCR.RE bit is 0.

TEND Flag (Transmit End Flag)

Indicates completion of transmission.

[Setting conditions]

- When the SCR.TE bit is set to 0 (serial transmission is disabled)
  When the SCR.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When the TDR register is not updated at the time of transmission of the tail-end bit of a character being transmitted

[Clearing condition]

- When transmit data are written to the TDR register while the SCR.TE bit is 1
  When setting the TEND flag to 0 to complete the interrupt handling, refer to section 15.5.2, Level Detection.

PER Flag (Parity Error Flag)

Indicates that a parity error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When a parity error is detected during reception while data match detection is disabled (for SCI0 to SCI11)
- When a parity error is detected during reception (for SCI12)
  Although receive data when the parity error occurs is transferred to the RDR register, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to the RDR register.

[Clearing condition]

- When 0 is written to the PER flag after reading PER = 1
  When setting the PER flag to 0 to complete the interrupt handling, refer to section 15.5.2, Level Detection.
  Even when the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

FER Flag (Framing Error Flag)

Indicates that a framing error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When the stop bit is 0 while data match detection is disabled (for SCI0 to SCI11)
- When the stop bit is 0 (for SCI12)
  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 the RDR register, no RXI interrupt request occurs. In addition, when the FER flag is being set to 1, the subsequent receive data is not transferred to the RDR register.

[Clearing condition]

- When 0 is written to the FER flag after reading FER = 1
  When setting the FER flag to 0 to complete the interrupt handling, refer to section 15.5.2, Level Detection.
  Even when the SCR.RE bit is set to 0, the FER flag is not affected and retains its previous value.

ORER Flag (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data is read from the RDR register
  In the RDR register, receive data prior to an overrun error occurrence is retained, but data received after the overrun error occurrence is lost. When the ORER flag is set to 1, subsequent serial reception cannot be performed. Note that, in clock synchronous mode, serial transmission also cannot continue.

[Clearing condition]

- When 0 is written to the ORER flag after reading ORER = 1
  When setting the ORER flag to 0 to complete the interrupt handling, refer to section 15.5.2, Level Detection.
  Even when the SCR.RE bit is set to 0, the ORER flag is not affected and retains its previous value.

RDRF Flag (Receive Data Full Flag)

Indicates whether the RDR register has received data.

[Setting condition]

- When data has been received normally, and transferred from the RSR register to the RDR register

[Clearing condition]

- When data is read from the RDR register

TDRE Flag (Transmit Data Empty Flag)

Indicates whether the TDR register has data to be transmitted.

[Setting condition]

- When data is transferred from the TDR register to the TSR register

[Clearing condition]

- When data is written to the TDR register

(2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SMCI0.SSR 0008 A004h, SMCI1.SSR 0008 A024h, SMCI2.SSR 0008 A044h, SMCI3.SSR 0008 A064h, SMCI4.SSR 0008 A084h, SMCI5.SSR 0008 A0A4h, SMCI6.SSR 0008 A0C4h, SMCI7.SSR 0008 A0E4h, SMCI8.SSR 0008 A104h, SMCI9.SSR 0008 A124h, SMCI10.SSR 000D 0044h, SMCI11.SSR 000D 0064h, SMCI12.SSR 0008 B304h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
Value after reset:	1	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MPBT	Multi-Processor Bit Transfer	This bit should be set to 0 in smart card interface mode.	R/W
b1	MPB	Multi-Processor	This bit is not used in smart card interface mode. It should be set to 0.	R
b2	TEND	Transmit End Flag	0: A character is being transmitted. 1: Character transfer has been completed.	R
b3	PER	Parity Error Flag	0: No parity error occurred 1: A parity error has occurred	R/(W) *1
b4	ERS	Error Signal Status Flag	0: Low error signal not responded 1: Low error signal responded	R/(W) *1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error has occurred	R/(W) *1
b6	RDRF	Receive Data Full Flag	0: No valid data is held in the RDR register 1: Received data is held in the RDR register	R/(W) *2
b7	TDRE	Transmit Data Empty Flag	0: Data to be transmitted is held in the TDR register 1: No data is held in the TDR register	R/(W) *2

Note 1. Only 0 can be written to this bit, to clear the flag. To clear this flag, confirm that the flag is 1 and then set it to 0.
Note 2. Write 1 when writing is necessary.

TEND Flag (Transmit End Flag)

With no error signal from the receiving side, this bit is set to 1 when further data for transfer is ready to be transferred to the TDR register.

[Setting conditions]

- When the SCR.TE bit is 0 (serial transmission is disabled)
  When the SCR.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When a specified period has elapsed after the latest transmission of 1 byte, the ERS flag is 0, and the TDR register is not updated
  The set timing is determined by register settings as listed below.
  When SMR.GM = 0 and SMR.BLK = 0, 12.5 etu after the start of transmission
  When SMR.GM = 0 and SMR.BLK = 1, 11.5 etu after the start of transmission
  When SMR.GM = 1 and SMR.BLK = 0, 11.0 etu after the start of transmission
  When SMR.GM = 1 and SMR.BLK = 1, 11.0 etu after the start of transmission

[Clearing condition]

- When transmit data are written to the TDR register while the SCR.TE bit is 1
  When setting the TEND flag to 0 to complete the interrupt handling, refer to section 15.5.2, Level Detection.

PER Flag (Parity Error Flag)

Indicates that a parity error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When a parity error is detected during reception
  Although receive data when the parity error occurs is transferred to the RDR register, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to the RDR register.

[Clearing condition]

- When 0 is written to the PER flag after reading PER = 1
  When setting the PER flag to 0 to complete the interrupt handling, refer to section 15.5.2, Level Detection.
  Even when the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

ERS Flag (Error Signal Status Flag)

[Setting condition]

- When a low error signal is sampled

[Clearing condition]

- When 0 is written to the ERS flag after reading ERS = 1
  When setting the ERS flag to 0 to complete the interrupt handling, refer to section 15.5.2, Level Detection.
  Even when the SCR.RE bit is set to 0, the ERS flag is not affected and retains its previous value.

ORER Flag (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data is read from the RDR register
  In the RDR register, 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 the ORER flag after reading ORER = 1
  When setting the ORER flag to 0 to complete the interrupt handling, refer to section 15.5.2, Level Detection.
  Even when the SCR.RE bit is set to 0, the ORER flag is not affected and retains its previous value.

RDRF Flag (Receive Data Full Flag)

Indicates whether the RDR register has received data.

[Setting condition]

- When data has been received normally, and transferred from the RSR register to the RDR register

[Clearing condition]

- When data is read from the RDR register

TDRE Flag (Transmit Data Empty Flag)

Indicates whether the TDR register has data to be transmitted.

[Setting condition]

- When data is transferred from the TDR register to the TSR register

[Clearing condition]

- When data is written to the TDR register

(3) Non-Smart Card Interface Mode and FIFO Mode (SCMR.SMIF = 0 and FCR.FM = 1)

Address(es): SCI10.SSRFIFO 000D 0044h, SCI11.SSRFIFO 000D 0064h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDFE	RDF	ORER	FER	PER	TEND	—	DR
Value after reset	1	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DR	Receive Data Ready Flag*1	0: Reception in progress, or receive FIFO is empty 1: Reception completed, and number of data in the receive FIFO is less than the threshold value	R/(W) *2
b1	—	Reserved	The read value is undefined. The write value should be 1	R/W
b2	TEND	Transmit End Flag	0: A character is being transmitted 1: Character transfer has been completed	R/(W) *2
b3	PER	Parity Error Flag	0: No parity error occurred 1: A parity error has occurred	R/(W) *2
b4	FER	Framing Error Flag	0: No framing error occurred 1: A framing error has occurred	R/(W) *2
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error has occurred	R/(W) *2
b6	RDF	Receive FIFO Full Flag	0: The number of data stored in the receive FIFO is less than the threshold value 1: The number of data stored in the receive FIFO is equal to or greater than the threshold value	R/(W) *2
b7	TDFE	Transmit FIFO Empty Flag	0: The number of data stored in the transmit FIFO is greater than the threshold value 1: The number of data stored in the transmit FIFO is equal to or less than the threshold value	R/(W) *2

Note 1. This flag is only enabled in asynchronous mode. This flag does not become 1 in clock synchronous mode.
Note 2. This flag can only be written to 0 to clear the flag. To clear this flag, confirm that the flag is 1 and then set it to 0.

DR Flag (Receive Data Ready Flag)

After receiving data, this flag indicates when 15 etus (Elementary Time Unit: time required to transmit a single bit) have elapsed since the number of data stored in the receive FIFO has dropped to less than the threshold value (FCR.RTRG[3:0]).

[Setting condition]

- When the last received data has neither framing nor parity error, and the number of data in the receive FIFO is less than the threshold value at the time that data is transferred from the RSR register, and the next data is not yet fully received even after 15 etus have elapsed since the last stop bit.

[Clearing condition]

- When 0 is written to the DR flag, after all data in the receive FIFO is read and the DR flag is confirmed to be 1.
- When the FCR.FM bit is changed from 0 to 1.

When the FCR.DRES bit is 1 (ERI interrupt), refer to section 15.5.2, Level Detection when setting the DR flag to 0 to exit from the interrupt handling routine.

TEND Flag (Transmit End Flag)

This flag indicates that transmission has completed.
[Setting condition]

- When transmitting the last bit of a transmit character, and there is no transmit data in the FTDR register.

[Clearing condition]

- When the SCR.TE bit is 1 and transmit data is written to the FTDR register.
- When the SCR.TE bit is 1, and 0 is written to the TEND flag after it is confirmed to be 1.
- When the FCR.FM bit is changed from 0 to 1.

PER Flag (Parity Error Flag)

In asynchronous mode when data match detection is disabled, this flag indicates that a parity error was detected for data in the receive FIFO.
[Setting condition]

- When data match detection is disabled, and a parity error is detected in the received data.

[Clearing condition]

- When 0 is written to the PER flag after it is confirmed to be 1.

When setting the PER flag to 0 to exit from the interrupt handling routine, refer to section 15.5.2, Level Detection.
Unlike in non-FIFO mode, the receive operation continues even after a parity error is detected in the receive data.
Even when the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

FER Flag (Framing Error Flag)

In asynchronous mode when data match detection is disabled, this flag indicates that a framing error was detected for data in the receive FIFO.
[Setting condition]

- When the stop bit of the received data is 0 while data match detection is disabled.

[Clearing condition]

- When 0 is written to the FER flag after it is confirmed to be 1.

When setting the FER flag to 0 to exit from the interrupt handling routine, refer to section 15.5.2, Level Detection.
Unlike in non-FIFO mode, the receive operation continues even after a framing error is detected in the receive data.
Even when the SCR.RE bit is set to 0, the FER flag is not affected and retains its previous value.

ORER Flag (Overrun Error Flag)

This flag indicates that an overrun error has occurred during reception and the reception ends abnormally
[Setting condition]

- When the receive FIFO has 16 frames of data, and the next data reception has completed.

[Clearing condition]

- When 0 is written to the ORER flag after it is confirmed to be 1.

When setting the ORER flag to 0 to exit from the interrupt handling routine, refer to section 15.5.2, Level Detection.
Even when the SCR.RE bit is set to 0, the ORER flag is not affected and retains its previous value.

RDF Flag (Receive FIFO Full Flag)

This flag indicates when the number of data stored in the receive FIFO is equal to or greater than the threshold value (FCR.RTRG[3:0]).
[Setting condition]

- When the data stored in the receive FIFO becomes equal to or greater than the threshold value.

[Clearing condition]

- When 0 is written to the RDF flag after it is confirmed to be 1.
- When the receive FIFO contains fewer receive data than the receive FIFO threshold value on completion of DTC or DMA transfer to read receive data from the FRDR register.

When the conditions for this flag to become 0 and 1 occur simultaneously, the RDF flag initially becomes 0. At this time, if the data stored in the receive FIFO is equal to or greater than the threshold value, the RDF flag becomes 1 after 1 PCLK.

TDFE Flag (Transmit FIFO Empty Flag)

This flag indicates that data has been transferred from the transmit FIFO to the TSR register, and the number of data remaining in the transmit FIFO is equal to or less than the threshold value (FCR.TTRG[3:0]).
[Setting condition]

- When the SCR.TE bit is 0.
- When the data stored in the transmit FIFO becomes equal to or less than the threshold value.

[Clearing condition]

- When the transmit FIFO contains more transmit data than the transmit FIFO threshold value on completion of DTC or DMA transfer to write transmit data to the FTDR register.
- When 0 is written to the TDFE flag after it is confirmed to be 1.

When the SCR.TE bit is set to 0, the TDFE flag becomes 1 regardless of any other condition. In all other cases, when the conditions for this flag to become 1 and 0 occur simultaneously, the TDFE flag initially becomes 0. At this time, if the data stored in the transmit FIFO is equal to or less than the threshold value, the TDFE flag becomes 1 after 1 PCLK.
To enable use of DMA or DTC transfer, do not write 0 to the TDFE flag.

35.2.12 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 000D 0046h, SCI11.SCMR 000D 0066h, SCI12.SCMR 0008 B306h,
SMCI0.SCMR 0008 A006h, SMCI1.SCMR 0008 A026h, SMCI2.SCMR 0008 A046h, SMCI3.SCMR 0008 A066h, SMCI4.SCMR 0008 A086h, SMCI5.SCMR 0008 A0A6h, SMCI6.SCMR 0008 A0C6h, SMCI7.SCMR 0008 A0E6h, SMCI8.SCMR 0008 A106h, SMCI9.SCMR 0008 A126h, SMCI10.SCMR 000D 0046h, SMCI11.SCMR 000D 0066h, SMCI12.SCMR 0008 B306h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCP2	—	—	CHR1	SDIR	SINV	—	SMIF
Value after reset:	1	1	1	1	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	SMIF	Smart Card Interface Mode Select	0: Non-smart card interface mode (Asynchronous mode, clock synchronous mode, simple SPI mode, or simple I2C mode) 1: Smart card interface mode	R/W*1
b1	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b2	SINV	Transmitted/Received Data Invert*2, *3	0: Data bits in the TDR register are transferred to the TSR register as they are. Data bits in the RSR register are transferred to the RDR register as they are. 1: Data bits in the TDR register are transferred to the TSR register with inverting. Data bits in the RSR register are transferred to the RDR register with inverting.	R/W*1
b3	SDIR	Transmitted/Received Data Transfer Direction*2, *4	0: Transfer with LSB first 1: Transfer with MSB first	R/W*1
b4	CHR1	Character Length 1*5	Selects in combination with the SMR.CHR bit. CHR1 CHR 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length*6	R/W*1
b6, b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b7	BCP2	Base Clock Pulse 2	Selects the number of base clock cycles in combination with the SMR.BCP[1:0] bits. Table 35.10 lists the combinations of the SCMR.BCP2 bit and SMR.BCP[1:0] bits.	R/W*1

Note 1. Writable only when the SCR.TE bit is 0 and the SCR.RE bit is 0 (both serial transmission and reception are disabled).
Note 2. This bit can be used in the smart card interface mode, asynchronous mode (multi-processor mode), clock synchronous mode, and simple SPI mode.
Note 3. Set this bit to 0 if operation is to be in simple I2C mode.
Note 4. Set this bit to 1 if operation is to be in simple I2C mode.
Note 5. This bit is only valid in asynchronous mode. The setting is invalid and a fixed data length of 8 bits is used in modes other than asynchronous mode.
Note 6. LSB first should be selected and the value of MSB (b7) in the TDR register cannot be transmitted.

SMIF Bit (Smart Card Interface Mode Select)

When this bit is set to 1, smart card interface mode is selected.

When this bit is set to 0, non-smart card interface mode, i.e., asynchronous mode (including multi-processor mode), clock synchronous mode, simple SPI mode, or simple I2C mode is selected.

SINV Bit (Transmitted/Received Data Invert)

This bit is used to invert the logic level of the data bits when the data is transferred between data register and shift register. This bit does not affect the logic level of the parity bit. To invert the parity bit, invert the SMR.PM bit.

CHR1 Bit (Character Length 1)

Selects the data length of transmit/receive data.
Selects in combination with the SMR.CHR bit.
A fixed data length of 8 bits is used in modes other than asynchronous mode.

BCP2 Bit (Base Clock Pulse 2)

Selects the number of base clock cycles in a 1-bit data transfer time in smart card interface mode. Set this bit in combination with the SMR.BCP[1:0] bits.

Table 35.10 Combinations of the SCMR.BCP2 Bit and SMR.BCP[1:0] Bits

SCMR.BCP2 Bit	SMR.BCP[1:0] Bits		Number of Base Clock Cycles for 1-Bit Transfer Period
0	0	0	93 clock cycles (S = 93)*1
0	0	1	128 clock cycles (S = 128)*1
0	1	0	186 clock cycles (S = 186)*1
0	1	1	512 clock cycles (S = 512)*1
1	0	0	32 clock cycles (S = 32)*1 (Initial Value)
1	0	1	64 clock cycles (S = 64)*1
1	1	0	372 clock cycles (S = 372)*1
1	1	1	256 clock cycles (S = 256)*1

Note 1. S is the value of S in the BRR register (refer to section 35.2.13, Bit Rate Register (BRR)).

35.2.13 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 000D 0041h, SCI11.BRR 000D 0061h, SCI12.BRR 0008 B301h

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

The BRR register is an 8-bit register that adjusts the bit rate.
As each SCI channel has independent baud rate generator control, different bit rates can be set for each. Table 35.11 and Table 35.12 shows the relationship between the setting (N) in the BRR register and the bit rate (B) for normal asynchronous mode, multi-processor communication, clock synchronous mode, smart card interface mode, simple SPI mode, and simple I^2C mode.
For SCI10 and SCI11, do not set the BRR register to 00h when the SMR.CM bit is 1 (clock synchronous mode or simple SPI mode), the FCR.FM bit is 1 (FIFO mode), and the SMR.CKS[1:0] bits are 00b (PCLK).
The BRR register is writable only when the TE and RE bits in the SCR register are 0.

Table 35.11 Relationship between N Setting in the BRR Register and Bit Rate B (for SCI0 to SCI11)

Mode	SEMR Settings			BRR Setting	Error (%)
	BGDM bit	ABCS bit	ABCSE bit		
Asynchronous, multi-processor communication	0	0	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	0	1	0	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	1	0	0		
	1	1	0	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	0 or 1	0 or 1	1	$N = \frac{PCLK \times 10^6}{12 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
Clock synchronous, simple SPI				$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	
Smart card interface				$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1 \right\} \times 100$
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 on-chip baud rate generator ($0 \le N \le 255$)
PCLK: Operating frequency (MHz)
n and S: Determined by the settings of the SMR and SCMR registers as listed in Table 35.14 and Table 35.15.
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-bus standard.

Table 35.12 Relationship between N Setting in the BRR Register and Bit Rate B (for SCI12)

Mode	SEMR Settings		BRR Setting	Error (%)
	BGDM bit	ABCS bit		
Asynchronous, multi-processor communication	0	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	0	1	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	1	0		
	1	1	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
Clock synchronous, simple SPI			$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	
Smart card interface			$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1 \right\} \times 100$
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 on-chip baud rate generator ($0 \leq N \leq 255$)
PCLK: Operating frequency (MHz)
n and S: Determined by the settings of the SMR and SCMR registers as listed in Table 35.14 and Table 35.15.
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-bus standard.

Table 35.13 Calculating Widths at High and Low Level for SCL

Mode	SCL	Formula (Result in Seconds)
I^2C	High period (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 7 \times \frac{1}{PCLK \times 10^6}$
	Low period (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 8 \times \frac{1}{PCLK \times 10^6}$

Table 35.14 Clock Source Settings

SMR.CKS[1:0] Bit Setting	Clock Source	n
0 0	PCLK	0
0 1	PCLK/4	1
1 0	PCLK/16	2
1 1	PCLK/64	3

Table 35.15 Base Clock Settings in Smart Card Interface Mode

SCMR.BCP2 Bit Setting	SMR.BCP[1:0] Bit Setting	Base Clock Cycles for 1-bit Period	S
0	0 0	93 clock cycles	93
0	0 1	128 clock cycles	128
0	1 0	186 clock cycles	186
0	1 1	512 clock cycles	512
1	0 0	32 clock cycles	32
1	0 1	64 clock cycles	64
1	1 0	372 clock cycles	372
1	1 1	256 clock cycles	256

Table 35.16 lists examples of N settings in the BRR register in normal asynchronous mode. Table 35.18 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 35.21. Examples of BRR (N) settings in smart card interface mode are listed in Table 35.23. Examples of BRR (N) settings in simple I^2C mode are listed in Table 35.25. In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, refer to section 35.6.4, Receive Data Sampling Timing and Reception Margin. Table 35.19 and Table 35.22 list the maximum bit rates with external clock input.

When either the SEMR.ABCS or BGDM bit is set to 1 in asynchronous mode, the bit rate becomes twice that listed in Table 35.16. When both of those bits are set to 1, the bit rate becomes four times the listed value.

Table 35.16 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	8			9.8304			10			12			12.288		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	25	0.16	0	31	0.00	0	32	–1.36	0	38	0.16	0	39	0.00
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	–2.34	0	19	0.00
31250	0	7	0.00	0	9	–1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	—	—	—	0	7	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	14			16			17.2032			18			19.6608		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	–0.17	3	70	0.03	3	75	0.48	3	79	–0.12	3	86	0.31
150	2	181	0.16	2	207	0.16	2	223	0.00	2	233	0.16	2	255	0.00
300	2	90	0.16	2	103	0.16	2	111	0.00	2	116	0.16	2	127	0.00
600	1	181	0.16	1	207	0.16	1	223	0.00	1	233	0.16	1	255	0.00
1200	1	90	0.16	1	103	0.16	1	111	0.00	1	116	0.16	1	127	0.00
2400	0	181	0.16	0	207	0.16	0	223	0.00	0	233	0.16	0	255	0.00
4800	0	90	0.16	0	103	0.16	0	111	0.00	0	116	0.16	0	127	0.00
9600	0	45	–0.93	0	51	0.16	0	55	0.00	0	58	–0.69	0	63	0.00
19200	0	22	–0.93	0	25	0.16	0	27	0.00	0	28	1.02	0	31	0.00
31250	0	13	0.00	0	15	0.00	0	16	1.20	0	17	0.00	0	19	–1.70
38400	—	—	—	0	12	0.16	0	13	0.00	0	14	–2.34	0	15	0.00

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	20			25			30			33			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	88	–0.25	3	110	–0.02	3	132	0.13	3	145	0.33	3	177	–0.25
150	3	64	0.16	3	80	0.47	3	97	–0.35	3	106	0.39	3	129	0.16
300	2	129	0.16	2	162	–0.15	2	194	0.16	2	214	–0.07	3	64	0.16
600	2	64	0.16	2	80	0.47	2	97	–0.35	2	106	0.39	2	129	0.16
1200	1	129	0.16	1	162	–0.15	1	194	0.16	1	214	–0.07	2	64	0.16
2400	1	64	0.16	1	80	0.47	1	97	–0.35	1	106	0.39	1	129	0.16
4800	0	129	0.16	0	162	–0.15	0	194	0.16	0	214	–0.07	1	64	0.16
9600	0	64	0.16	0	80	0.47	0	97	–0.35	0	106	0.39	0	129	0.16
19200	0	32	–1.36	0	40	–0.76	0	48	–0.35	0	53	–0.54	0	64	0.16
31250	0	19	0.00	0	24	0.00	0	29	0.00	0	32	0.00	0	39	0.00
38400	0	15	1.73	0	19	1.73	0	23	1.73	0	26	–0.54	0	32	–1.36

Bit Rate (bps)	Operating Frequency PCLK (MHz)								
	50			60			120*1		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	221	–0.02						
150	3	162	–0.15	3	194	0.16			
300	3	80	0.47	3	97	–0.35	3	194	0.16
600	2	162	–0.15	3	48	–0.35	3	97	–0.35
1200	2	80	0.47	2	97	–0.35	3	48	–0.35
2400	1	162	–0.15	2	48	–0.35	2	97	–0.35
4800	1	80	0.47	1	97	–0.35	2	48	–0.35
9600	0	162	–0.15	1	48	–0.35	1	97	–0.35
19200	0	80	0.47	0	97	–0.35	1	48	–0.35
31250	0	49	0.00	0	59	0.00	0	119	0.00
38400	0	40	–0.76	0	48	–0.35	0	97	–0.35

Note: This is an example when the ABCS, ABCSE, and BGDM bits in the SEMR register are 0.
When either the ABCS bit or BGDM bit is set to 1, the bit rate doubles.
When both ABCS and BGDM bits in the SEMR register are set to 1, the bit rate increases four times.
When the ABCSE bit is set to 1, the bit rate increases 16/3 times.

Note 1. Supported by SCI10 and SCI11 only.

Table 35.17 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode) (for SCI0 to SCI11)

PCLK (MHz)	SEMR Settings					Maximum Bit Rate (bps)	PCLK (MHz)	SEMR Settings					Maximum Bit Rate (bps)
	BGDM Bit	ABCS Bit	ABCSE Bit	n	N			BGDM Bit	ABCS Bit	ABCSE Bit	n	N	
8	0	0	0	0	0	250000	19.6608	0	0	0	0	0	614400
		1	0	0	0	500000			1	0	0	0	1228800
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1000000			1	0	0	0	2457600
	0 or 1	0 or 1	1	0	0	1333333		0 or 1	0 or 1	1	0	0	3276800
9.8304	0	0	0	0	0	307200	20	0	0	0	0	0	625000
		1	0	0	0	614400			1	0	0	0	1250000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1228800			1	0	0	0	2500000
	0 or 1	0 or 1	1	0	0	1638400		0 or 1	0 or 1	1	0	0	3333333
10	0	0	0	0	0	312500	25	0	0	0	0	0	781250
		1	0	0	0	625000			1	0	0	0	1562500
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1250000			1	0	0	0	3125000
	0 or 1	0 or 1	1	0	0	1666667		0 or 1	0 or 1	1	0	0	4166667
12	0	0	0	0	0	375000	30	0	0	0	0	0	937500
		1	0	0	0	750000			1	0	0	0	1875000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1500000			1	0	0	0	3750000
	0 or 1	0 or 1	1	0	0	2000000		0 or 1	0 or 1	1	0	0	5000000
12.288	0	0	0	0	0	384000	33	0	0	0	0	0	1031250
		1	0	0	0	768000			1	0	0	0	2062500
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1536000			1	0	0	0	4125000
	0 or 1	0 or 1	1	0	0	2048000		0 or 1	0 or 1	1	0	0	5500000
14	0	0	0	0	0	437500	40	0	0	0	0	0	1250000
		1	0	0	0	875000			1	0	0	0	2500000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1750000			1	0	0	0	5000000
	0 or 1	0 or 1	1	0	0	2333333		0 or 1	0 or 1	1	0	0	6666667
16	0	0	0	0	0	500000	50	0	0	0	0	0	1562500
		1	0	0	0	1000000			1	0	0	0	3125000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	2000000			1	0	0	0	6250000
	0 or 1	0 or 1	1	0	0	2666667		0 or 1	0 or 1	1	0	0	8333333
17.2032	0	0	0	0	0	537600	60	0	0	0	0	0	1875000
		1	0	0	0	1075200			1	0	0	0	3750000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	2150400			1	0	0	0	7500000
	0 or 1	0 or 1	1	0	0	2867200		0 or 1	0 or 1	1	0	0	10000000
18	0	0	0	0	0	562500	120*1	0	0	0	0	0	3750000
		1	0	0	0	1125000			1	0	0	0	7500000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	2250000			1	0	0	0	15000000
	0 or 1	0 or 1	1	0	0	3000000		0 or 1	0 or 1	1	0	0	20000000

Note 1. Supported by SCI10 and SCI11 only.

Table 35.18 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode) (for SCI12)

	SEMR Settings						SEMR Settings				
PCLK (MHz)	BGDM Bit	ABCS Bit	n	N	Maximum Bit Rate (bps)	PCLK (MHz)	BGDM Bit	ABCS Bit	n	N	Maximum Bit Rate (bps)
8	0	0	0	0	250000	19.6608	0	0	0	0	614400
		1	0	0	500000			1	0	0	1228800
	1	0	0	0			1	0	0	0	
		1	0	0	1000000			1	0	0	2457600
9.8304	0	0	0	0	307200	20	0	0	0	0	625000
		1	0	0	614400			1	0	0	1250000
	1	0	0	0			1	0	0	0	
		1	0	0	1228800			1	0	0	2500000
10	0	0	0	0	312500	25	0	0	0	0	781250
		1	0	0	625000			1	0	0	1562500
	1	0	0	0			1	0	0	0	
		1	0	0	1250000			1	0	0	3125000
12	0	0	0	0	375000	30	0	0	0	0	937500
		1	0	0	750000			1	0	0	1875000
	1	0	0	0			1	0	0	0	
		1	0	0	1500000			1	0	0	3750000
12.288	0	0	0	0	384000	33	0	0	0	0	1031250
		1	0	0	768000			1	0	0	2062500
	1	0	0	0			1	0	0	0	
		1	0	0	1536000			1	0	0	4125000
14	0	0	0	0	437500	40	0	0	0	0	1250000
		1	0	0	875000			1	0	0	2500000
	1	0	0	0			1	0	0	0	
		1	0	0	1750000			1	0	0	5000000
16	0	0	0	0	500000	50	0	0	0	0	1562500
		1	0	0	1000000			1	0	0	3125000
	1	0	0	0			1	0	0	0	
		1	0	0	2000000			1	0	0	6250000
17.2032	0	0	0	0	537600	60	0	0	0	0	1875000
		1	0	0	1075200			1	0	0	3750000
	1	0	0	0			1	0	0	0	
		1	0	0	2150400			1	0	0	7500000
18	0	0	0	0	562500						
		1	0	0	1125000						
	1	0	0	0							
		1	0	0	2250000						

Table 35.19 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

		Maximum Bit Rate (bps)	
PCLK (MHz)	External Input Clock (MHz)	SEMR.ABCS Bit = 0	SEMR.ABCS Bit = 1
8	2.0000	125000	250000
9.8304	2.4576	153600	307200
10	2.5000	156250	312500
12	3.0000	187500	375000
12.288	3.0720	192000	384000
14	3.5000	218750	437500
16	4.0000	250000	500000
17.2032	4.3008	268800	537600
18	4.5000	281250	562500
19.6608	4.9152	307200	614400
20	5.0000	312500	625000
25	6.2500	390625	781250
30	7.5000	468750	937500
33	8.2500	515625	1031250
40	10.0000	625000	1250000
50	12.5000	781250	1562500
60	15.0000	937500	1875000
120*1	30.0000	1875000	3750000

Note 1. Supported by SCI10 and SCI11 only.

Table 35.20 Maximum Bit Rate with TMR Clock Input (Asynchronous Mode)

		Maximum Bit Rate (bps)	
PCLK (MHz)	TMR Clock (MHz)	SEMR.ABCS Bit = 0	SEMR.ABCS Bit = 1
8	4	250000	500000
9.8304	4.9152	307200	614400
10	5	312500	625000
12	6	375000	750000
12.288	6.144	384000	768000
14	7	437500	875000
16	8	500000	1000000
17.2032	8.6016	537600	1075200
18	9	562500	1125000
19.6608	9.8304	614400	1228800
20	10	625000	1250000
25	12.5	781250	1562500
30	15	937500	1875000
33	16.5	1031250	2062500
40	20	1250000	2500000
50	25	1562500	3125000
60	30	1875000	3750000

Table 35.21 BRR Settings for Various Bit Rates (Clock Synchronous Mode, Simple SPI Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)																					
	8		10		16		20		25		30		33		40		50		60		120*1	
	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N
110																						
250	3	124	3	155	3	249																
500	2	249	3	77	3	124	3	155	3	194	3	233	3	255								
1 k	2	124	2	155	2	249	3	77	3	97	3	116	3	128	3	155	3	194	3	233		
2.5 k	1	199	1	249	2	99	2	124	2	155	2	187	2	205	2	249	3	77	3	93	3	187
5 k	1	99	1	124	1	199	1	249	2	77	2	93	2	102	2	124	2	155	3	46	3	93
10 k	0	199	0	249	1	99	1	124	1	155	1	187	1	205	1	249	2	77	2	93	3	46
25 k	0	79	0	99	0	159	0	199	0	249	1	74	1	82	1	99	1	124	1	149	2	74
50 k	0	39	0	49	0	79	0	99	0	124	0	149	0	164	1	49	1	61	1	74	1	149
100 k	0	19	0	24	0	39	0	49	0	62	0	74	0	82	0	99	0	124	0	149	1	74
250 k	0	7	0	9	0	15	0	19	0	24	0	29	0	32	0	39	0	49	0	59	1	29
500 k	0	3	0	4	0	7	0	9	—	—	0	14	—	—	0	19	0	24	0	29	1	14
1 M	0	1			0	3	0	4	—	—			—	—	0	9	—	—	0	14	0	29
2.5 M			0	0*2			0	1			0	2			0	3	0	4	0	5	0	11
5 M							0	0*2							0	1			0	2	0	5
7.5 M											0	0*2							0	1	0	3

Blank cell: Cannot be set since the bit rate error exceeds 5%.

—: Can be set, but a bit rate error of 1 to 5% will occur.

Note 1. Supported by SCI10 and SCI11 only.

Note 2. For SCI10 and SCI11, this setting cannot be used when the FCR.FM bit is 1 (FIFO mode).
When this setting is used while the FCR.FM bit is 0 (non-FIFO mode) or with other channel, continuous transmission or reception is not possible. After transmitting/receiving one frame of data, there is an interval of a 1-bit period before starting transmitting/receiving the next frame of data. The output of the synchronization clock is stopped for a 1-bit period. For this reason, it takes 9 bits worth of time to transfer one frame (8 bits) of data, and the average transfer rate is 8/9 times the bit rate.

Table 35.22 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode, Simple SPI Mode)

PCLK (MHz)	External Input Clock (MHz)	Maximum Bit Rate (Mbps)
8	1.3333	1.3333
10	1.6667	1.6667
12	2.0000	2.0000
14	2.3333	2.3333
16	2.6667	2.6667
18	3.0000	3.0000
20	3.3333	3.3333
25	4.1667	4.1667
30	5.0000	5.0000
33	5.5000	5.5000
40	6.6667	6.6667
50	8.3333	8.3333
60	10.0000	10.0000
120*1	20.0000	20.0000

Note 1. Supported by SCI10 and SCI11 only.

Table 35.23 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	–30.00
	10.7136	0	1	–25.00
	13.00	0	1	–8.99
	14.2848	0	1	0.00
	16.00	0	1	12.01
	18.00	0	2	–15.99
	20.00	0	2	–6.66
	25.00	0	3	–12.49
	30.00	0	3	5.01
	33.00	0	4	–7.59
	40.00	0	5	–6.66
	50.00	0	6	0.01
	60.00	0	7	5.01
	120.00*1	0	16	–1.17

Note 1. Supported by SCI10 and SCI11 only.

Table 35.24 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 32)

PCLK (MHz)	Maximum Bit Rate (bps)	n	N
10.00	156250	0	0
10.7136	167400	0	0
13.00	203125	0	0
16.00	250000	0	0
18.00	281250	0	0
20.00	312500	0	0
25.00	390625	0	0
30.00	468750	0	0
33.00	515625	0	0
40.00	625000	0	0
50.00	781250	0	0
60.00	937500	0	0
120.00*1	1875000	0	0

Note 1. Supported by SCI10 and SCI11 only.

Table 35.25 BRR Settings for Various Bit Rates (Simple I²C Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	8			10			16			20			25		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10 k	0	24	0.0	0	31	–2.3	1	12	–3.8	1	15	–2.3	1	19	–2.3
25 k	0	9	0.0	0	12	–3.8	1	4	0.0	1	6	–10.7	1	7	–2.3
50 k	0	4	0.0	0	6	–10.7	1	2	–16.7	1	3	–21.9	1	3	–2.3
100 k	0	2	–16.7	0	3	–21.9	0	4	0.0	0	6	–10.7	1	1	–2.3
250 k	0	0	0.0	0	1	–37.5	0	1	0.0	0	2	–16.7	0	3	–21.9
350 k										0	1	–10.7	0	2	–25.6

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	30			33			40			50			60		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10 k	1	23	–2.3	1	25	–0.8	0	124	0.00	2	9	–2.3	1	46	–0.27
25 k	1	9	–6.3	1	10	–6.3	0	40	0.00	2	3	–2.3	0	74	0.00
50 k	1	4	–6.3	1	5	–14.1	0	24	0.00	2	1	–2.3	0	37	–1.32
100 k	1	2	–21.9	1	2	–14.1	0	12	–3.85	1	3	–2.3	0	18	–1.32
250 k	0	3	–6.3	0	4	–17.5	0	4	0.00	0	6	–10.7	0	7	–6.25
350 k	0	2	–10.7	0	2	–1.8	0	3	–10.71	0	4	–10.7	0	4	7.14

Bit Rate (bps)	Operating Frequency PCLK (MHz)		
	120*1		
	n	N	Error (%)
10 k	1	93	–0.27
25 k	0	149	0.00
50 k	0	74	0.00
100 k	0	37	–1.31
250 k	0	14	0.00
350 k	0	10	–2.60

Note 1. Supported by SCI10 and SCI11 only.

Table 35.26 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple I²C Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	8			10			16			20		
	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)
10 k	0	24	43.75/50.00	0	31	44.80/51.20	1	12	45.50/52.00	1	15	44.80/51.20
25 k	0	9	17.50/20.00	0	12	18.20/20.80	1	4	17.50/20.00	1	6	19.60/22.40
50 k	0	4	8.75/10.00	0	6	9.80/11.20	1	2	10.50/12.00	1	3	11.20/12.80
100 k	0	2	5.25/6.00	0	3	5.60/6.40	0	4	4.37/5.00	0	6	4.90/5.60
250 k	0	0	1.75/2.00	0	1	2.80/3.20	0	1	1.75/2.00	0	2	2.10/2.40
350 k										0	1	1.40/1.60

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	25			30			33			40		
	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)
10 k	1	19	44.80/51.20	1	23	44.80/51.20	1	25	44.12/50.42	1	32	46.20/52.80
25 k	1	7	17.92/20.48	1	9	18.66/21.33	1	10	18.66/21.33	1	12	18.20/20.80
50 k	1	3	8.96/10.24	1	4	9.33/10.66	1	5	10.18/11.63	1	6	9.80/11.20
100 k	1	1	4.48/5.12	1	2	5.60/6.40	1	2	5.09/5.81	0	13	4.90/5.60
250 k	0	3	2.24/2.56	0	3	1.86/2.13	0	4	2.12/2.42	0	4	1.75/2.00
350 k	0	2	1.68/1.92	0	2	1.40/1.60	0	2	1.27/1.45	0	3	1.40/1.60

Bit Rate (bps)	Operating Frequency PCLK (MHz)								
	50			60			120*1		
	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)
10 k	2	9	44.80/51.20	1	47	44.80/51.20	1	93	43.87/50.13
25 k	2	3	17.92/20.48	0	74	17.50/20.00	0	149	17.50/20.00
50 k	2	1	8.96/10.24	0	37	8.87/10.13	0	74	8.75/10.00
100 k	1	3	4.48/5.12	0	18	4.43/5.07	0	37	4.43/5.07
250 k	0	6	1.96/2.24	0	7	1.87/2.13	0	15	1.87/2.13
350 k	0	4	1.40/1.60	0	5	1.40/1.60	0	10	1.28/1.47

Note 1. Supported by SCI10 and SCI11 only.

35.2.14 Modulation Duty Register (MDDR)

Address(es): SCI0.MDDR 0008 A012h, SCI1.MDDR 0008 A032h, SCI2.MDDR 0008 A052h, SCI3.MDDR 0008 A072h, SCI4.MDDR 0008 A092h, SCI5.MDDR 0008 A0B2h, SCI6.MDDR 0008 A0D2h, SCI7.MDDR 0008 A0F2h, SCI8.MDDR 0008 A112h, SCI9.MDDR 0008 A132h, SCI10.MDDR 000D 0052h, SCI11.MDDR 000D 0072h, SCI12.MDDR 0008 B312h

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

The MDDR register corrects the bit rate adjusted by the BRR register.
When the SEMR.BRME bit is set to 1, the bit rate generated by the on-chip baud rate generator is evenly corrected according to the settings of the MDDR register (M/256). The relationship between the MDDR register setting (M) and the bit rate (B) is given in Table 35.27 and Table 35.28.
The range of the value that can be set in the MDDR register is from 80h to FFh. A value other than these cannot be set.
The MDDR register is writable only when the TE and RE bits in the SCR register are 0.

Table 35.27 Relationship between MDDR Setting (M) and Bit Rate (B) When Bit Rate Modulation Function is Used (for SCI0 to SCI11)

Mode	SEMR Settings			BRR Setting	Error (%)
	BGDM Bit	ABCS Bit	ABCSE Bit		
Asynchronous, multi-processor communication	0	0	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
	1	0	0	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
	0	1	0		
	1	1	0	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
	0 or 1	0 or 1	1	$N = \frac{PCLK \times 10^6}{12 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
Clock synchronous, simple SPI*1				$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	
Smart card interface				$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
Simple I^2C*2				$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	

B: Bit rate (bps)
M: MDDR setting (128 ≤ MDDR ≤ 256)
N: BRR setting for baud rate generator (0 ≤ N ≤ 255)
PCLK: Operating frequency (MHz)
n and S: Determined by the settings of the SMR and SCMR registers as listed in Table 35.14 and Table 35.15, section 35.2.13, Bit Rate Register (BRR).

Note 1. Do not use this function in clock synchronous mode and in the highest speed settings in simple SPI mode (SMR.CKS[1:0] = 00b, SCR.CKE[1] = 0, and BRR = 0).
Note 2. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I²C mode satisfy the I²C-bus standard.

Table 35.28 Relationship between MDDR Setting (M) and Bit Rate (B) When Bit Rate Modulation Function is Used (for SCI12)

Mode	SEMR Settings		BRR Setting	Error (%)
	BGDM Bit	ABCS Bit		
Asynchronous, multi-processor communication	0	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
	1	0	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
	0	1		
	1	1	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
Clock synchronous, simple SPI*1			$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	
Smart card interface			$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
Simple I²C*2			$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	

B: Bit rate (bps)
M: MDDR setting (128 ≤ MDDR ≤ 256)
N: BRR setting for baud rate generator (0 ≤ N ≤ 255)
PCLK: Operating frequency (MHz)
n and S: Determined by the settings of the SMR and SCMR registers as listed in Table 35.14 and Table 35.15, section 35.2.13, Bit Rate Register (BRR).

Note 1. Do not use this function in clock synchronous mode and in the highest speed settings in simple SPI mode (SMR.CKS[1:0] = 00b, SCR.CKE[1] = 0, and BRR = 0).
Note 2. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I²C mode satisfy the I²C-bus standard.

Smaller settings of the SMR.CKS[1:0] bits and larger settings of the BRR register reduce difference in the length of the 1-bit period.

35.2.15 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 000D 0047h, SCI11.SEMR 000D 0067h, SCI12.SEMR 0008 B307h

	b7	b6	b5	b4	b3	b2	b1	b0
	RXDESEL	BGDM	NFEN	ABCS	ABCSE	BRME	ITE	ACS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ACS0	Asynchronous Mode Clock Source Select	(Valid only in asynchronous mode) 0: External clock input 1: Logical AND of two compare matches output from TMR (valid for SCI5, SCI6, and SCI12 only) Available compare match output varies per SCI channel.	R/W*1
b1	ITE	Instant Transmission Enable*2	(Valid only in asynchronous mode) 0: Internal wait time is inserted after the transmission is enabled 1: Data transmission is started immediately after the transmission is enabled.	R/W*1
b2	BRME	Bit Rate Modulation Enable	0: Bit rate modulation function is disabled. 1: Bit rate modulation function is enabled.	R/W*1
b3	ABCSE	Asynchronous Mode Base Clock Select Extended*2	(Valid only when using a on-chip baud rate generator in asynchronous mode) 0: The number of clock cycles for 1-bit period depends on the BGDM and ABCS bits setting. 1: Selects 6 base clock cycles for 1-bit period.	R/W*1
b4	ABCS	Asynchronous Mode Base Clock Select	(Valid only in asynchronous mode) 0: Selects 16 base clock cycles for 1-bit period. 1: Selects 8 base clock cycles for 1-bit period.	R/W*1
b5	NFEN	Digital Noise Filter Function Enable	(In asynchronous mode) 0: Noise cancellation function for the RXDn input signal is disabled. 1: Noise cancellation function for the RXDn input signal is enabled. (in simple I^2C mode) 0: Noise cancellation function for the SSCLn and SSDAn input signals is disabled. 1: Noise cancellation function for the SSCLn and SSDAn input signals is enabled. The NFEN bit should be 0 in any mode other than above.	R/W*1
b6	BGDM	Baud Rate Generator Double-Speed Mode Select	(Only valid the SCR.CKE[1] bit is 0 in asynchronous mode) 0: Baud rate generator outputs the clock with normal frequency. 1: Baud rate generator outputs the clock with doubled frequency.	R/W*1
b7	RXDESEL	Asynchronous Start Bit Edge Detection Select	(Valid only in asynchronous mode) 0: The low level on the RXDn pin is detected as the start bit. 1: A falling edge on the RXDn pin is detected as the start bit.	R/W*1

Note 1. Writable only when the SCR.TE bit is 0 and the SCR.RE bit is 0 (both serial transmission and reception are disabled).
Note 2. This bit is reserved for SCI12. It is read as 0. The write value should be 0.

The SEMR register is used to select a clock source for 1-bit period in asynchronous mode or a detection method of the start bit.

ACS0 Bit (Asynchronous Mode Clock Source Select)

Selects the clock source in the asynchronous mode.

The ACS0 bit is valid in asynchronous mode (SMR.CM bit = 0) and when an external clock input is selected (SCR.CKE[1:0] bits = 10b or 11b). This bit is used to select an external clock input or the logical AND of compare matches output from the internal TMR.

Set the ACS0 bit to 0 in other than asynchronous mode.

For SCI5, SCI6, and SCI12, the TMOn output (n = 0 to 3) of TMR units 0 and 1 can be set as the base clock source. Refer to Table 35.29 for details.

The ACS0 bits for SCI0 to SCI4 and SCI7 to SCI11 are reserved. The write values to these bits for SCI0 to SCI4 and SCI7 to SCI11 should be 0.

Table 35.29 Correspondence between SCI Channels and Compare Match Outputs

SCI	TMR	Compare Match Output
SCI5	Unit 0	TMO0, TMO1
SCI6	Unit 1	TMO2, TMO3
SCI12	Unit 0	TMO0, TMO1

Figure 35.5 shows a setting example of when TMO0 and TMO1 in the TMR unit 0 are selected for output.

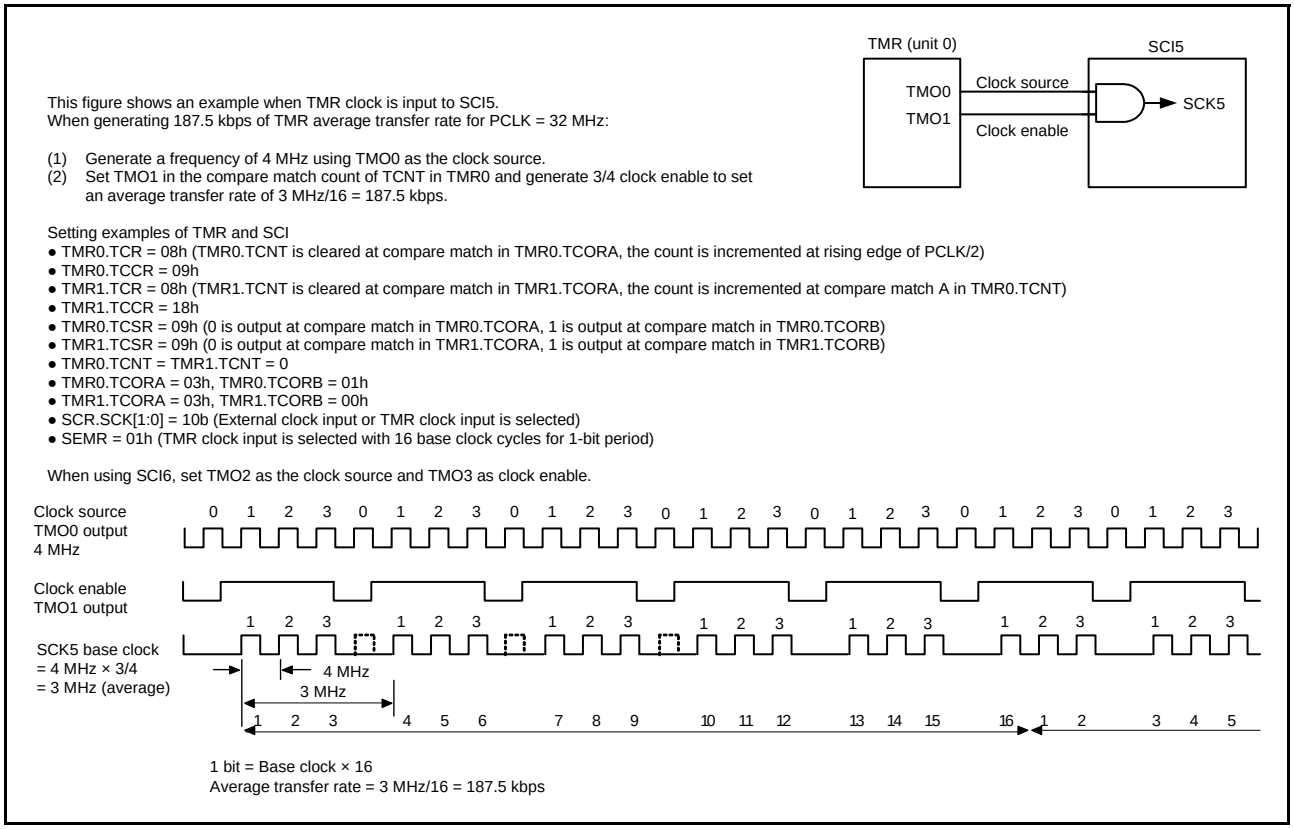


Figure 35.5 Example of Average Transfer Rate Setting When TMR Clock is Input

ITE Bit (Instant Transmission Enable)

This bit is used to start data transmission without internal wait time in asynchronous mode. When the TE bit is set to 1 while this bit is 0, one frame of internal wait time is inserted before the data transmission is started. When this bit is 1, the data transmission is started immediately after the TE bit is set to 1.

BRME Bit (Bit Rate Modulation Enable)

This bit enables and disables the bit rate modulation function. The bit rate generated by on-chip baud rate generator is evenly corrected when this function is enabled.

ABCSE Bit (Asynchronous Mode Base Clock Select Extended*2)

When setting this bit to 1, the 1-bit period becomes 6 cycles of the base clock and the clock of doubled frequency is output from the baud rate generator.
This bit is valid when the on-chip baud rate generator is selected as the clock source (SCR.CKE[1] = 0) in asynchronous mode (SMR.CM = 0).
When the bit rate is made to be 1/6 of PCLK frequency, set this bit to 1, the SMR.CKS[1:0] bits to 00b, and the BRR register to 00h.

NFEN Bit (Digital Noise Filter Function Enable)

This bit enables or disables the digital noise filter function.
When the function is enabled, noise cancellation is applied to the RXDn input signal in asynchronous mode, and noise cancellation is applied to the SSDAn and SSCLn input signals in simple I^2C mode.
In any mode other than above, set the NFEN bit to 0 to disable the digital noise filter function.
When the function is disabled, input signals are transferred as is, as internal signals.

BGDM Bit (Baud Rate Generator Double-Speed Mode Select)

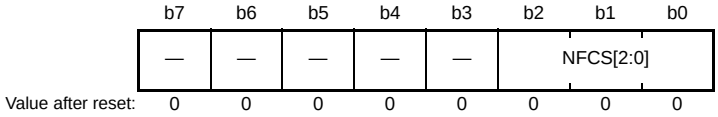
Selects the cycle of output clock for the baud rate generator.
This bit is valid when the on-chip baud rate generator is selected as the clock source (SCR.CKE[1] = 0) in asynchronous mode (SMR.CM = 0). For the clock output from the baud rate generator, either normal or doubled frequency can be selected. The base clock is generated by the clock output from the baud rate generator. When the BGDM bit is set to 1, the base clock cycle is halved and the bit rate is doubled.
Set this bit to 0 in modes other than asynchronous mode.

RXDESEL Bit (Asynchronous Start Bit Edge Detection Select)

Selects the detection method of the start bit for reception in asynchronous mode. When a break occurs, data receiving operation depends on the settings of this bit. Set this bit to 1 when reception should be stopped while a break occurs or when reception should be started without retaining the RXDn pin input at high level for the period of one data frame or longer after completion of the break.
Set this bit to 0 in modes other than asynchronous mode.

35.2.16 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 000D 0048h, SCI11.SNFR 000D 0068h, 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 1 is used with the noise filter. In simple I²C mode, the standard settings for the clock source of the on-chip baud rate generator selected by the SMR.CKS[1:0] bits are given below. b2 b0 0 0 1: The clock signal divided by 1 is used with the noise filter. 0 1 0: The clock signal divided by 2 is used with the noise filter. 0 1 1: The clock signal divided by 4 is used with the noise filter. 1 0 0: The clock signal divided by 8 is used with the noise filter. Settings other than above are prohibited.	R/W*1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR register are 0 (serial reception and transmission disabled).

NFCS[2:0] Bits (Noise Filter Clock Select)

These bits select the sampling clock for the digital noise filter. To use the noise filter in asynchronous mode, set these bits to 000b. In simple I²C mode, set the bits to a value in the range from 001b to 100b.

35.2.17 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 000D 0049h, SCI11.SIMR1 000D 0069h, SCI12.SIMR1 0008 B309h

	b7	b6	b5	b4	b3	b2	b1	b0
	IICDL[4:0]					—	—	IICM
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IICM	Simple I²C Mode Select	SMIF IICM 0 0: Asynchronous mode, Multi-processor mode, Clock synchronous mode (in asynchronous mode, synchronous, or simple SPI mode) 0 1: Simple I²C mode 1 0: Smart card interface mode 1 1: Setting prohibited.	R/W*1
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7 to b3	IICDL[4:0]	SSDA Output Delay Select	(Cycles below are of the clock signal from the on-chip baud rate generator.) b7 b3 0 0 0 0 0: No output delay 0 0 0 0 1: 0 to 1 cycle 0 0 0 1 0: 1 to 2 cycles 0 0 0 1 1: 2 to 3 cycles 0 0 1 0 0: 3 to 4 cycles 0 0 1 0 1: 4 to 5 cycles : : 1 1 1 1 0: 29 to 30 cycles 1 1 1 1 1: 30 to 31 cycles	R/W*1

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR register are 0 (both serial transmission and reception are disabled).

SIMR1 is used to select simple I²C mode and the number of delay stages for the SSDA output.

IICM Bit (Simple I²C Mode Select)

In conjunction with the SCMR.SMIF bit, this bit selects the operating mode.

IICDL[4:0] Bits (SSDA Output Delay Select)

These bits are used to set a delay for output on the SSDAn pin relative to the falling edge of the output on the SSCLn pin. The available delay settings range from no delay to 31 cycles, with the clock signal from the on-chip baud rate generator as the base. The signal obtained by frequency-dividing PCLK by the divisor set in the SMR.CKS[1:0] bits is supplied as the clock signal from the on-chip baud rate generator. Set these bits to 00000b unless operation is in simple I²C mode. In simple I²C mode, set the bits to a value in the range from 00001b to 11111b.

35.2.18 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 000D 004Ah, SCI11.SIMR2 000D 006Ah, 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 register are 0 (serial reception and transmission disabled).

SIMR2 is used to select how reception and transmission are controlled in simple I²C mode.

IICINTM Bit (I²C Interrupt Mode Select)

This bit selects the sources of interrupt requests in simple I²C mode.

IICCSC Bit (Clock Synchronization)

Set the IICCSC bit to 1 if the internally generated SSCLn clock signal is to be synchronized when the SSCLn pin has been placed at the low level in the case of a wait inserted by the other device, etc.

The SSCLn clock signal is not synchronized if the IICCSC bit is 0. The SSCLn clock signal is generated in accord with the rate selected in the BRR regardless of the level being input on the SSCLn pin.

Set the IICCSC bit to 1 except during debugging.

IICACKT Bit (ACK Transmission Data)

Transmitted data contains ACK bits. Set this bit to 1 when ACK and NACK bits are received.

35.2.19 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 000D 004Bh, SCI11.SIMR3 000D 006Bh, SCI12.SIMR3 0008 B30Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	IICSCLS[1:0]		IICSDAS[1:0]		IICSTIF	IICSTPREQ	IICRSTAREQ	IICSTAREQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IICSTAREQ	Start Condition Generation	0: A start condition is not generated. 1: A start condition is generated.*1, *3, *4, *5	R/W
b1	IICRSTAREQ	Restart Condition Generation	0: A restart condition is not generated. 1: A restart condition is generated.*2, *3, *4, *5	R/W
b2	IICSTPREQ	Stop Condition Generation	0: A stop condition is not generated. 1: A stop condition is generated.*2, *3, *4, *5	R/W
b3	IICSTIF	Issuing of Start, Restart, or Stop Condition Completed Flag	0: There are no requests for generating conditions or a condition is being generated. 1: A start, restart, or stop condition is completely generated.	R/W
b5, b4	IICSDAS[1:0]	SSDA Output Select	b5 b4 0 0: Serial data output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SSDAn pin. 1 1: Place the SSDAn pin in the high-impedance state.	R/W
b7, b6	IICSCLS[1:0]	SSCL Output Select	b7 b6 0 0: Serial clock output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SSCLn pin. 1 1: Place the SSCLn pin in the high-impedance state.	R/W

Note 1. Generate a start condition only when the SSCLn and SSDAn pins are both high (the corresponding bits in the corresponding PIDR registers are 1).
Note 2. Generate a restart or stop condition only when the SSCLn pin is low (the corresponding bit in the PIDR register is 0).
Note 3. Do not set more than one from among the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits to 1 at a given time.
Note 4. Execute the generation of a condition after the value of the IICSTIF flag is 0.
Note 5. Do not write 0 to this bit while it is 1. Generation of a condition is suspended by writing 0 to this bit while it is 1.

SIMR3 is used to control the simple I²C mode start and stop conditions, and to hold the SSDAn and SSCLn pins at fixed levels.

IICSTAREQ Bit (Start Condition Generation)

When a start condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTAREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the start condition

IICRSTAREQ Bit (Restart Condition Generation)

When a restart condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICRSTAREQ bit to 1.
[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the restart condition

IICSTPREQ Bit (Stop Condition Generation)

When a stop condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTPREQ bit to 1.
[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the stop condition

IICSTIF Flag (Issuing of Start, Restart, or Stop Condition Completed Flag)

After generating a condition, this bit indicates that the generation is completed. When using the IICSTAREQ, IICRSTAREQ, or IICSTPREQ bit to cause generation of a condition, do so after setting the IICSTIF flag to 0.
When the IICSTIF flag is 1 while an interrupt request is enabled by setting the SCR.TEIE bit, an STI request is output.
[Setting condition]

- Completion of the generation of a start, restart, or stop condition (however, in cases where this conflicts with any of the conditions for the flag becoming 0 listed below, the other condition takes precedence)

[Clearing conditions]

- Writing 0 to the bit (confirm that the IICSTIF flag is 0 before doing so)
- Writing 0 to the SIMR1.IICM bit (when operation is not in simple I^2C mode)
- Writing 0 to the SCR.TE bit

IICSDAS[1:0] Bits (SSDA Output Select)

These bits control output from the SSDAn pin.
Set the IICSDAS[1:0] and IICSCLS[1:0] bits to the same value during normal operations.

IICSCLS[1:0] Bits (SSCL Output Select)

These bits control output from the SSCLn pin.
Set the IICSCLS[1:0] and IICSDAS[1:0] bits to the same value during normal operations.

35.2.20 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 000D 004Ch, SCI11.SISR 000D 006Ch, SCI12.SISR 0008 B30Ch

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

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	IICACKR	ACK Reception Data Flag	0: ACK received 1: NACK received	R/W*1
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	—	Reserved	The read value is undefined.	R
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	—	Reserved	The read value is undefined.	R
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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

SISR is used to monitor state in relation to simple I²C mode.

IICACKR Flag (ACK Reception Data Flag)

Received ACK and NACK bits can be read from this bit.

The IICACKR flag is updated at the rising of SSCLn clock for the ACK/NACK receiving bit.

35.2.21 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 000D 004Dh, SCI11.SPMR 000D 006Dh, SCI12.SPMR 0008 B30Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	CKPH	CKPOL	—	MFF	—	MSS	CTSE	SSE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SSE	SSn# Pin Function Enable	0: SSn# pin function is disabled. 1: SSn# pin function is enabled.	R/W*1
b1	CTSE	CTS Enable	0: CTS function is disabled (RTS output function is enabled). 1: CTS function is enabled.	R/W*1
b2	MSS	Master Slave Select	0: Transmission is through the SMOSIn pin and reception is through the SMISOn pin (master mode). 1: Reception is through the SMOSIn pin and transmission is through the SMISOn pin (slave mode).	R/W*1
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	MFF	Mode Fault Flag	0: No mode fault error 1: Mode fault error	R/W*2
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	CKPOL	Clock Polarity Select	0: Clock polarity is not inverted. 1: Clock polarity is inverted.	R/W*1
b7	CKPH	Clock Phase Select	0: Clock is not delayed. 1: Clock is delayed.	R/W*1

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR register are 0 (both serial transmission and reception are disabled).
Note 2. Only 0 can be written to these bits, which clears the flag.

SPMR is used to select the extension settings in asynchronous and clock synchronous modes.

SSE Bit (SSn# Pin Function Enable)

Set this bit to 1 if the SSn# pin is to be used in control of transmission and reception (in simple SPI mode). Set this bit to 0 in any other mode. Furthermore, even for usage in simple SPI mode, the SSn# pin on the master side is not required to control reception and transmission when master mode (SCR.CKE[1:0] = 00b and MSS = 0) is selected and there is a single master, so the setting for the SSE bit is 0. Do not set both the SSE and CTSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

CTSE Bit (CTS Enable)

Set this bit to 1 if the SSn# pin is to be used for inputting of the CTS control signal to control of transmission and reception. The RTS signal is output when this bit is set to 0. Set this bit to 0 in smart card interface mode, simple SPI mode, and simple I^2C mode. Do not set both the CTSE and SSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

MSS Bit (Master Slave Select)

This bit selects between master and slave operation in simple SPI mode. When the MSS bit is set to 1, data is received through the SMOSIn pin and transmitted through the SMISOn pin.
Set this bit to 0 in modes other than simple SPI mode.

MFF Flag (Mode Fault Flag)

This bit indicates mode fault errors.
In a multi-master configuration, determine the mode fault error occurrence by reading the MFF flag.
[Setting condition]

- Input on the SSn# pin being at the low level during master operation in simple SPI mode (SSE bit = 1 and MSS bit = 0)

[Clearing condition]

- Writing 0 to the bit after it was read as 1

CKPOL Bit (Clock Polarity Select)

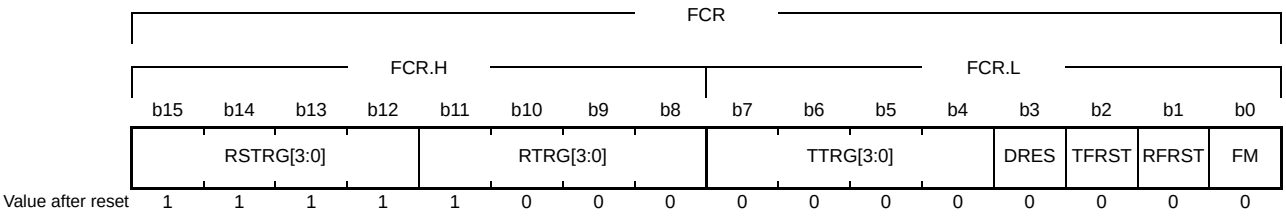
This bit selects the polarity of the clock signal output through the SCKn pin. Refer to Figure 35.64 for details.
Set the bit to 0 in other than simple SPI mode and clock synchronous mode.

CKPH Bit (Clock Phase Select)

This bit selects the phase of the clock signal output through the SCKn pin. Refer to Figure 35.64 for details.
Set the bit to 0 in other than simple SPI mode and clock synchronous mode.

35.2.22 FIFO Control Register (FCR)

Address(es): SCI10.FCR 000D 0054h, SCI11.FCR 000D 0074h, SCI10.FCR.H 000D 0054h, SCI11.FCR.H 000D 0074h, SCI10.FCR.L 000D 0055h, SCI11.FCR.L 000D 0075h



Bit	Symbol	Bit Name	Description	R/W
b0	FM	FIFO Mode Select*1	0: Non-FIFO mode (TDR and RDR registers are used for transmit and receive) 1: FIFO mode (FTDR and FRDR registers are used for transmit and receive)	R/W*2
b1	RFRST	Receive FIFO Reset*3	Writing 1 clears the number of data stored in the receive FIFO to 0	R/W
b2	TFRST	Transmit FIFO Reset*3	Writing 1 clears the number of data stored in the transmit FIFO to 0	R/W
b3	DRES	Receive Data Ready Interrupt Select	0: Receive data full interrupt (RXI) 1: Error interrupt (ERI)	R/W*2
b7 to b4	TTRG[3:0]	Transmit FIFO Threshold Setting *1	Specifies the threshold value for the SSRFIFO.TDFE flag to become 1 b7 b4 0 0 0 0: Transmit FIFO threshold value is 0 0 0 0 1: Transmit FIFO threshold value is 1 : : 1 1 1 1: Transmit FIFO threshold value is 15	R/W
b11 to b8	RTRG[3:0]	Receive FIFO Threshold Setting *1	Specifies the threshold value for the SSRFIFO.RDF flag to become 1 b11 b8 0 0 0 0: Setting prohibited 0 0 0 1: Receive FIFO threshold value is 1 : : 1 1 1 1: Receive FIFO threshold value is 15	R/W
b15 to b12	RSTRG[3:0]	RTS# Output Threshold Setting *4	Specifies the threshold value for the RTSn# pin to be driven high b15 b12 0 0 0 0: Setting prohibited 0 0 0 1: Receive FIFO threshold value is 1 : : 1 1 1 1: Receive FIFO threshold value is 15	R/W

Note 1. These bits are enabled in asynchronous mode and clock synchronous mode only.
Note 2. These bits can be rewritten only when the TE and RE bits in the SCR register are both 0.
Note 3. These bits are enabled only when the FM bit is 1.
Note 4. These bits are enabled when the FM bit is 1, and the CTSE and SSE bits in the SPMR register are both 0 (RTS# output is enabled).

FM Bit (FIFO Mode Select)

This bit is used to enable FIFO mode. Setting this bit to 1 enables FIFO, and causes FTDR and FRDR to be used as the data register for transmit and receive, respectively.

Setting this bit to 0 disables FIFO, and causes the TDR and RDR registers, or the TDRHL, TDRH, TDRL, RDRHL, RDRH, and RDRL registers to be used as the data register for transmit and receive.

This bit is enabled in asynchronous mode and clock synchronous mode only. This bit should be set to 0 in any other mode. When rewriting this bit, transmit and receive should be disabled.

RFRST Bit (Receive FIFO Reset)

Setting the RFRST bit to 1, causes the FDR.R[4:0] bits to become 0. The value of the RFRST bit automatically returns to 0 after 1 PCLK.

TFRST Bit (Transmit FIFO Reset)

Setting the TFRST bit to 1, causes the FDR.T[4:0] bits to become 0. The value of the TFRST bit automatically returns to 0 after 1 PCLK.

DRES Bit (Receive Data Ready Interrupt Select)

This bit is used to select the interrupt request that is generated when the SSRFIFO.DR flag becomes 1. When this bit is 0, a DR flag of 1 generates a receive data full interrupt (RXI) request. When this bit is 1, a DR flag of 1 generates a receive error interrupt (ERI) request.

TTRG[3:0] Bits (Transmit FIFO Threshold Setting)

The SSRFIFO.TDFE flag becomes 1 when the value of the FDR.T[4:0] bits becomes equal to or less than the value set in the TTRG[3:0] bits. At this time, if the SCR.TIE bit is 1, a transmit data empty interrupt (TXI) request is generated.

RTRG[3:0] Bits (Receive FIFO Threshold Setting)

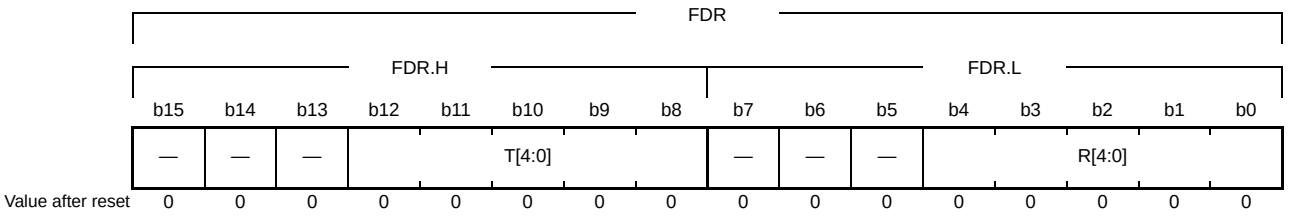
The SSRFIFO.RDF flag becomes 1 when the values of the FDR.R[4:0] bits are equal to or greater than the values set in the RTRG[3:0] bits. At this time, if the SCR.RIE bit is 1, a receive data full interrupt (RXI) request is generated.

RSTRG[3:0] Bits (RTS# Output Threshold Setting)

When the values of the FDR.R[4:0] bits becomes equal to or greater than the values set in the RSTRG[3:0] bits, the RTSn# pin is driven high. When using this function, set both the CTSE and SSE bits in the SPMR register to 0 to enable RTSn# output.

35.2.23 FIFO Data Count Register (FDR)

Address(es): SCI10.FDR 000D 0056h, SCI11.FDR 000D 0076h,
SCI10.FDR.H 000D 0056h, SCI11.FDR.H 000D 0076h,
SCI10.FDR.L 000D 0057h, SCI11.FDR.L 000D 0077h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	R[4:0]	Receive FIFO Data Count*1	These bits indicate the number of data stored in the receive FIFO	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12 to b8	T[4:0]	Transmit FIFO Data Count*1	These bits indicate the number of data stored in the transmit FIFO	R
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. These bits are valid only in asynchronous mode or clock synchronous mode.

R[4:0] Bits (Receive FIFO Data Count)

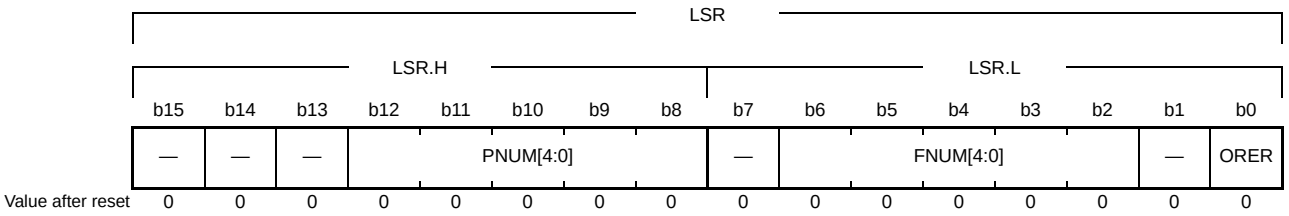
These bits indicate the number of data stored in the receive FIFO. 00h indicates that no received data is in FIFO, and 10h indicates that 16 frame data received have been stored in FIFO.
These bits never have values in the range from 11h to 1Fh.

T[4:0] Bits (Transmit FIFO Data Count)

These bits indicate the number of untransmitted data in the transmit FIFO. 00h indicates that no untransmitted data is in FIFO, and 10h indicates that 16 frames of untransmitted data are stored in FIFO.
These bits never have values in the range from 11h to 1Fh.

35.2.24 Line Status Register (LSR)

Address(es): SCI10.LSR 000D 0058h, SCI11.LSR 000D 0078h,
SCI10.LSR.H 000D 0058h, SCI11.LSR.H 000D 0078h,
SCI10.LSR.L 000D 0059h, SCI11.LSR.L 000D 0079h



Bit	Symbol	Bit Name	Description	R/W
b0	ORER	Overrun Error Flag*1	The same value as that in the SSRFIFO.ORER flag is read. 0: An overrun error has not occurred. 1: An overrun error has occurred.	R*2
b1	—	Reserved	This bit are read as 0. The write value should be 0.	R
b6 to b2	FNUM[4:0]	Framing Error Count	These bits indicate the number of data with a framing error among the data stored in the receive FIFO.	R
b7	—	Reserved	This bit are read as 0. The write value should be 0.	R
b12 to b8	PNUM[4:0]	Parity Error Count	These bits indicate the number of data with a parity error among the data stored in the receive FIFO.	R
b15 to b13	—	Reserved	This bit are read as 0. The write value should be 0.	R

Note 1. The bit can be read only in FIFO mode, and asynchronous mode or clock synchronous mode.
Note 2. To clear this flag, confirm that the SSRFIFO.ORER flag is 1 first, and then write 0 to the SSRFIFO.ORER flag.

ORER Flag (Overrun Error Flag)

This bit becomes 1 when an overrun error occurs. The value of the SSRFIFO.ORER flag is reflected to this bit. To clear this flag, clear the SSRFIFO.ORER flag.

FNUM[4:0] Bits (Framing Error Count)

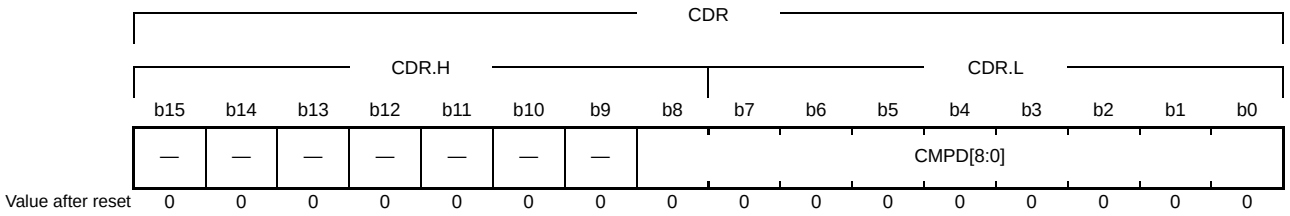
These bits indicate the number of data in which a framing error has occurred among the received data in the receive FIFO.

PNUM[4:0] Bits (Parity Error Count)

These bits indicate the number of data in which a parity error has occurred among the received data in the receive FIFO.

35.2.25 Comparison Data Register (CDR)

Address(es): SCI0.CDR 0008 A01Ah, SCI1.CDR 0008 A03Ah, SCI2.CDR 0008 A05Ah, SCI3.CDR 0008 A07Ah, SCI4.CDR 0008 A09Ah, SCI5.CDR 0008 A0BAh, SCI6.CDR 0008 A0DAh, SCI7.CDR 0008 A0FAh, SCI8.CDR 0008 A11Ah, SCI9.CDR 0008 A13Ah, SCI10.CDR 000D 005Ah, SCI11.CDR 000D 007Ah,
SCI0.CDR.H 0008 A01Ah, SCI1.CDR.H 0008 A03Ah, SCI2.CDR.H 0008 A05Ah, SCI3.CDR.H 0008 A07Ah, SCI4.CDR.H 0008 A09Ah, SCI5.CDR.H 0008 A0BAh, SCI6.CDR.H 0008 A0DAh, SCI7.CDR.H 0008 A0FAh, SCI8.CDR.H 0008 A11Ah, SCI9.CDR.H 0008 A13Ah, SCI10.CDR.H 000D 005Ah, SCI11.CDR.H 000D 007Ah,
SCI0.CDR.L 0008 A01Bh, SCI1.CDR.L 0008 A03Bh, SCI2.CDR.L 0008 A05Bh, SCI3.CDR.L 0008 A07Bh, SCI4.CDR.L 0008 A09Bh, SCI5.CDR.L 0008 A0BBh, SCI6.CDR.L 0008 A0DBh, SCI7.CDR.L 0008 A0FBh, SCI8.CDR.L 0008 A11Bh, SCI9.CDR.L 0008 A13Bh, SCI10.CDR.L 000D 005Bh, SCI11.CDR.L 000D 007Bh



Bit	Symbol	Bit Name	Description	R/W
b8 to b0	CMPD[8:0]	Comparison Data	These bits specify the data of comparison source when using the data match detection function.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R

CMPD[8:0] Bits (Comparison Data)

These bits are used for detecting a data match. The length of the valid bit is the same as the character length set in the SMR.CHR and SCMR.CHR1 bits.

The DCCR.DCMF flag becomes 1 when the received data matches with the value of these bits.

35.2.26 Data Comparison Control Register (DCCR)

Address(es): SCI0.DCCR 0008 A013h, SCI1.DCCR 0008 A033h, SCI2.DCCR 0008 A053h, SCI3.DCCR 0008 A073h, SCI4.DCCR 0008 A093h, SCI5.DCCR 0008 A0B3h, SCI6.DCCR 0008 A0D3h, SCI7.DCCR 0008 A0F3h, SCI8.DCCR 0008 A113h, SCI9.DCCR 0008 A133h, SCI10.DCCR 000D 0053h, SCI11.DCCR 000D 0073h

	b7	b6	b5	b4	b3	b2	b1	b0
	DCME	IDSEL	—	DFER	DPER	—	—	DCMF
Value after reset	0	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DCMF	Data Match Flag	0: Data is not matched 1: Data is matched	R/(W) *1
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0	R/W
b3	DPER	Match Data Parity Error Flag	0: A parity error is not found in the matched data. 1: A parity error is found in the matched data.	R/(W) *1
b4	DFER	Match Data Framing Error Flag	0: A framing error is not found in the matched data. 1: A framing error is found in the matched data.	R/(W) *1
b5	—	Reserved	This bit is read as 0. The write value should be 0	R/W
b6	IDSEL	ID Frame Select*2	0: All data is to be compared 1: The data with the multi-processor bit set to 1 is to be compared	R/W
b7	DCME	Data Match Detection Enable*2	0: Data match detection is disabled 1: Data match detection is enabled	R/W

Note 1. Only 0 can be written to this bit, which clears the flag. To clear this flag, confirm that the flag is 1, and then write 0 to the flag.
Note 2. This bit is only valid in asynchronous mode.

DCMF Flag (Data Match Flag)

This flag indicates the comparison result of the received data and the value of the CDR register.

[Setting condition]

- When the received data matches with the value in the CDR register while the DCME bit is 1

[Clearing condition]

- When 0 is written to the flag after reading this flag as 1

Even when the SCR.RE bit is set to 0, the DCMF flag has no effect and retains the previous state.

DPER Flag (Match Data Parity Error Flag)

This flag indicates if a parity error has occurred in the matched data.

[Setting condition]

- When a parity error is found in the received data in which a data match is detected.

[Clearing condition]

- When 0 is written to the flag after reading the flag as 1.

Refer to section 15.5.2, Level Detection to exit from the interrupt handling routine after setting the DPER flag to 0.

Even when the SCR.RE bit is set to 0, the DPER flag has no effect and retains the previous state.

DFER Flag (Match Data Framing Error Flag)

This flag indicates if a framing error has occurred in the matched data.

[Setting condition]

- When the stop bit of the received frame in which a data match is detected is 0

[Clearing condition]

- When 0 is written to the flag after reading the flag as 1.

Refer to section 15.5.2, Level Detection to exit from the interrupt handling routine after setting the DFER flag to 0.

Even when the SCR.RE bit is set to 0, the DFER flag has no effect and retains the previous state.

IDSEL Bit (ID Frame Select)

This bit specifies the condition of the received data to be compared. The bit is valid only when the DCME bit is 1.
When setting this bit to 1, only data in received frames (ID frames) in which the multi-processor bit has the value 1 are compared.
When this bit is set to 0, all received data is compared.

DCME Bit (Data Match Detection Enable)

This bit enables or disables the data match detection function. The function is valid only in asynchronous mode. In other modes, set this bit to 0.
This bit automatically becomes 0 when a data match is detected.

35.2.27 Serial Port Register (SPTR)

Address(es): SCI0.SPTR 0008 A01Ch, SCI1.SPTR 0008 A03Ch, SCI2.SPTR 0008 A05Ch, SCI3.SPTR 0008 A07Ch, SCI4.SPTR 0008 A09Ch, SCI5.SPTR 0008 A0BCh, SCI6.SPTR 0008 A0DCh, SCI7.SPTR 0008 A0FCh, SCI8.SPTR 0008 A11Ch, SCI9.SPTR 0008 A13Ch, SCI10.SPTR 000D 005Ch, SCI11.SPTR 000D 007Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	TTADJ	RTADJ	TINV	RINV	—	SPB2IO	SPB2DT	RXDMON
Value after reset	0	0	0	0	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	RXDMON	RXD Line Monitoring Flag	When the RINV bit is 0 0: The RXDn pin level is low 1: The RXDn pin level is high When the RINV bit is 1 0: The RXDn pin level is high 1: The RXDn pin level is low	R
b1	SPB2DT	Serial Port Break Data*1	Combine bits SPB2DT, SPB2IO, TINV, and SCR.TE to control the TXDn pin. Refer to Table 35.30 for details.	R/W
b2	SPB2IO	Serial Port Break I/O*1		R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b4	RINV	Receiver Input Invert*2	0: Does not invert the input signal from the RXD pin 1: Inverts the input signal from the RXD pin	R/W *3
b5	TINV	Transmitter Output Invert*2	0: Does not invert the output signal to the TXD pin 1: Invert the output signal to the TXD pin	R/W *3
b6	RTADJ	Receive Data Sampling Timing Adjustment*4	0: Does not adjust the sampling point of receive data 1: Adjust the sampling point of receive data	R/W *3
b7	TTADJ	Transmit Signal Transition Timing Adjustment*4	0: Does not adjust the transition timing of transmit data 1: Adjust the transition timing of transmit data	R/W *3

Note 1. This bit is only valid in asynchronous mode.
Note 2. Set this bit to 0 if operation is to be in smart card interface mode or simple I^2C mode.
Note 3. This bit is rewritable only when the TE and RE bits in the SCR register are both 0.
Note 4. This bit is only valid when the on-chip baud rate generator is selected as the clock source in asynchronous mode.

RXDMON Flag (RXD Line Monitoring Flag)

This flag is used for monitoring the level of the RXDn pin.

SPB2DT Bit (Serial Port Break Data)

This bit specifies the output level of the TXDn pin when the SCR.TE bit is 0. Refer to Table 35.30 for details.

SPB2IO Bit (Serial Port Break I/O)

This bit specifies input or output of the TXDn pin when the SCR.TE bit is 0. Set this bit to 1 (output) when controlling the TXDn pin by software.

Table 35.30 Controlling the TXDn pin

SCR.TE Bit Setting	SPB2IO Bit Setting	SPB2DT Bit Setting	TINV Bit Setting	TXDn Pin Status
0 (Transmission disabled)	0 (Input)	0 or 1	0 or 1	Hi-Z
	1 (Output)	0	0	Low is output
			1	High is output
		1	0	High is output
			1	Low is output
1 (Transmission enabled)	0 or 1	0 or 1	0 or 1	Transmit data output pin

RINV Bit (Receiver Input Invert)

This bit is used to logically invert the input signal from the RXDn pin in front of the receive shift register. Besides data bits, start bit, parity bit, and stop bit are inverted.

TINV Bit (Transmitter Output Invert)

This bit is used to logically invert the output signal from the transmit shift register in front of the TXDn pin. Besides data bits, start bit, parity bit, and stop bit are inverted.

RTADJ Bit (Receive Data Sampling Timing Adjustment)

This bit is used to change the sampling point of the receive data from the default sampling point. This bit is used to improve the receive margin when the high and low widths of the receive signal are changed due to the characteristics of the transmission line or interface devices. Normally, set this bit to 0.

TTADJ Bit (Transmit Signal Transition Timing Adjustment)

This bit is used to change the transition point of the transmit data from the default transition point. This bit is used to improve the receive margin of the receiver when the high and low widths of the transmitted signal are intended to be changed due to the characteristics of the transmission line or interface devices. Normally, set this bit to 0.

35.2.28 Transmit/Receive Timing Select Register (TMGR)

Address(es): SCI0.TMGR 0008 A01Dh, SCI1.TMGR 0008 A03Dh, SCI2.TMGR 0008 A05Dh, SCI3.TMGR 0008 A07Dh, SCI4.TMGR 0008 A09Dh, SCI5.TMGR 0008 A0BDh, SCI6.TMGR 0008 A0DDh, SCI7.TMGR 0008 A0FDh, SCI8.TMGR 0008 A11Dh, SCI9.TMGR 0008 A13Dh, SCI10.TMGR 000D 005Dh, SCI11.TMGR 000D 007Dh

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

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	RTMG[3:0]	Receive Data Sampling Timing Select*1	b3 b0 1 1 1 1: Data are sampled 7 clocks earlier than default point. 1 1 1 0: Data are sampled 6 clocks earlier than default point. 1 1 0 1: Data are sampled 5 clocks earlier than default point. 1 1 0 0: Data are sampled 4 clocks earlier than default point. 1 0 1 1: Data are sampled 3 clocks earlier than default point. 1 0 1 0: Data are sampled 2 clocks earlier than default point. 1 0 0 1: Data are sampled 1 clock earlier than default point. x 0 0 0: Data are sampled at default point. 0 0 0 1: Data are sampled 1 clock later than default point. 0 0 1 0: Data are sampled 2 clocks later than default point. 0 0 1 1: Data are sampled 3 clocks later than default point. 0 1 0 0: Data are sampled 4 clocks later than default point. 0 1 0 1: Data are sampled 5 clocks later than default point. 0 1 1 0: Data are sampled 6 clocks later than default point. 0 1 1 1: Data are sampled 7 clocks later than default point.	R/W *2
b7 to b4	TTMG[3:0]	Transmit Signal Transition Timing Select*3	b7 b4 1 1 1 1: Delays the 1 to 0 transitions for 7 clocks. 1 1 1 0: Delays the 1 to 0 transitions for 6 clocks. 1 1 0 1: Delays the 1 to 0 transitions for 5 clocks. 1 1 0 0: Delays the 1 to 0 transitions for 4 clocks. 1 0 1 1: Delays the 1 to 0 transitions for 3 clocks. 1 0 1 0: Delays the 1 to 0 transitions for 2 clocks. 1 0 0 1: Delays the 1 to 0 transitions for 1 clock. x 0 0 0: Does not change the waveform. 0 0 0 1: Delays the 0 to 1 transitions for 1 clock. 0 0 1 0: Delays the 0 to 1 transitions for 2 clocks. 0 0 1 1: Delays the 0 to 1 transitions for 3 clocks. 0 1 0 0: Delays the 0 to 1 transitions for 4 clocks. 0 1 0 1: Delays the 0 to 1 transitions for 5 clocks. 0 1 1 0: Delays the 0 to 1 transitions for 6 clocks. 0 1 1 1: Delays the 0 to 1 transitions for 7 clocks.	R/W *4

Note 1. These bits are only valid when the SPTR.RTADJ bit is 1.
Note 2. These bits are rewritable only when the SPTR.RTADJ bit is 0.
Note 3. These bits are only valid when the SPTR.TTADJ bit is 1.
Note 4. These bits are rewritable only when the SPTR.TTADJ bit is 0.

The TMGR register is used to adjust the sampling point of the receive data and the transition timing of the transmit data. This register is only valid when the on-chip baud rate generator is selected as the clock source in asynchronous mode. This register is not present in SCI12.

RTMG[3:0] Bits (Receive Data Sampling Timing Select)

These bit are used to select the sampling points of the receive data. These bit are only valid when the SPTR.RTADJ bit is 1. When the RTMG[3] bit is 0, each bit is sampled later than the default sampling point. When 1, each bit is sampled earlier than the default sampling point.
Set the amount of movement of the sampling points to the RTMG[2:0] bits by the number of the base clocks. Refer to Table 35.31 for the range of the value that can be set in the RTMG[2:0] bits.

Table 35.31 Range of Setting Values for the RTMG[2:0] Bits

SEMR.ABCSE Bit Setting	SEMR.ABCS Bit Setting	Number of the Base Clock Cycles for 1 Data Bit	Setting Range
0	0	16 cycles	0 to 7 (000b to 111b)
0	1	8 cycles	0 to 3 (000b to 011b)
1	0 or 1	6 cycles	0 to 2 (000b to 010b)

TTMG[3:0] Bits (Transmit Signal Transition Timing Select)

These bits are used to select the transition timing of the transmit signal in the transmit shift register. These bits are only valid when the SPTR.TTADJ bit is 1.

When the TTMG[3] bit is 0, the 0 to 1 transitions are delayed. When the TTMG[3] bit is 1, the 1 to 0 transitions are delayed. Output signal from the TXDn pin depends on the SPTR.TINV bit as follows.

(1) When the SPTR.TINV bit is 0

When the TTMG[3] bit is 0, high pulse width is narrower than low pulse width because low to high transitions (rising edges) are delayed.

When the TTMG[3] bit is 1, high pulse width is wider than low pulse width because high to low transitions (falling edges) are delayed.

(2) When the TINV bit is 1

When the TTMG[3] bit is 0, high pulse width is wider than low pulse width because high to low transitions (falling edges) are delayed.

When the TTMG[3] bit is 1, high pulse width is narrower than low pulse width because low to high transitions (rising edges) are delayed.

Set the delay amount to the TTMG[2:0] bits in number of the base clock. Refer to Table 35.32 for the range of the value that can be set in the TTMG[2:0] bits.

Table 35.32 Range of Setting Values for the TTMG[2:0] Bits

SEMR.ABCSE Bit Setting	SEMR.ABCS Bit Setting	Number of the Base Clock Cycles for 1 Data Bit	Setting Range
0	0	16 cycles	0 to 7 (000b to 111b)
0	1	8 cycles	0 to 7 (000b to 111b)
1	0 or 1	6 cycles	0 to 5 (000b to 101b)

35.2.29 Extended Serial Module Enable Register (ESMER)

Address(es): SCI12.ESMER 0008 B320h

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

Bit	Symbol	Bit Name	Description	R/W
b0	ESME	Extended Serial Mode Enable	0: The extended serial mode is disabled. 1: The extended serial mode is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ESME Bit (Extended Serial Mode Enable)

When the ESME bit is 1, the facilities of the extended serial mode control section are enabled.

When the ESME bit is 0, the extended serial mode control section is initialized.

Table 35.33 Settings of the ESME Bit and Timer Operation Mode

ESME Bit	Timer Mode	Break Field Low Width Determination Mode	Break Field Low Width Output Mode
0	Available*1	Not available	Not available
1	Available	Available	Available

Note 1. Operation is only possible with PCLK selected.

35.2.30 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

35.2.31 Control Register 1 (CR1)

Address(es): SCI12.CR1 0008 B322h

	b7	b6	b5	b4	b3	b2	b1	b0
	PIBS[2:0]			PIBE	CF1DS[1:0]		CF0RE	BFE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BFE	Break Field Enable	0: Break Field detection is disabled. 1: Break Field detection is enabled.	R/W
b1	CF0RE	Control Field 0 Reception Enable	0: Reception of Control Field 0 is disabled. 1: Reception of Control Field 0 is enabled.	R/W
b3, b2	CF1DS[1:0]	Control Field 1 Data Register Select	b3 b2 0 0: Selects comparison with the value in the PCF1DR register. 0 1: Selects comparison with the value in the SCF1DR register. 1 0: Selects comparison with the values in the PCF1DR and SCF1DR registers. 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

35.2.32 Control Register 2 (CR2)

Address(es): SCI12.CR2 0008 B323h

	b7	b6	b5	b4	b3	b2	b1	b0
	RTS[1:0]		BCCS[1:0]		—	DFCS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	DFCS[2:0]	RXDX12 Signal Digital Filter Clock Select	b2 b0 0 0 0: Filter is disabled. 0 0 1: Filter clock is base clock*1, *2 0 1 0: Filter clock is PCLK/8 0 1 1: Filter clock is PCLK/16 1 0 0: Filter clock is PCLK/32 1 0 1: Filter clock is PCLK/64 1 1 0: Filter clock is PCLK/128 1 1 1: Setting prohibited	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	BCCS[1:0]	Bus Collision Detection Clock Select	• When SEMR.BGDM = 0 or SEMR.BGDM = 1 and SMR.CKS[1:0] = a value other than 00b b5 b4 0 0: Base clock 0 1: Base clock frequency divided by 2 1 0: Base clock frequency divided by 4 1 1: Setting prohibited • When SEMR.BGDM = 1 and SMR.CKS[1:0] = 00b b5 b4 0 0: Base clock frequency divided by 2 0 1: Base clock frequency divided by 4 1 0: Setting prohibited 1 1: Setting prohibited	R/W
b7, b6	RTS[1:0]	RXDX12 Reception Sampling Timing Select	• When SCI12.SEMR.ABCS = 0 b7 b6 0 0: Rising edge of the 8th cycle of base clock 0 1: Rising edge of the 10th cycle of base clock 1 0: Rising edge of the 12th cycle of base clock 1 1: Rising edge of the 14th cycle of base clock • When SCI12.SEMR.ABCS = 1 b7 b6 0 0: Rising edge of the 4th cycle of base clock 0 1: Rising edge of the 5th cycle of base clock 1 0: Rising edge of the 6th cycle of base clock 1 1: Rising edge of the 7th cycle of base clock	R/W

Note: The period of the base clock is 1/16 of a single bit period when the SCI12.SEMR.ABCS is 0, and 1/8 of a single bit period when the SCI12.SEMR.ABCS is 1.
Note 1. To use the base clock, set the SCI12.SCR.TE bit to 1.
Note 2. The base clock divided by 2 is the filter clock when the SEMR.BGDM bit is 1 and the SMR.CKS[1:0] bits are 00b.

35.2.33 Control Register 3 (CR3)

Address(es): SCI12.CR3 0008 B324h

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

Bit	Symbol	Bit Name	Description	R/W
b0	SDST	Start Frame Detection Start	0: Detection of Start Frame is not performed. 1: Detection of Start Frame is performed.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SDST Bit (Start Frame Detection Start)

Detection of a Start Frame begins when this bit is set to 1. The bit is read as 0.

35.2.34 Port Control Register (PCR)

Address(es): SCI12.PCR 0008 B325h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SHARPS	—	—	RXDXPS	TXDXPS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TXDXPS	TXDX12 Signal Polarity Select	0: The polarity of TXDX12 signal is not inverted for output. 1: The polarity of TXDX12 signal is inverted for output.	R/W
b1	RXDXPS	RXDX12 Signal Polarity Select	0: The polarity of RXDX12 signal is not inverted for input. 1: The polarity of RXDX12 signal is inverted for input.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SHARPS	TXDX12/RXDX12 Pin Multiplexing Select	0: The TXDX12 and RXDX12 pins are independent. 1: The TXDX12 and RXDX12 signals are multiplexed on the same pin.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SHARPS Bit (TXDX12/RXDX12 Pin Multiplexing Select)

When this bit is set to 1, the TXDX12 and RXDX12 signals are multiplexed on the same pin so that half-duplex communications become possible.

35.2.35 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

35.2.36 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 STCR.BFDCL bit	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 STCR.CF0MCL bit	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 STCR.CF1MCL bit	R
b3	PIBDF	Priority Interrupt Bit Detection Flag	[Setting condition] • Detection of the priority interrupt bit [Clearing condition] • Writing 1 to the STCR.PIBDCL bit	R
b4	BCDF	Bus Collision Detected Flag	[Setting condition] • Detection of the bus collision [Clearing condition] • Writing 1 to the STCR.BCDCL bit	R
b5	AEDF	Valid Edge Detection Flag	[Setting condition] • Detection of a valid edge [Clearing condition] • Writing 1 to the STCR.AEDCL bit	R
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R

35.2.37 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

35.2.38 Control Field 0 Data Register (CF0DR)

Address(es): SCI12.CF0DR 0008 B329h

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

The CF0DR register is an 8-bit readable and writable register that holds a value for comparison with Control Field 0.

35.2.39 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

35.2.40 Control Field 0 Receive Data Register (CF0RR)

Address(es): SCI12.CF0RR 0008 B32Bh

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

CF0RR is a readable register that holds the value received in Control Field 0.

35.2.41 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.

35.2.42 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.

35.2.43 Control Field 1 Compare Enable Register (CF1CR)

Address(es): SCI12.CF1CR 0008 B32Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	CF1CE7	CF1CE6	CF1CE5	CF1CE4	CF1CE3	CF1CE2	CF1CE1	CF1CE0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CF1CE0	Control Field 1 Bit 0 Compare Enable	0: Comparison with bit 0 of Control Field 1 is disabled. 1: Comparison with bit 0 of Control Field 1 is enabled.	R/W
b1	CF1CE1	Control Field 1 Bit 1 Compare Enable	0: Comparison with bit 1 of Control Field 1 is disabled. 1: Comparison with bit 1 of Control Field 1 is enabled.	R/W
b2	CF1CE2	Control Field 1 Bit 2 Compare Enable	0: Comparison with bit 2 of Control Field 1 is disabled. 1: Comparison with bit 2 of Control Field 1 is enabled.	R/W
b3	CF1CE3	Control Field 1 Bit 3 Compare Enable	0: Comparison with bit 3 of Control Field 1 is disabled. 1: Comparison with bit 3 of Control Field 1 is enabled.	R/W
b4	CF1CE4	Control Field 1 Bit 4 Compare Enable	0: Comparison with bit 4 of Control Field 1 is disabled. 1: Comparison with bit 4 of Control Field 1 is enabled.	R/W
b5	CF1CE5	Control Field 1 Bit 5 Compare Enable	0: Comparison with bit 5 of Control Field 1 is disabled. 1: Comparison with bit 5 of Control Field 1 is enabled.	R/W
b6	CF1CE6	Control Field 1 Bit 6 Compare Enable	0: Comparison with bit 6 of Control Field 1 is disabled. 1: Comparison with bit 6 of Control Field 1 is enabled.	R/W
b7	CF1CE7	Control Field 1 Bit 7 Compare Enable	0: Comparison with bit 7 of Control Field 1 is disabled. 1: Comparison with bit 7 of Control Field 1 is enabled.	R/W

35.2.44 Control Field 1 Receive Data Register (CF1RR)

Address(es): SCI12.CF1RR 0008 B32Fh

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

CF1RR is a readable register that holds the value received in Control Field 1.

35.2.45 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

35.2.46 Timer Mode Register (TMR)

Address(es): SCI12.TMR 0008 B331h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	TCSS[2:0]			TWRC	—	TOMS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TOMS[1:0]	Timer Operating Mode Select*1	b1 b0 0 0: Timer mode 0 1: Break Field low width determination mode 1 0: Break Field low width output mode 1 1: Setting prohibited	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	TWRC	Counter Write Control	0: Data is written to the reload register and counter 1: Data is written to the reload register only	R/W
b6 to b4	TCSS[2:0]	Timer Count Clock Source Select*1	b6 b4 0 0 0: PCLK 0 0 1: PCLK/2 0 1 0: PCLK/4 0 1 1: PCLK/8 1 0 0: PCLK/16 1 0 1: PCLK/32 1 1 0: PCLK/64 1 1 1: PCLK/128	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Rewrite the TOMS[1:0] and TCSS[2:0] bits only when the timer is stopped (TCST = 0).

TWRC Bit (Counter Write Control)

This bit determines whether a value written to the TPRE or TCNT register is written to the reload register only or is written to both the reload register and the counter.

35.2.47 Timer Prescaler Register (TPRE)

Address(es): SCI12.TPRE 0008 B332h

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

TPRE consists of an 8-bit reload register, a read buffer, and a counter, each of which has FFh as its initial value. The counter counts down in synchronization with the counter clock selected by the TMR.TCSS[2:0] bits, and is reloaded with the value in the reload register when it underflows. Underflows of this register provide the clock source to drive counting by the TCNT register. The reload register and read buffer are allocated to the same address. Data is written to the reload register in writing, and the counter value that has been transferred to the read buffer is returned in reading.
It takes one PCLK cycle to load a value from the reload register to the counter.

35.2.48 Timer Count Register (TCNT)

Address(es): SCI12.TCNT 0008 B333h

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

TCNT consists of an 8-bit reload register, a read buffer, and a counter, each of which has FFh as its initial value. This down-counter counts underflows of the TPRE register until the TCNT register underflows, and is then reloaded with the value from the reload register. The reload register and read buffer are allocated to the same address. Data is written to the reload register in writing, and the counter value that has been transferred to the read buffer is returned in reading.
It takes one PCLK cycle to load a value from the reload register to the counter.

35.3 Operation in Asynchronous Mode

Figure 35.6 shows the general format for asynchronous serial communications.
One frame consists of a start bit (low level), transmit/receive data, a parity bit, and stop bits (high level).
In asynchronous serial communications, the communications line is usually held in the mark state (high level).
The SCI monitors the communications line. When the SCI detects a low, it regards that as a start bit and starts serial communication.
Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception. As SCI10 and SCI11 incorporate a FIFO, data can be received and transmitted more efficiently.

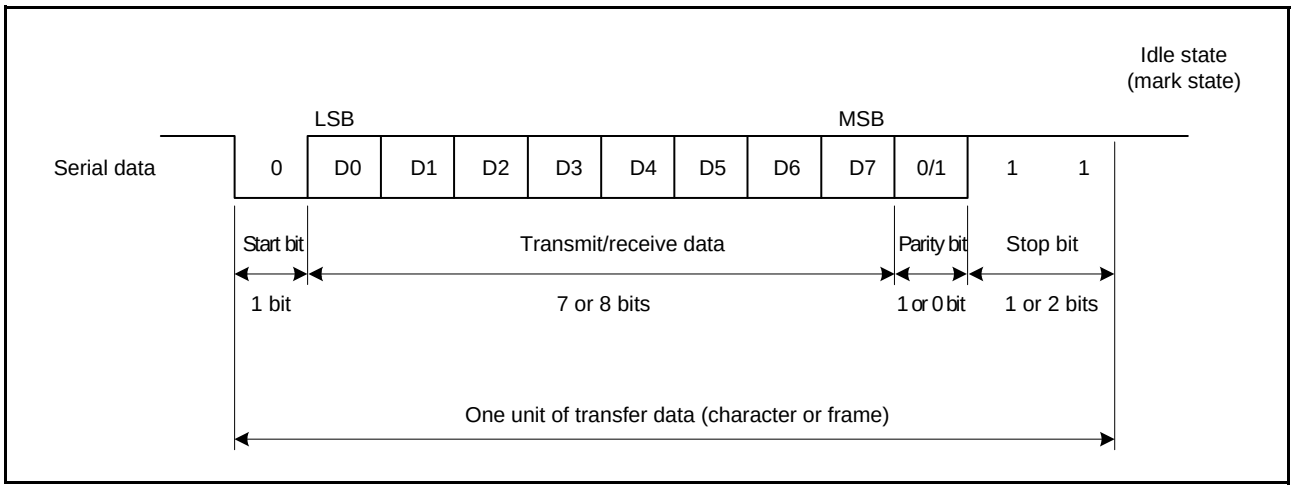


Figure 35.6 Data Format in Asynchronous Serial Communications (Example with 8-Bit Data, Parity, 2 Stop Bits)

35.3.1 Serial Data Transfer Format

Table 35.34 lists the serial data transfer formats that can be used in asynchronous mode.
Any of 18 transfer formats can be selected according to the SMR and SCMR setting. For details of multi-processor function, refer to section 35.4, Multi-Processor Communications Function.

Table 35.34 Serial Transfer Formats (Asynchronous Mode)

SCMR Setting	SMR Setting				Serial Transfer Format and Frame Length
CHR1	CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12 13
0	0	0	0	0	S 9-bit data STOP
0	0	0	0	1	S 9-bit data STOP STOP
0	0	1	0	0	S 9-bit data P STOP
0	0	1	0	1	S 9-bit data P STOP STOP
1	0	0	0	0	S 8-bit data STOP
1	0	0	0	1	S 8-bit data STOP STOP
1	0	1	0	0	S 8-bit data P STOP
1	0	1	0	1	S 8-bit data P STOP STOP
1	1	0	0	0	S 7-bit data STOP
1	1	0	0	1	S 7-bit data STOP STOP
1	1	1	0	0	S 7-bit data P STOP
1	1	1	0	1	S 7-bit data P STOP STOP
0	0	—	1	0	S 9-bit data MPB STOP
0	0	—	1	1	S 9-bit data MPB STOP STOP
1	0	—	1	0	S 8-bit data MPB STOP
1	0	—	1	1	S 8-bit data MPB STOP STOP
1	1	—	1	0	S 7-bit data MPB STOP
1	1	—	1	1	S 7-bit data MPB STOP STOP

S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multi-processor bit

35.3.2 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*2 of each bit, as shown in Figure 35.7. Thus the reception margin in asynchronous mode is determined by formula (1) below.

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ (\%) \quad \cdots \text{Formula (1)}$$

M: Reception margin
N: Ratio of bit rate to clock
(N = 16 when SEMR.ABCSE = 0 and SEMR.ABCS = 0, N = 8 when SEMR.ABCSE = 0 and SEMR.ABCS = 1, N = 6 when SEMR.ABCSE = 1)
D: Duty cycle of clock (D = 0.5 to 1.0)
L: Frame length (L = 9 to 13)
F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined by the formula below.

M = {0.5 – 1/(2 × 16)} × 100 (%) = 46.875 (%)

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

Note 1. These are values when the ABCSE and ABCS bits in the SEMR register are 0. When the ABCSE bit is 0 and 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. When the ABCSE bit is 1, a frequency of 6 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 3rd pulse of the base clock.
Note 2. This is when the SPTR.RTADJ bit is 0.

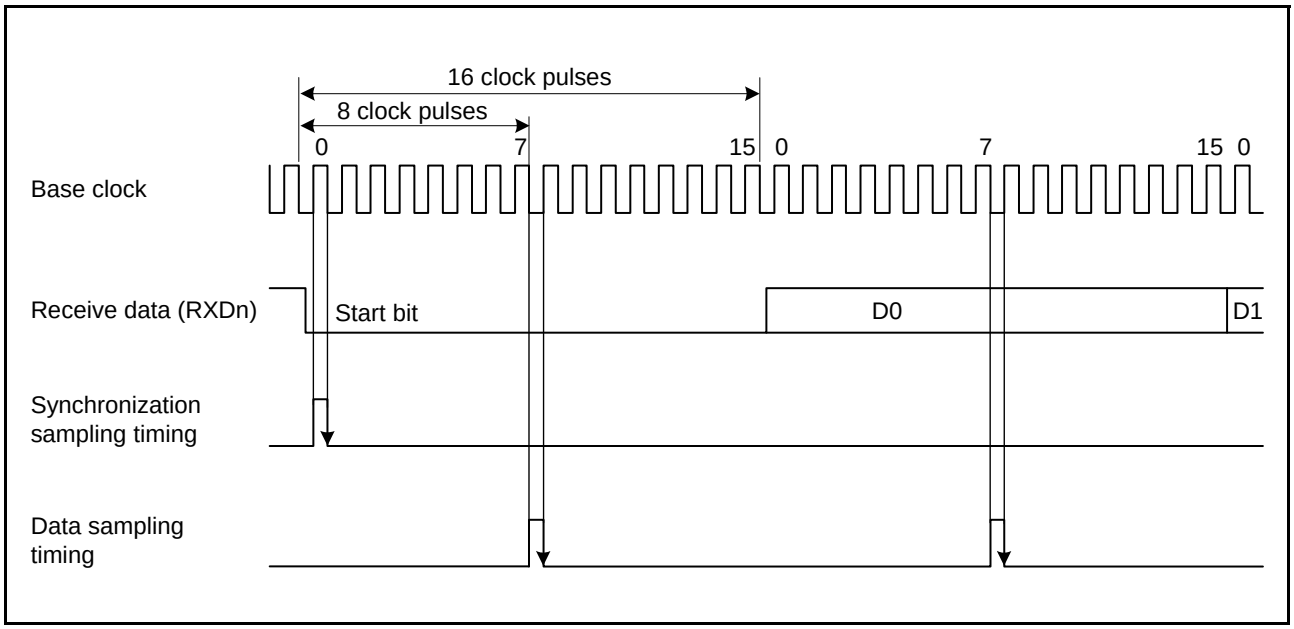


Figure 35.7 Receive Data Sampling Timing in Asynchronous Mode

In SCI0 to SCI11, there are functions to adjust the sampling point of the receive data and the transition timing of the transmit data against that the high width and low width of the signal changes affected by the devices on the transmission line.

35.3.2.1 Receive Data Sampling Timing Adjustment

When a waveform such that high width is different from low width because the difference between rising time and falling time is large is received, adjust the timing for data to be sampled at the center of the narrower pulse. When low width is narrower than high width, move the sampling timing forward, and when high width is narrower than low width, move the sampling timing backward.

When the TMGR.RTMG[3:0] bits are set to an offset to the default sampling point and then the SPTR.RTADJ bit is set to 1, received data is sampled at the specified position.

Figure 35.8 shows an example of the sampling timing adjustment.

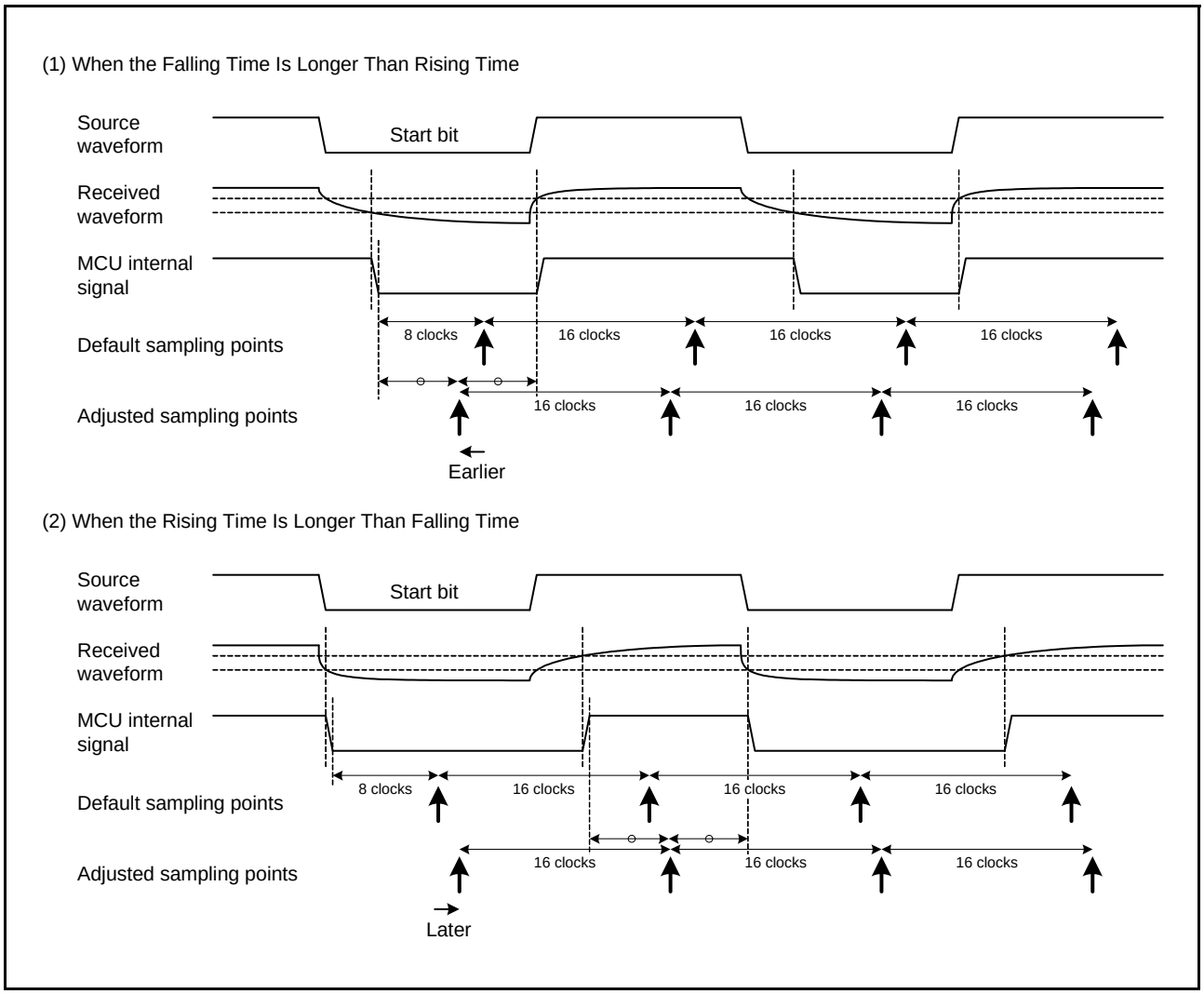


Figure 35.8 Example of Sampling Timing Adjustment (SEMR.ABCSE bit = 0 and SEMR.ABCS bit = 0)

35.3.2.2 Transmit Data Transition Timing Adjustment

When a difference between high width and low width for a waveform transmitted by this MCU arises at the receiver, it is also possible to make a difference between the high width and the low width beforehand at transmission to adjust so that the difference disappears at the receiver. When high width becomes narrower at the receiver, expand the high width of the transmit signal by delaying the falling edges, and when low width becomes narrower at the receiver, expand the low width of the transmit signal by delaying the rising edges.

When the TMGR.TTMG[3:0] bits are set to the transition direction and the delay amount and the SPTR.TTADJ bit is set to 1, transmit data changes at the specified point.

Figure 35.9 shows an example of the transition timing adjustment.

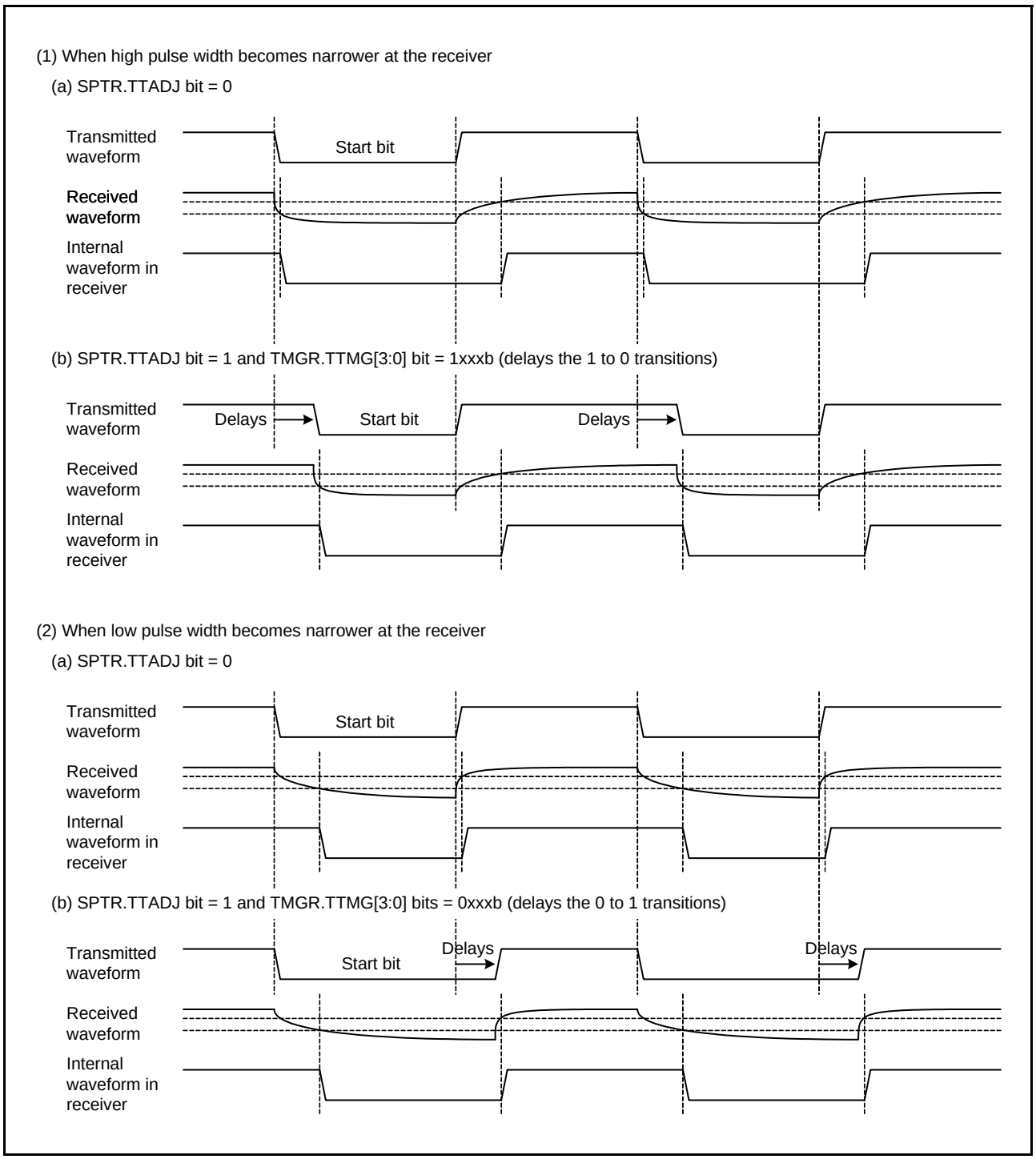


Figure 35.9 Example of Transition Timing Adjustment

35.3.3 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 SMR.CM bit and the SCR.CKE[1:0] bits.
When an external clock is input to the SCKn pin, the clock frequency should be 16 times the bit rate (when SEMR.ABCS bit = 0) and 8 times the bit rate (when SEMR.ABCS bit = 1). In addition, when an external clock is specified, the base clock of TMR0 and TMR1 can be selected by the SCIn.SEMR.ACS0 bit (n = 5, 6, 12).
When the SCI is operated on an internal clock, the clock can be output from the SCKn pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in Figure 35.10.

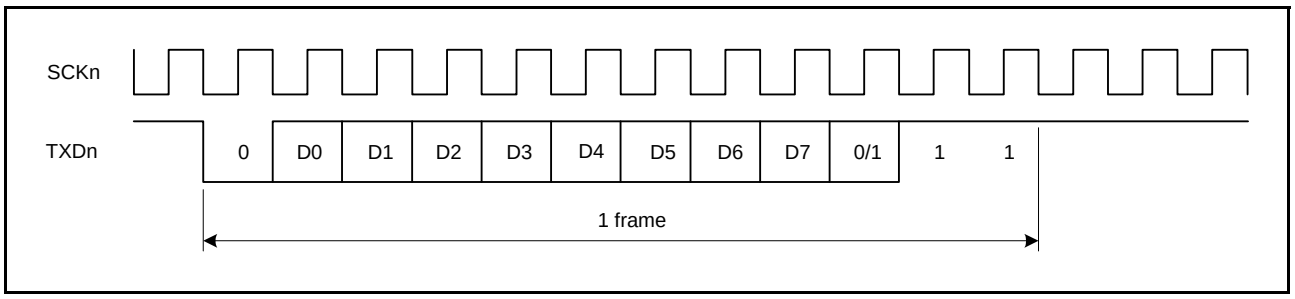


Figure 35.10 Phase Relationship between Output Clock and Transmit Data (Asynchronous Mode: SMR.CHR = 0, PE = 1, MP = 0, STOP = 1)

35.3.4 Double-Speed Mode and Divide-by-6 Mode

The output clock frequency of the on-chip baud rate generator is doubled by setting the SEMR.BGDM bit to 1, enabling high-speed communication at a doubled bit rate. If the SEMR.ABCS bit is set to 1 under the above condition, the number of base clock cycles changes from 16 to 8, so the bit rate becomes four times faster than the initial state.
When the SEMR.ABCSE bit is set to 1, the number of base clock pulses in 1-bit period becomes 6 and the period of the base clock becomes 1/2, where the bit rate becomes 16/3 times faster compared to a case that all of the ABCS, BDGM, and ABCSE bits are set to 0.
As shown by Formula (1) in section 35.3.2, Receive Data Sampling Timing and Reception Margin in Asynchronous Mode, setting the SEMR.ABCS bit to 1 changes the number of cycles to 8, and the sampling interval becomes longer. This causes the reception margin to decrease. Therefore, setting the SEMR.BGDM bit to 1 and the SEMR.ABCS bit to 0 is recommended instead of setting the SEMR.BGDM bit to 0 and the SEMR.ABCS bit to 1 for high-speed operation at a doubled bit rate.

35.3.5 CTS and RTS Functions

The CTS function is the use of input on the CTSn# pin in transmission control. Setting the SPMR.CTSE bit to 1 enables the CTS function. When the CTS function is enabled, placing the low level on the CTSn# pin causes transmission to start.
Applying the high level to the CTS# pin while transmission is in progress does not affect transmission of the current frame, which continues.
In the RTS function, by using the function of output on the RTSn# pin, a low level is output when reception becomes possible. Conditions for output of the low and high level are shown below.

(a) SCI0 to SCI9 and SCI12, and SCI10 and SCI11 when FIFO is Disabled

[Conditions for low-level output]
When the following conditions are all satisfied.

- The SCR.RE bit is 1.
- Reception is not in progress.
- There are no received data yet to be read.
- The ORER, FER, and PER flags in the SSR register are all 0.

[Condition for high-level output]
When the conditions for low-level output are not satisfied.

(b) SCI10 and SCI11 When FIFO is Enabled

[Conditions for low-level output]
When the following conditions are all satisfied.

- The SCR.RE bit is 1.
- The number of data stored in the receive FIFO is less than the threshold (FCR.RTRG[3:0]).
- The SSRFIFO.ORER flag is 0.

[Condition for high-level output]
When the conditions for low-level output are not satisfied.

Note that either one of CTS and RTS can be selected.

35.3.6 Data Match Detection

The data match detection function is available in asynchronous mode for SCI0 to SCI11.
When the DCCR.DCME bit is set to 1, the received data is compared with the contents of the CDR.CMPD[8:0] bits*1.
Upon a match of the value, a receive data full interrupt (RXI) request is generated.
When the SMR.MP bit is 0, all received data is compared.
When the SMR.MP bit is set to 1 and if the DCCR.IDSEL bit is 1, only the data in which the multiprocessor bit is 1 is compared and data in which the bit is 0 is ignored. When the DCCR.IDSEL bit is 0, all of the received data is compared regardless of the value of the multiprocessor bit.
The received data is not stored or the flag is not updated until the received data matches with the value of the CDR.CMPD[8:0] bits. When the data is matched, the DCCR.DCME bit is automatically set to 0 and the DCMF flag becomes 1. At this time, if the DCCR.IDSEL bit is 1, the SCR.MPIE bit is automatically set to 0. In addition, an receive data full interrupt (RXI) request is generated when the SCR.RIE bit is 1.
When a framing error is found in the matched data, the DCCR.DFER flag becomes 1. When a parity error is found in the matched data, the DCCR.DPER flag becomes 1. The received data matched with the value of the CDR.CMPD[8:0] bits is not stored in the received buffer or the SSR.RDRF flag (the SSRFIFO.RDF flag when the FCR.FM bit is 1) does not become 1.
After a data match is detected and the DCCR.DCME bit is set to 0, data is normally received.
When the DCCR.DFER or DCCR.DPER flag is 1, a data match is not detected. These flags should be set to 0 before enabling the data match detection function.

Note 1. Only the portion that corresponds to the character length specified by the SMR.CHR and SCMR.CHR1 bits is compared.

Figure 35.11 and Figure 35.12 show examples of data match detection.

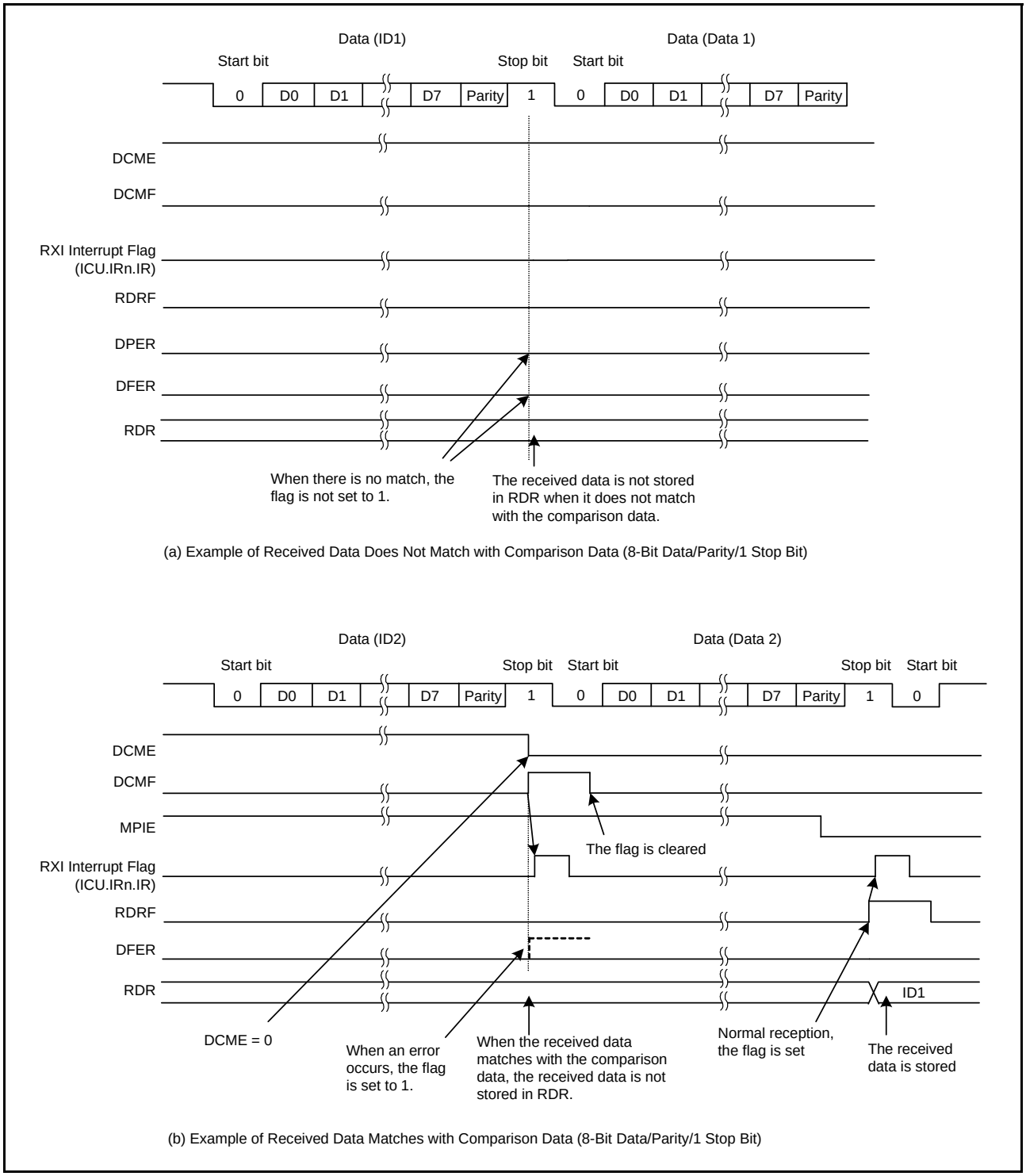


Figure 35.11 Example of Data Match Detection (1) Non Multi-Processor Mode

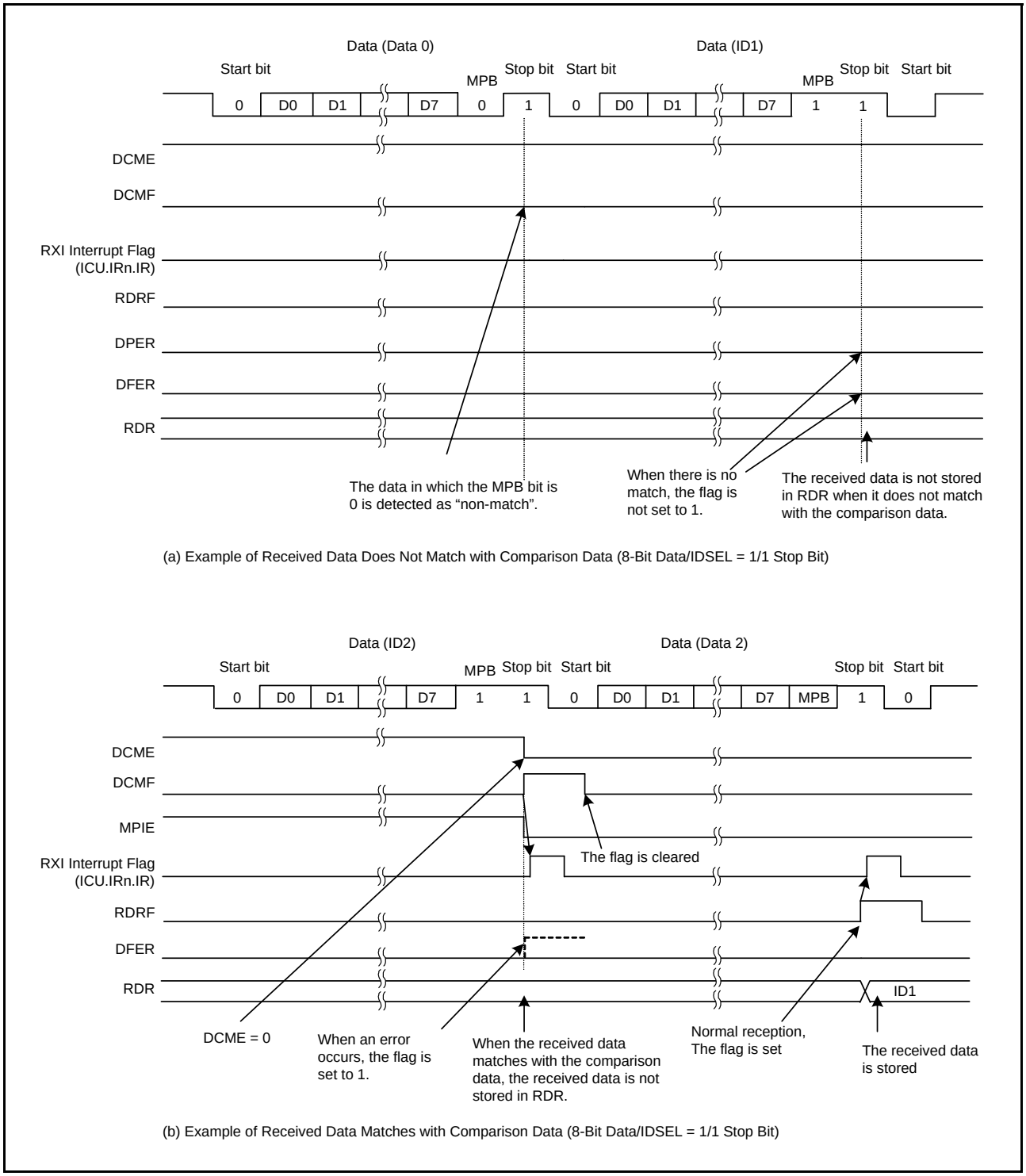


Figure 35.12 Example of Data Match Detection (2) Non Multi-Processor Mode

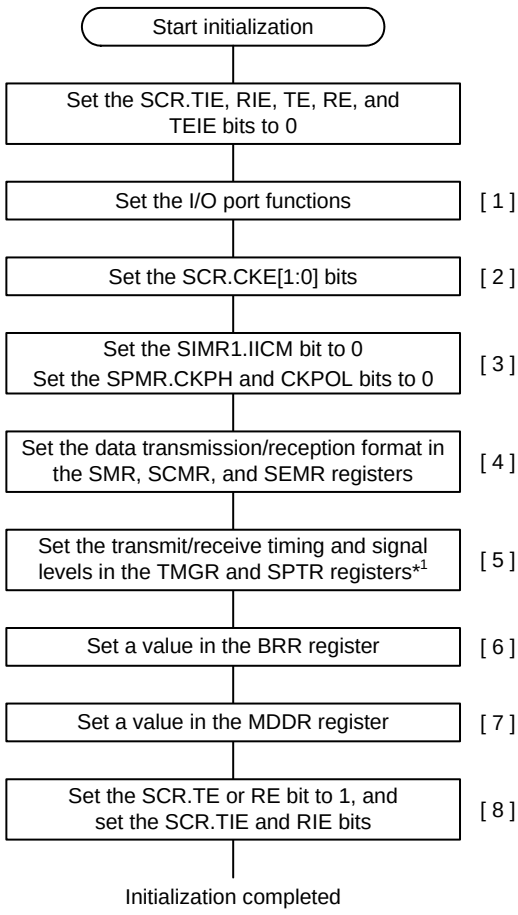
35.3.7 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, start by writing the initial value 00h to the SCR register and then continue through the procedure for SCI given in Figure 35.13 or Figure 35.14. Whenever the operating mode or transfer format is changed, the SCR register must be initialized before the change is made.

When the external clock is used in asynchronous mode, ensure that the clock signal is supplied even during initialization.

Note that setting the SCR.RE bit to 0 initializes neither the ORER, FER, PER, and RDRF flags in the SSR register nor registers RDR, RDRH, and RDRL. Note also that the SSRFIFO.TEND flag does not become 1 even when the SCR.TE bit is set to 0.

In addition, note that setting bits TIE, TE, and TEIE in the SCR register to 1 simultaneously leads to the generation of a transmit end interrupt (TEI) request before the generation of a transmit data empty interrupt (TXI) request.



[1] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[2] Set the clock selection in the SCR register.
When the clock output is selected in asynchronous mode, the clock is output immediately after the SCR register settings are made.

[3] Set the SIMR1.IICM bit to 0.
Set the SPMR.CKPH and CKPOL bits to 0.
This step can be skipped if the values have not been changed from the initial values.

[4] Set data transmission/reception format in the SMR, SCMR, and SEMR registers.

[5] Set the signal level on the TXD pin, signal polarity of transmit/receive signal, and data transmit/receive timing in the TMGR and SPTR registers*1.

[6] Write a value corresponding to the bit rate to the BRR register. This step is not necessary if an external clock is used.

[7] Write the value obtained by correcting a bit rate error in the MDDR register. This step is not necessary if the SEMR.BRME bit is set to 0 or an external clock is used.

[8] Set the SCR.TE or RE bit to 1. At this time, also set the SCR.TIE and RIE bits.
Setting the TE and RE bits allows TXDn and RXDn to be used.

Note 1. Only for SCI0 to SCI9.

Figure 35.13 Sample SCI Initialization Flowchart (Asynchronous Mode) (for SCI0 to SCI9 and SCI12)

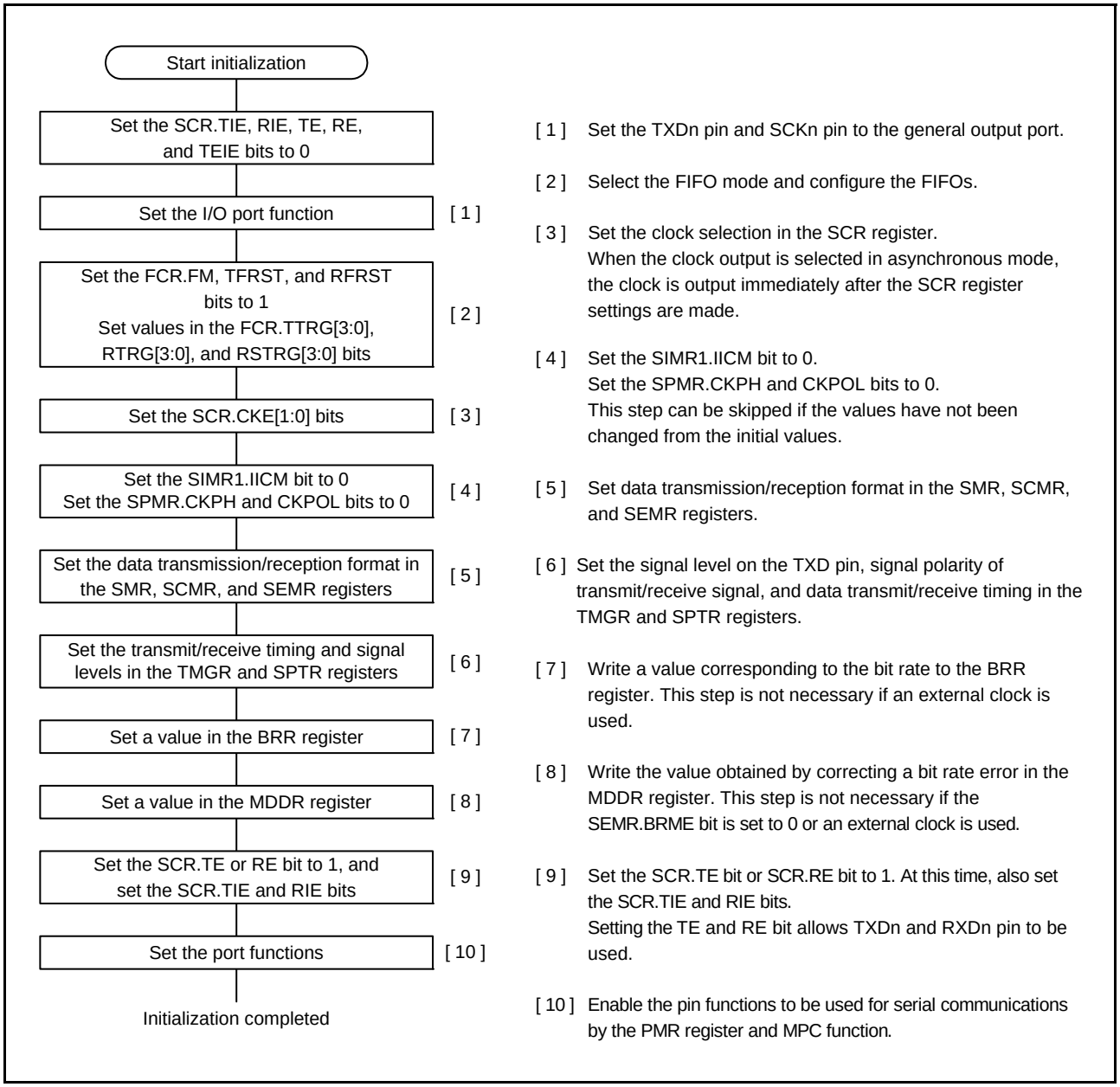


Figure 35.14 Sample SCI Initialization Flowchart (Asynchronous Mode, FIFO enabled) (for SCI10 and SCI11)

Figure 35.15 shows an example of data transmission when the SCI is set to asynchronous mode according to the flow described in Figure 35.13 after a reset. When the pin function is set to the TXD pin, it is still high-impedance because the SCR.TE bit is 0. When the transmit data is written after setting the TE bit to 1, a data transmission starts. After the TE bit is set to 1, one frame of high is output from TXD pin (internal wait time)*1 and then the data transmission starts.

Note 1. This is when the SEMR.ITE bit is 0. When the ITE bit is 1, the internal wait time is not inserted.

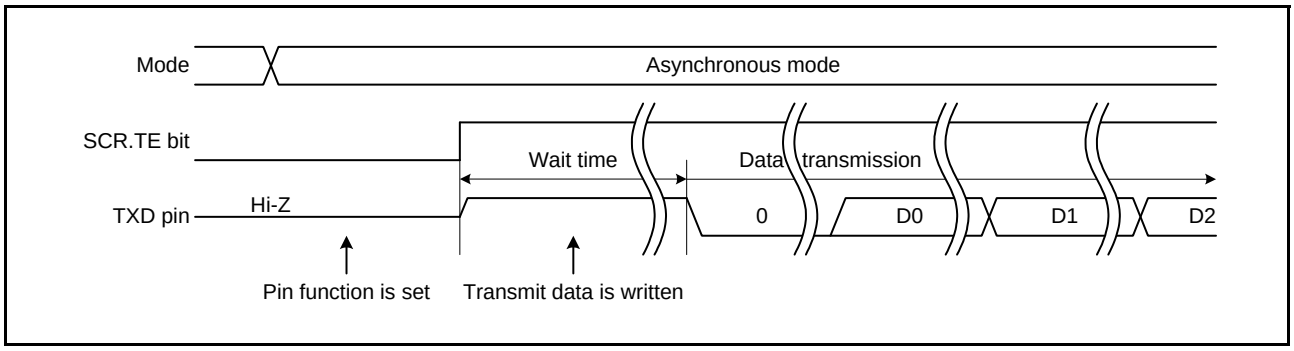


Figure 35.15 Example of Data Transmission Timing in Asynchronous Mode

35.3.8 Serial Data Transmission (Asynchronous Mode)

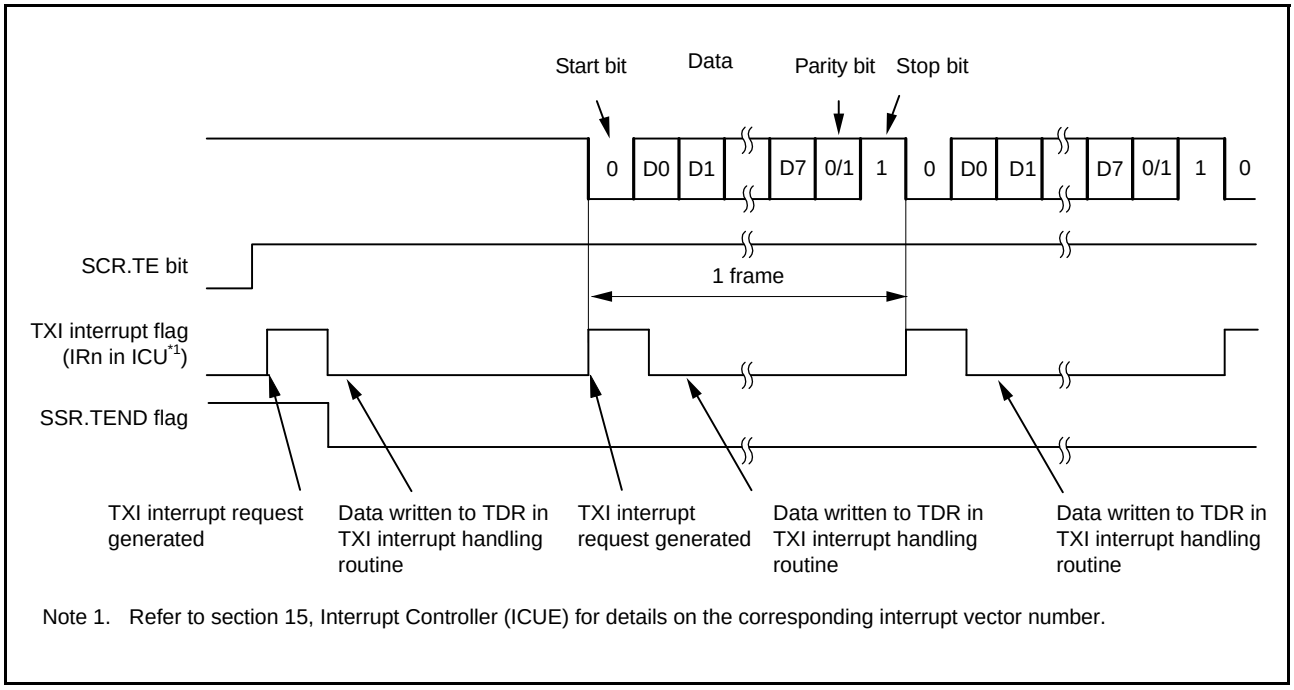
(1) SCI0 to SCI9 and SCI12, and SCI10 and SCI11 when FIFO is Disabled

Figure 35.16 to Figure 35.18 show an example of the operation for serial transmission in asynchronous mode.
In serial transmission, the SCI operates as described below.

1. The SCI transfers data from the TDR register*1 to the TSR register when data is written to the TDR register*1 in the TXI interrupt handling routine. The TXI interrupt request at the beginning of transmission is generated when the SCR.TE bit is set to 1 after the SCR.TIE bit is set to 1 or when these 2 bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the SPMR.CTSE bit is set to 0 (CTS function is disabled) and a low level on the CTSn# pin causes data transfer from the TDR register*1 to the TSR register. If the SCR.TIE bit is 1 at this time, a TXI interrupt request is generated. Continuous transmission is obtainable by writing the next transmit data to the TDR register*1 in the TXI interrupt handling routine before transmission of the current transmit data is completed. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (a TXI interrupt request is disabled) and the SCR.TEIE bit to 1 (a TEI interrupt request is enabled) after the last of the data to be transmitted are written to the TDR register*1, *2 from the handling routine for TXI requests.
3. Data is sent from the TXDn pin in the following order: start bit, transmit data, parity bit or multi-processor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks for updating of (writing to) the TDR register*3 at the time of stop bit output.
5. When the TDR register*3 is updated, setting of the SPMR.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn# pin cause the next transfer of the next transmit data from the TDR register*1 to the TSR register and sending of the stop bit, after which serial transmission of the next frame starts.
6. If the TDR register*3 is not updated, the SCI sets the SSR.TEND flag to 1, sends the stop bit, and then outputs high to put the line in the mark state. If the SCR.TEIE bit is 1 at this time, the SSR.TEND flag is set to 1 and a TEI interrupt request is generated.

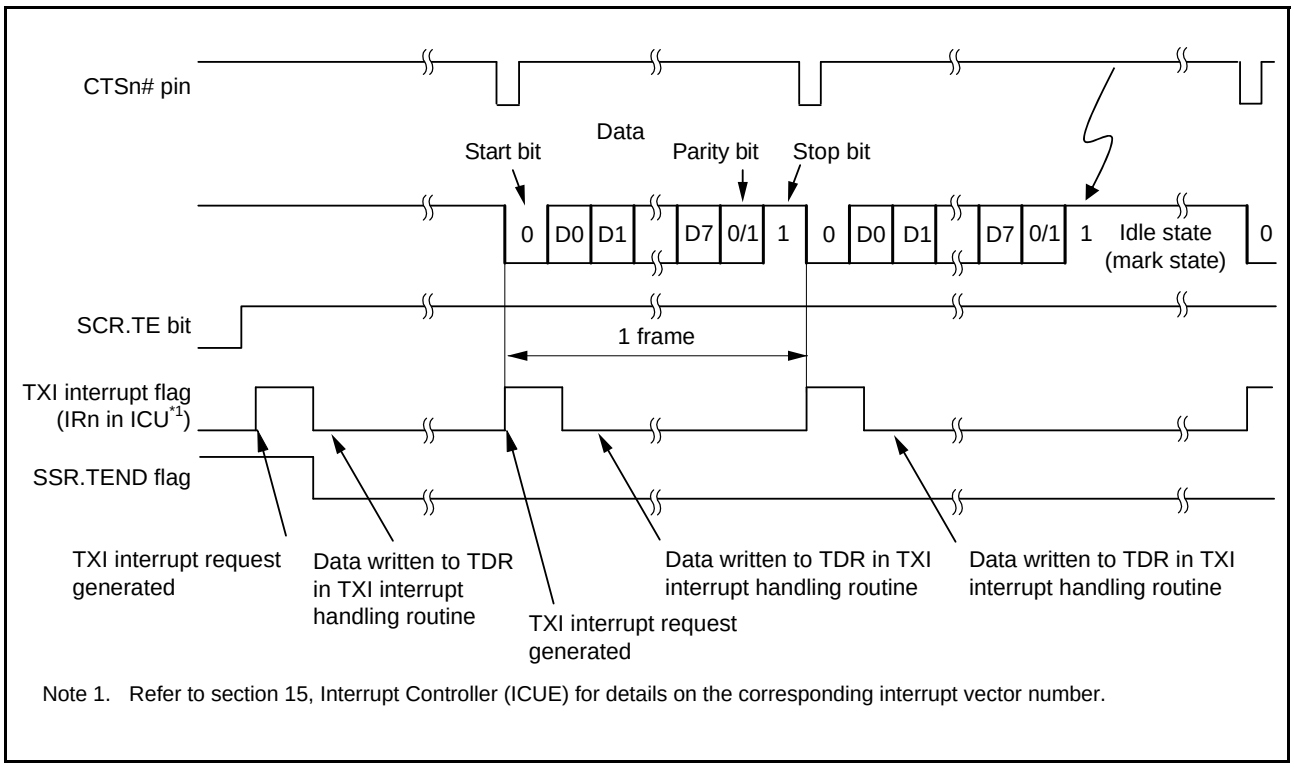
Note 1. Write data not to the TDR register but to the TDRH and TDRL registers when 9-bit data length is selected.
Note 2. Write data in the order from the TDRH register to the TDRL register when 9-bit data length is selected.
Note 3. The SCI checks for updating of the TDRL register only and does not check for updating of the TDRH register when 9-bit data length is selected.

Figure 35.19 shows a sample flowchart for serial transmission in asynchronous mode.



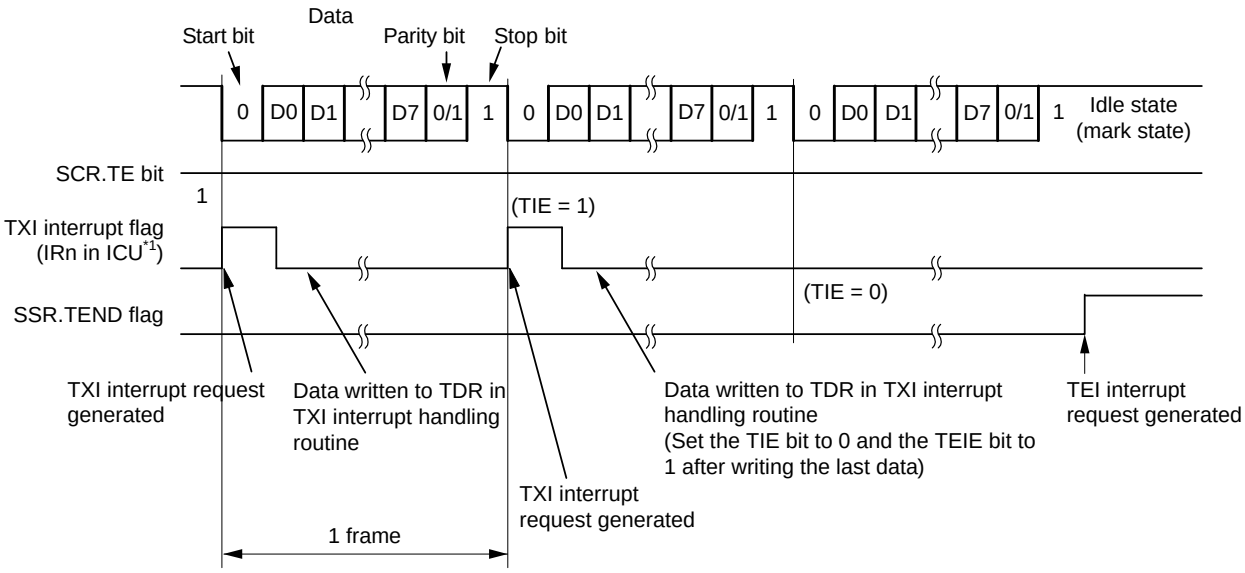
Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 35.16 Example of Operation for Serial Transmission in Asynchronous Mode (1) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, at the Beginning of Transmission)



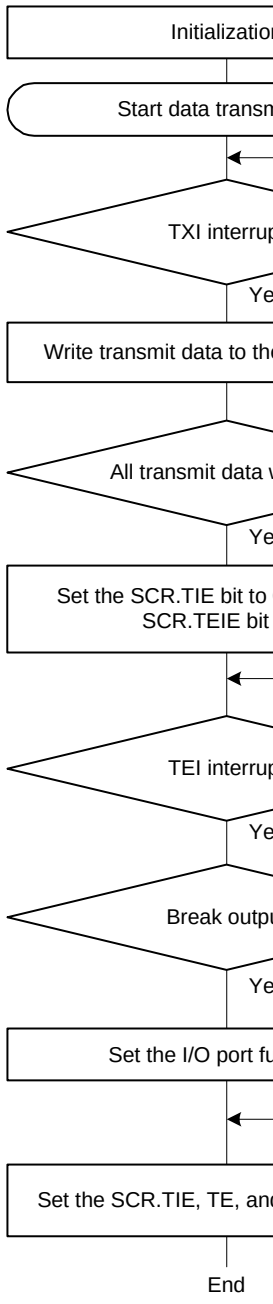
Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 35.17 Example of Operation for Serial Transmission in Asynchronous Mode (2) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Used, at the Beginning of Transmission)



Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 35.18 Example of Operation for Serial Transmission in Asynchronous Mode (3) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, from the Middle of Transmission until Transmission Completion)



[1] SCI initialization:
Set data transmission.
After the SCR.TE bit is set to 1, high is output for a frame, and transmission is enabled.

[2] Transmit data write to the TDR register by a TXI interrupt request:
When transmit data is transferred from the TDR register to the TSR register, a transmit data empty interrupt (TXI) request is generated.
Write transmit data to the TDR register once in the TXI interrupt handling routine.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to the TDR register once using a TXI interrupt request.
Transmit data can also be written to the TDR register by activating the DMAC or DTC.
When TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted are written to the TDR register.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set the I/O port function corresponding to the TXDn pin (corresponding to output of the low level), and after switching the TXDn pin to the general I/O port function, set the SCR.TE bit to 0.

Note: The TDR register becomes the TDRH and TDRL registers when 9-bit data length is selected. Write data in the order from the TDRH register to the TDRL register.

Figure 35.19 Example of Serial Transmission Flowchart in Asynchronous Mode

(2) SCI10 and SCI11 When FIFO is Enabled

The TXI interrupt request at the beginning of transmission is generated when the SCR.TE and SCR.TIE bits are set to 1 simultaneously.

Set the transmit data to the FTDR register (the FTDR.L register for 7-bit and 8-bit) instead of the TDR register. When data is transferred from the transmit FIFO to the TSR register and if the number of data stored in the transmit FIFO is less than or equal to the threshold (FCR.TTRG[3:0] bits), a transmit data empty interrupt (TXI) request is generated.

Up to 16 minus FDR.T[4:0] frames of transmit data can be set within the TXI interrupt processing routine. When the settings of all transmit data are completed, set the SSRFIFO.TDFE flag to 0.

When the transmit data is set by using the DMAC or DTC, the TDFE flag is automatically set to 0.

When sending a break, use the SPB2IO and SPB2DT bits in the SPTR register. Upon completion of the setting, a break is sent by setting the SCR.TE bit to 0.

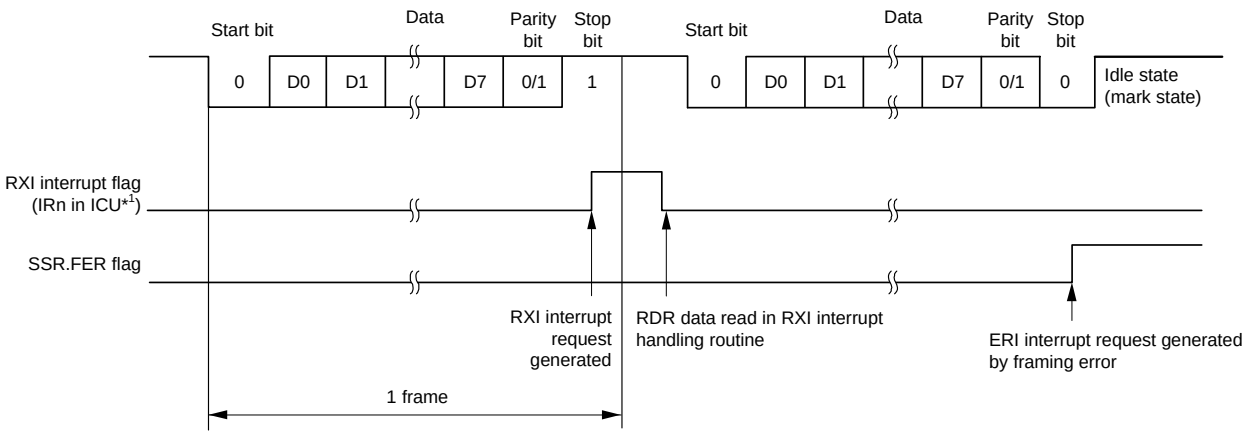
35.3.9 Serial Data Reception (Asynchronous Mode)

(1) SCI0 to SCI9 and SCI12, and SCI10 and SCI11 when FIFO is Disabled

Figure 35.20 and Figure 35.21 show an example of the operation for serial data reception in asynchronous mode. In serial data reception, the SCI operates as described below.

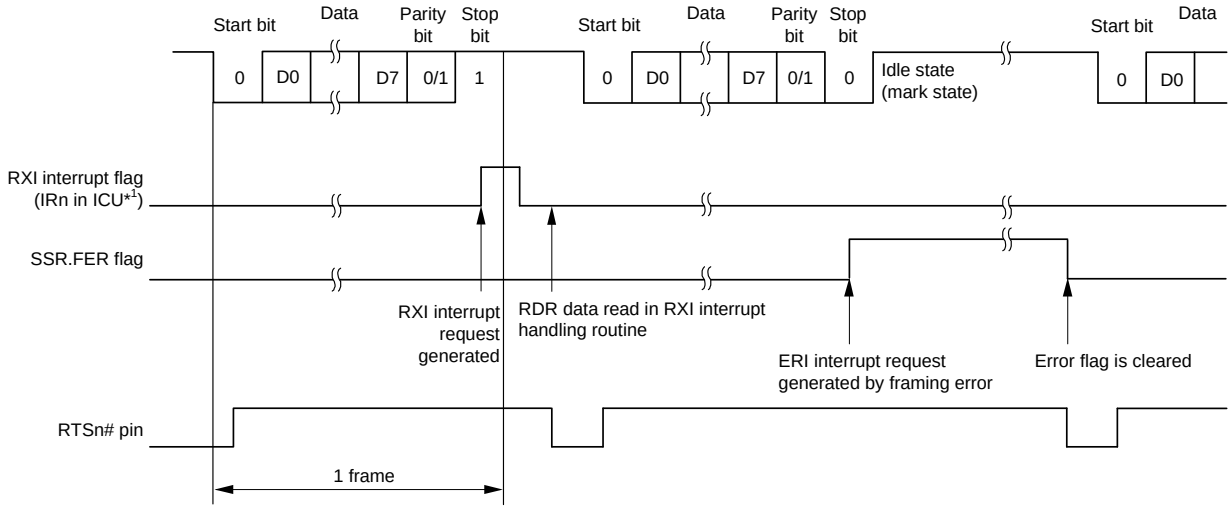
1. When the value of the SCR.RE bit becomes 1, the output signal on the RTSn# pin goes to the low level.
2. When the SCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in the RSR register, and checks the parity bit and stop bit.
3. If an overrun error occurs, the SSR.ORER flag is set to 1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to the RDR register*1.
4. If a parity error is detected, the SSR.PER flag is set to 1 and receive data is transferred to the RDR register*1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated.
5. If a framing error (when the stop bit is 0) is detected, the SSR.FER flag is set to 1 and receive data is transferred to the RDR register*1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated.
6. When reception finishes successfully, receive data is transferred to the RDR register*1. If the SCR.RIE bit is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the RDR register*1 in this RXI interrupt handling routine before reception of the next receive data is completed. Reading the received data that have been transferred to the RDR register*1 causes the RTSn# pin to output the low level.

Note 1. Read data not in the RDR register but in the RDRH and RDRL registers when 9-bit data length is selected.
Note 2. The SCI checks for reading of the RDRL register only and does not check for reading of the RDRH register when 9-bit data length is selected.



Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 35.20 Example of SCI Operation for Serial Reception in Asynchronous Mode (1) (When RTS Function is Not Used) (Example with 8-Bit Data, Parity, 1 Stop Bit)



Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 35.21 Example of SCI Operation for Serial Reception in Asynchronous Mode (2) (When RTS Function is Used) (Example with 8-Bit Data, Parity, 1 Stop Bit)

Table 35.35 lists the states of the status flags in the SSR register and receive data handling when a receive error is detected.

If a receive error is detected, an ERI interrupt request is generated but an RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, FER, and PER flags to 0 before resuming reception. Moreover, be sure to read the RDR (or the RDRL) register during overrun error processing.

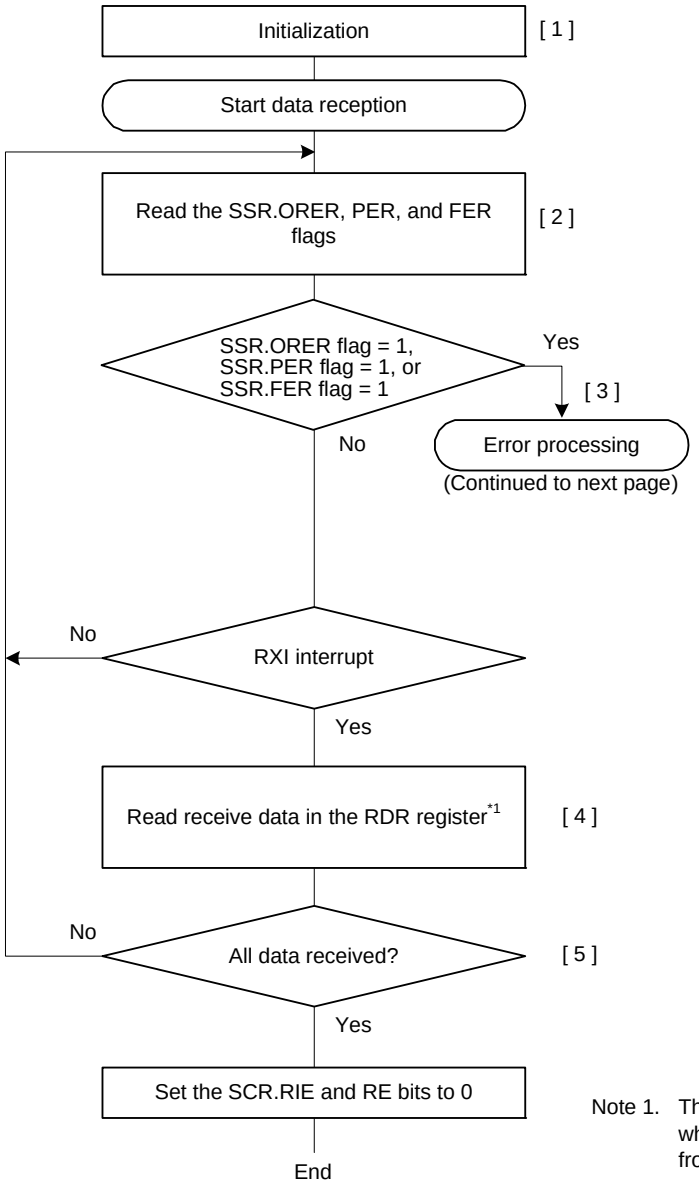
When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read the RDR (or the RDRL) register because received data which has not yet been read may be left in the RDR (or the RDRL) register.

Figure 35.22 and Figure 35.23 show samples of flowcharts for serial data reception.

Table 35.35 Status Flags in the SSR Register and Receive Data Handling

Status Flags in the SSR Register				
ORER	**FER**	**PER**	**Receive Data**	**Receive Error Type**
1	0	0	Lost	Overrun error
0	1	0	Transferred to the RDR register*1	Framing error
0	0	1	Transferred to the RDR register*1	Parity error
1	1	0	Lost	Overrun error + framing error
1	0	1	Lost	Overrun error + parity error
0	1	1	Transferred to the RDR register*1	Framing error + parity error
1	1	1	Lost	Overrun error + framing error + parity error

Note 1. Read data not in the RDR register but in the RDRH and RDRL registers when 9-bit data length is selected.



[1] SCI initialization:
Set data reception.

[2] [3] Receive error processing and break detection:
If a receive error occurs, an ERI interrupt is generated. An error is identified by reading the ORER, PER, and FER flags in the SSR register. After performing the appropriate error processing, be sure to set the ORER, PER, and FER flags to 0. Reception cannot be resumed if any of these flags is set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RXDn pin.

[4] Read the receive data in the RDR register once in the RXI interrupt handling routine.

[5] Serial reception continuation procedure:
To continue serial reception, before the stop bit of the current frame is received, read data from the RDR register in the RXI interrupt handling routine.
The RDR data can also be read by activating the DMAC or DTC.

Note 1. The RDR register becomes the RDRH and RDRL registers when 9-bit data length is selected. Read data in the order from the RDRH register to the RDRL register.

Figure 35.22 Example Flowchart of Serial Reception in Asynchronous Mode (1)

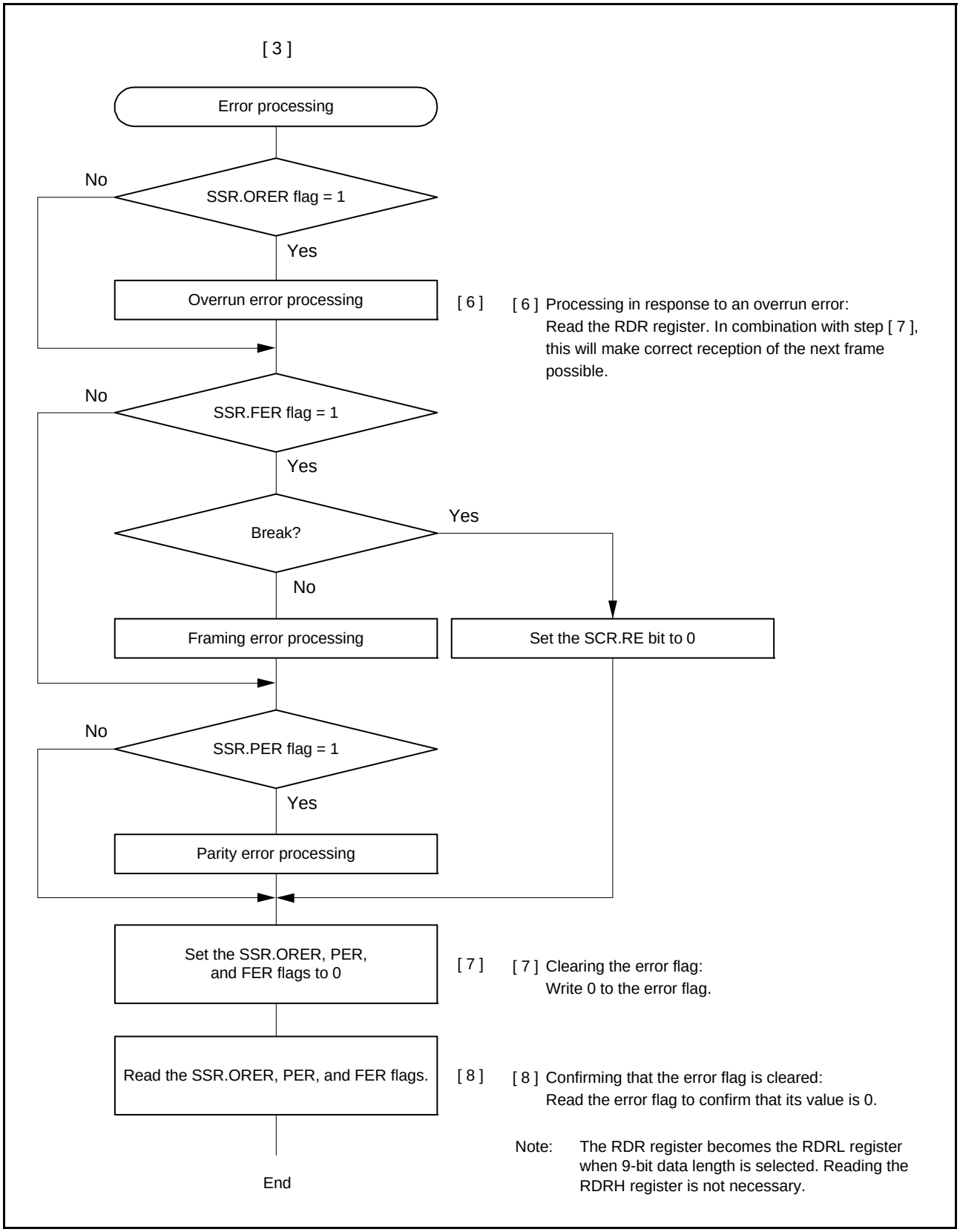


Figure 35.23 Example Flowchart of Serial Reception in Asynchronous Mode (2)

(2) SCI10 and SCI11 When FIFO is Enabled

The received data and status flag are read from the FRDR register instead of the RDR register. When reading in byte units, read the FRDR.H register and then the FRDR.L register in this order. Reading the FRDR.L register updates the FER, PER, and RDAT[8:0] bits in the FRDR register. The value of the RDF, ORER, and DR flags in the FRDR register is the same as that of the SSRFIFO register.
When receiving serial data, the SCI operates as follows.

1. If the SCR.RE bit is set to 1 and the RTSn# function is in use, the SCI drives the RTSn# pin low.
2. When the SCI monitors the communications line and detects a start bit, it performs internal synchronization, stores the received data to the RSR register, and checks the parity and stop bits.
3. When there is no space in the receive FIFO, an overrun error occurs. When an overrun error occurs, the SCI sets the SSRFIFO.ORER flag to 1. At this time, if the SCR.RIE bit is 1, an ERI interrupt request is generated. The received data is not transferred to the receive FIFO.
4. When a parity error is detected, the SCI transfers the received data to the receive FIFO and sets the PER flag in the receive FIFO to 1. At this time, if the SCR.RIE bit is 1, an ERI interrupt request is generated.
5. When a framing error (the stop bit is 0) is detected, the SCI transfers the received data to the receive FIFO and sets the FER flag in the receive FIFO to 1. At this time, if the SCR.RIE bit is 1, an ERI interrupt request is generated.
6. When a framing error is detected and the following one-frame data received is all 0s, the SCI stops receiving operation.
7. If the number of data stored in the receive FIFO is less than the threshold (FCR.RTRG[3:0] bits) and reception of the next frame is not completed even after 15 etus have elapsed from the stop bit of the previously received frame, the SSRFIFO.DR flag is set to 1. At this time, if the SCR.RIE bit is 1, an RXI interrupt request (when the FCR.DRES bit is 0) or an ERI interrupt request (when the FCR.DRES bit is 1) is generated.
8. When the frame is successfully received, the SCI transfers the received data to the receive FIFO. If the number of data stored in the receive FIFO is equal to or greater than the threshold (FCR.RTRG[3:0] bits), the SCI sets the SSRFIFO.RDF flag to 1. At this time, if the SCR.RIE bit is 1, an RXI interrupt request is generated. Using the RXI interrupt processing routine allows reading of the received data in the RDRF register before an overrun error occurs, thus enabling the sequence reception of data. When the received data that have been transferred to the receive FIFO are read and the number of unread data is less than the value of the FCR.RSTRG[3:0] bits, the SCI drives the RTSn# pin low.

35.4 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 35.24 shows an example of communication between processors by using a multi-processor format. First, a transmitting station transmits communication data in which the multi-processor bit set to 1 is added to the ID code of the receiving station. Next, the transmitting station transmits the communication data in which the multi-processor bit set to 0 is added to the transmit data. Upon receiving the communication data in which the multi-processor bit is set to 1, the receiving station compares the received ID with the ID of the receiving station itself and if the two match, receives the communication data that is subsequently transmitted. If the received ID does not match with the ID of the receiving station, the receiving station skips the communication data until again receiving the communication data in which the multi-processor bit is set to 1.

(1) SCI0 to SCI9 and SCI12, and SCI10 and SCI11 when FIFO is Disabled

For supporting this function, the SCI provides the SCR.MPIE bit. When the MPIE bit is set to 1, transfer of receive data from the RSR register to the RDR register (the RDRH and RDRL registers when 9-bit data length is selected), detection of a receive error, and setting the respective status flags RDRF, ORER, and FER in the SSR register are disabled until reception of data in which the multi-processor bit is set to 1. Upon receiving a reception character in which the multi-processor bit is set to 1, the SSR.MPB bit is set to 1 and the SCR.MPIE bit is automatically cleared, thus returning to a normal reception operation. During this time, an RXI interrupt is generated if the SCR.RIE bit is 1.
When the multi-processor format is specified, specification of the parity bit is disabled. Apart from this, there is no difference from the operation in the normal asynchronous mode. A clock which is used for the multi-processor communication is also the same as the clock used in the normal asynchronous mode.

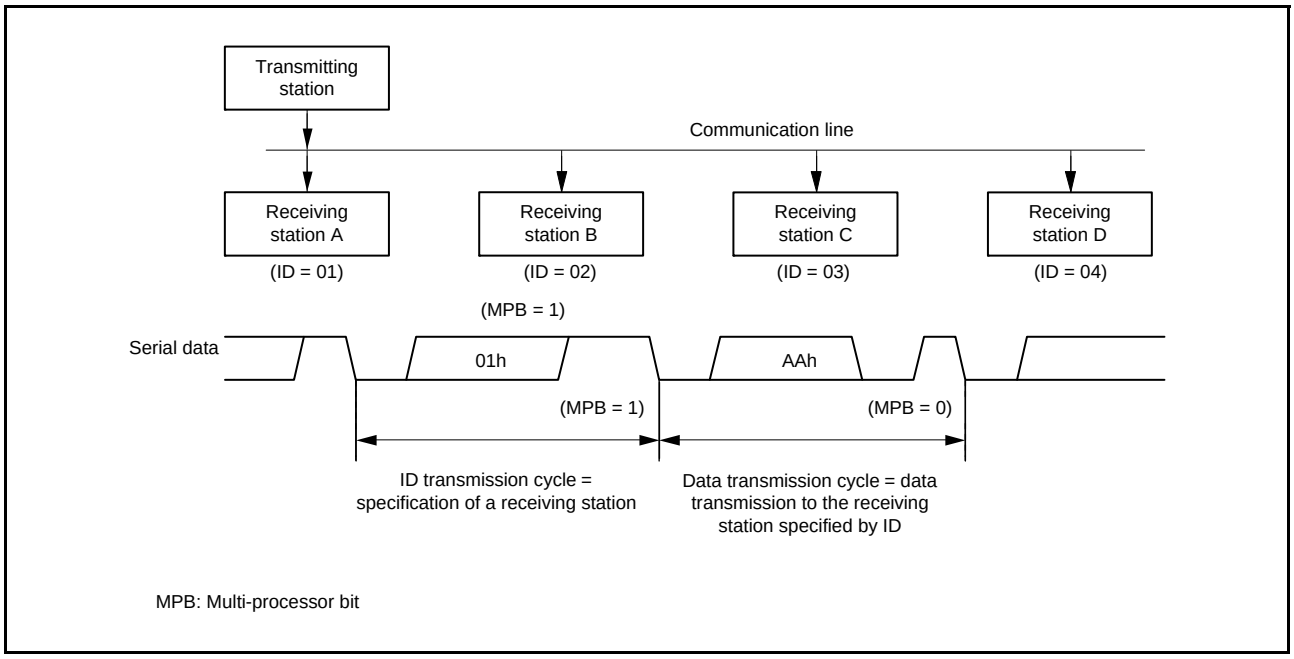


Figure 35.24 An Example of Communication using the Multi-Processor Format (Example of Transmission of Data AAh to Receiving Station A)

(2) SCI10 and SCI11 When FIFO is Enabled

In transmitting data, the SCI uses the FTDR.MPBT bit instead of the SSR.MPBT bit. Set a value and the transmit data to the FTDR register at the same timing.

In receiving data, the SCI uses the FRDR.MPB flag instead of the SSR.MPB flag. Upon completion of data reception, when the data in the RSR register is stored to the receive FIFO, the value of the multi-processor bit is stored at the same time.

Setting the SCR.MPIE bit to 1 disables transfer of the received data from the RSR register to the receive FIFO, detection of receive errors, and settings of the RDF, ORER, or FER flag in the SSRFIFO register until the data in which the multi-processor bit has the value 1 is received.

When the reception character in which the multi-processor bit has the value 1 is received, the multi-processor bit and received data are stored in the receive FIFO and the SCR.MPIE bit is set to 0, and then the SCI returns to the normal reception operation. At this time, if the SCR.RIE bit is 1, an RXI interrupt is generated.

When the multi-processor format is specified, setting of the parity bit is disables. Specifying a format other than that, the serial transfer format is the same as that in normal asynchronous mode.

35.4.1 Multi-Processor Serial Data Transmission

Figure 35.25 is a sample flowchart of multi-processor data transmission. In the ID transmission cycle, the ID should be transmitted with the SSR.MPBT bit set to 1. In the data transmission cycle, the data should be transmitted with the MPBT bit set to 0. The other operations are the same as the operations in asynchronous mode.

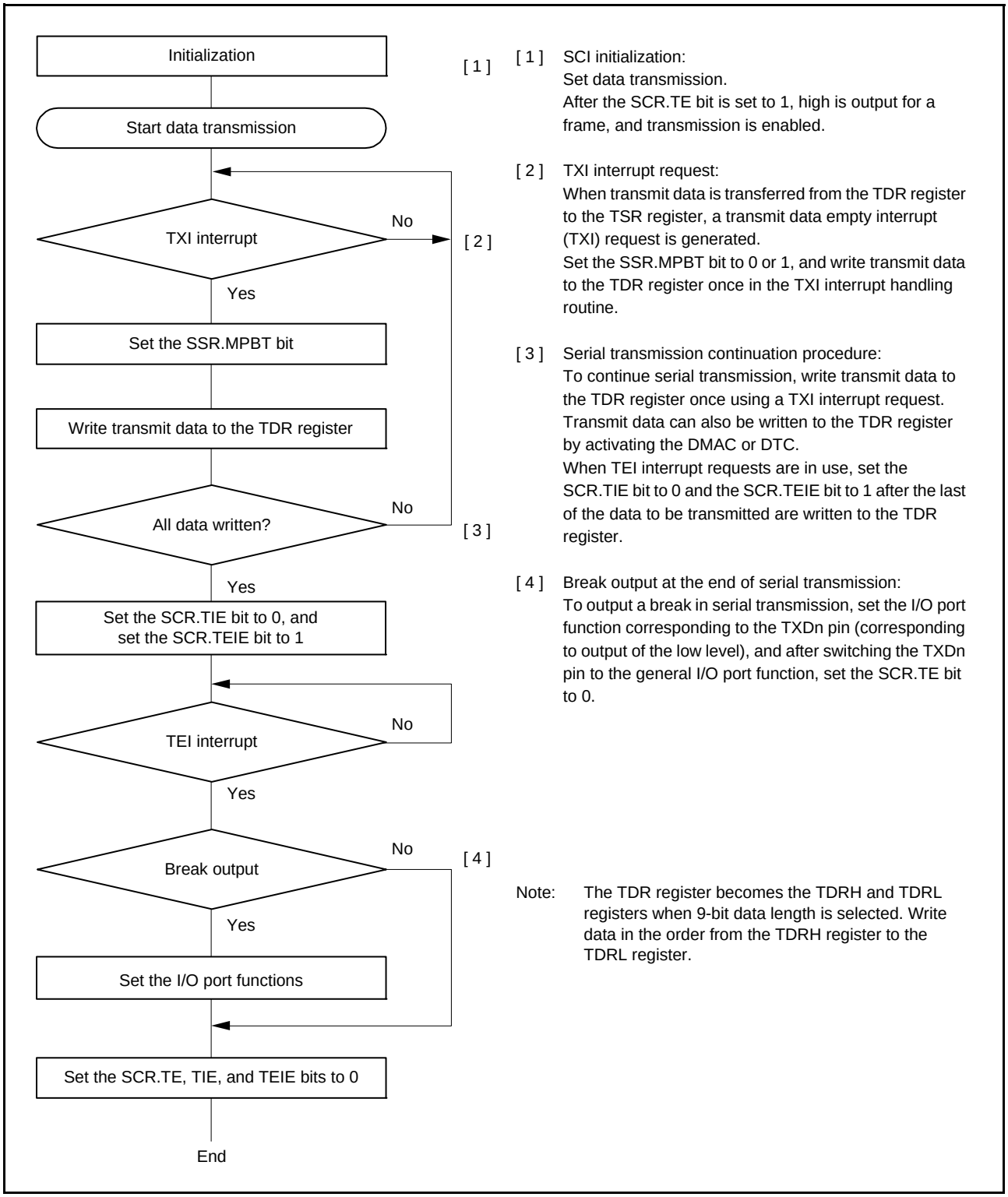
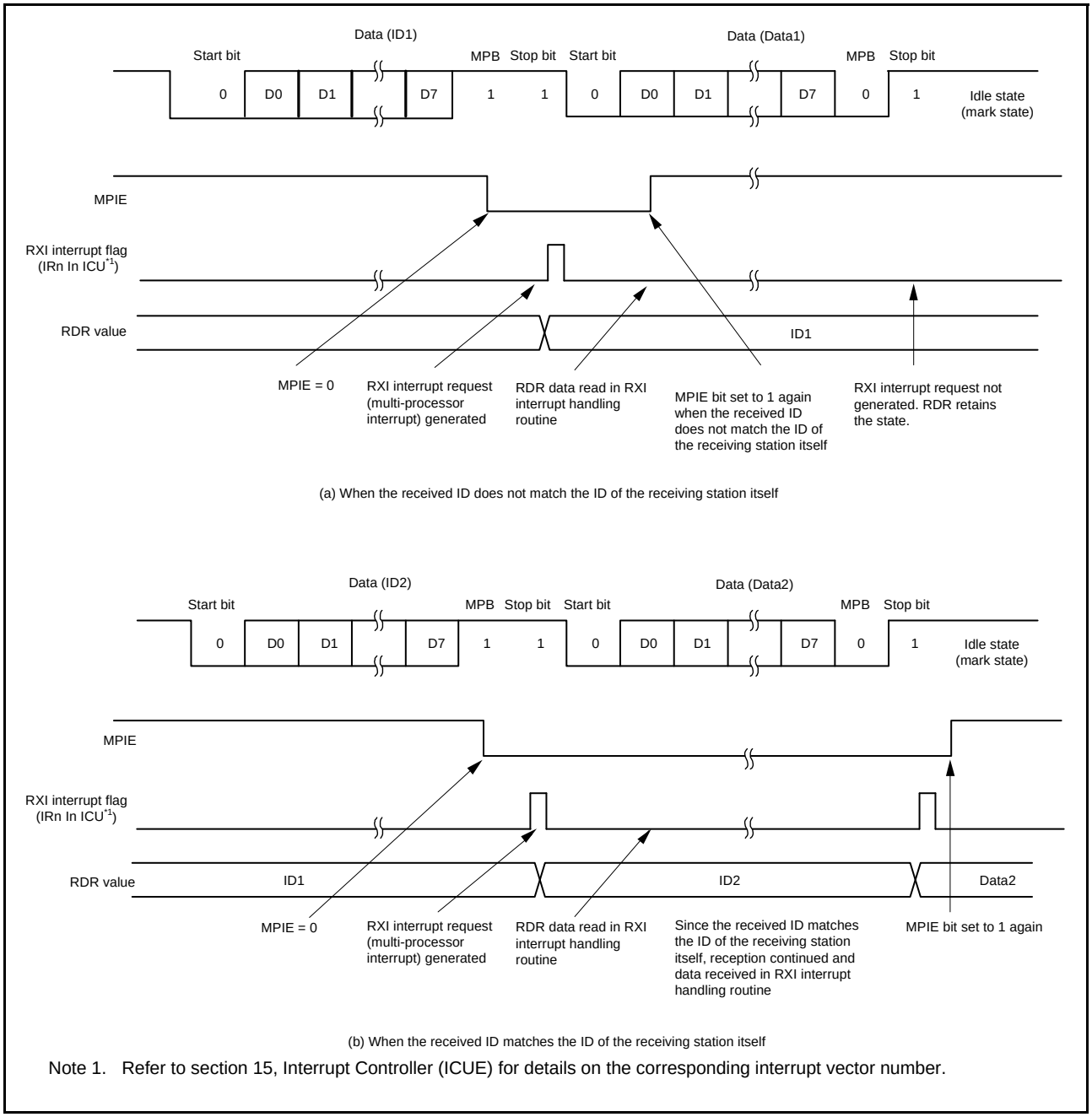


Figure 35.25 Example of Multi-Processor Serial Transmission Flowchart

35.4.2 Multi-Processor Serial Data Reception

Figure 35.27 and Figure 35.28 are sample flowcharts of multi-processor data reception. When the SCR.MPIE bit is set to 1, reading the communication data is skipped until reception of the communication data in which the multi-processor bit is set to 1. When the communication data in which the multi-processor bit is set to 1 is received, the received data is transferred to the RDR register (the RDRH and RDRL registers when 9-bit data length is selected). During this time, the RXI interrupt request is generated. The other operations are the same as the operations in asynchronous mode. Figure 35.26 is the example of operation for reception.



Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 35.26 Example of SCI Reception (8-Bit Data/Multi-Processor Bit/1 Stop Bit)

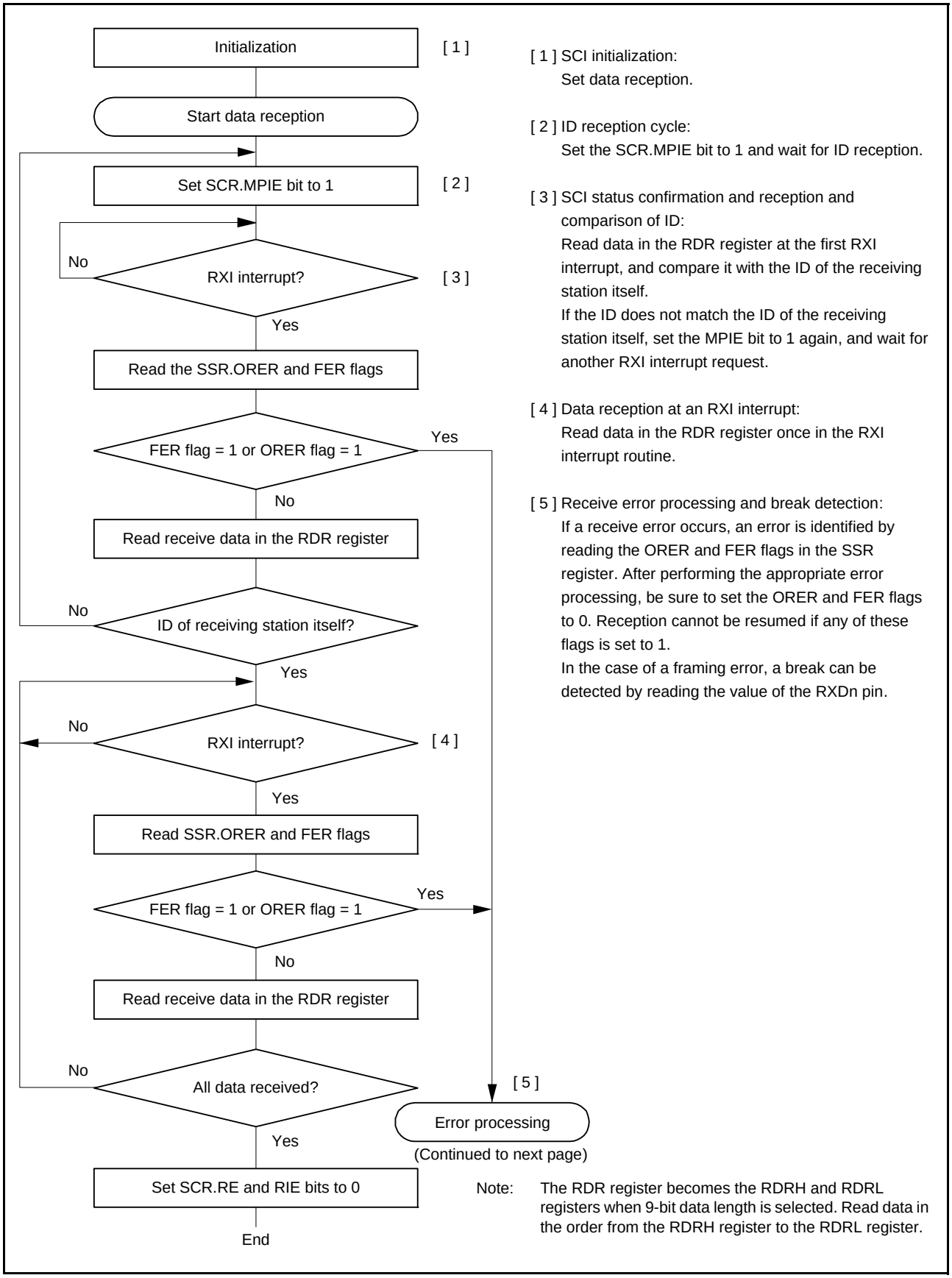


Figure 35.27 Example of Multi-Processor Serial Reception Flowchart (1)

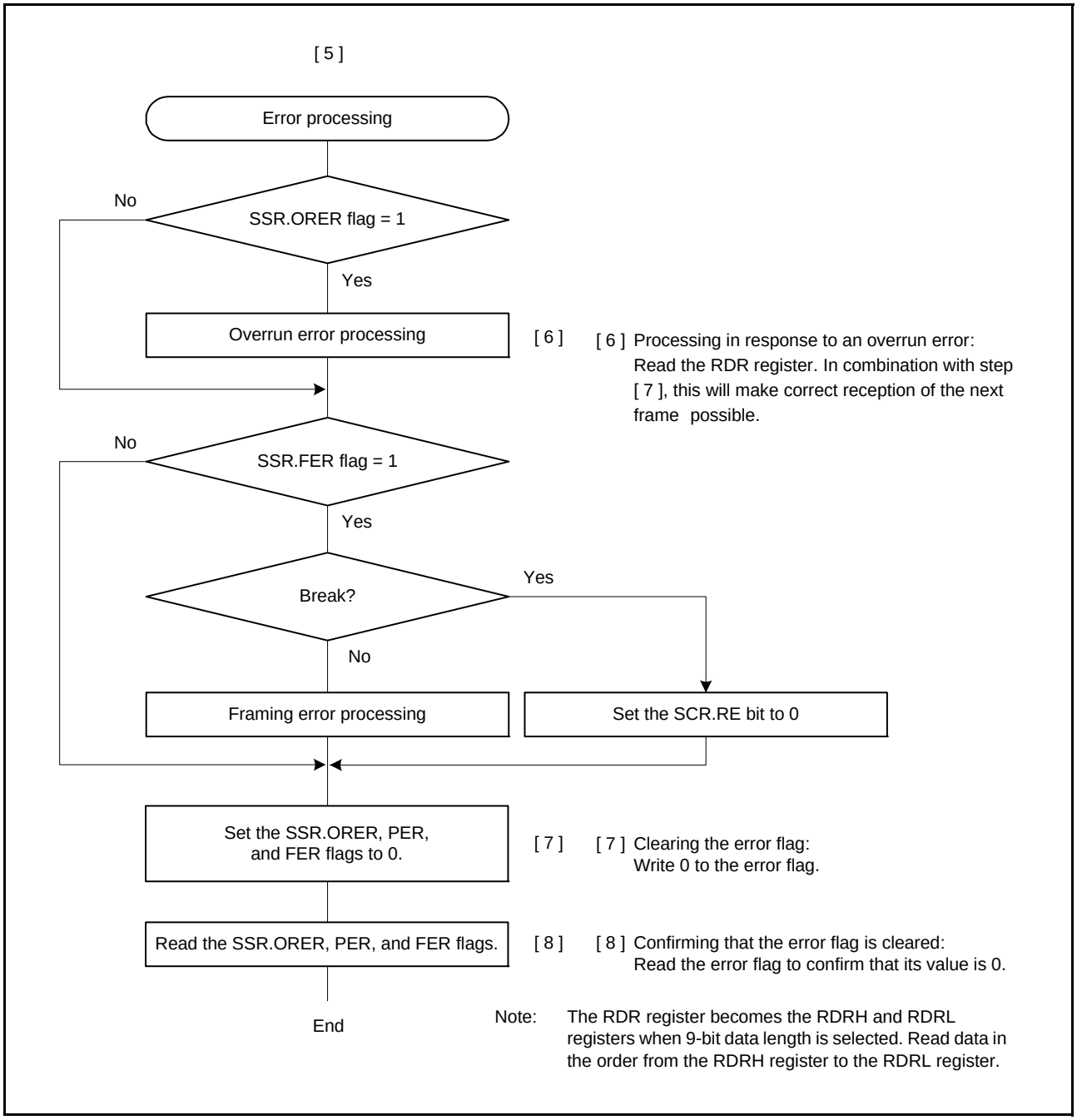


Figure 35.28 Example of Multi-Processor Serial Reception Flowchart (2)

35.5 Operation in Clock Synchronous Mode

Figure 35.29 shows the data format for clock synchronous serial data communications.
In clock synchronous mode, data is transmitted or received in synchronization with clock pulses. One character in transfer data consists of 8-bit data. In clock synchronous mode, no parity bit can be added.
In data transmission, the SCI outputs data from one falling edge of the synchronization clock to the next. In data reception, the SCI receives data in synchronization with the rising edge of the synchronization clock. After 8-bit data is output, the communication line holds the last bit output state.
Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications by use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so that the next transmit data can be written during transmission or the previous receive data can be read during reception, enabling continuous data transfer.

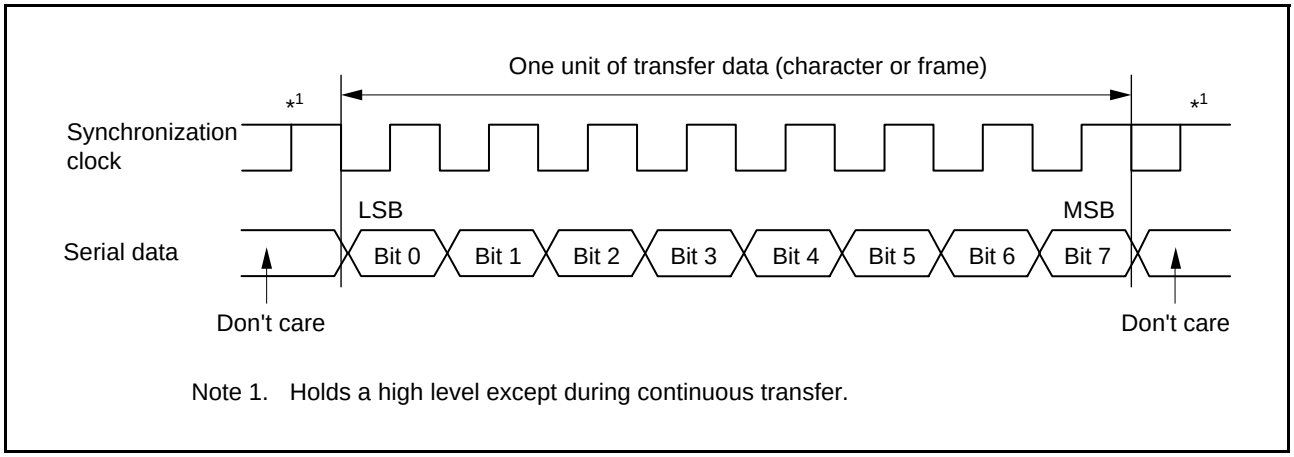


Figure 35.29 Data Format in Clock Synchronous Serial Communications (LSB First)

35.5.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCKn pin can be selected, according to the setting of the SCR.CKE[1:0] bits.
When the SCI is operated on an internal clock, the synchronization clock is output from the SCKn pin. Eight synchronization clock pulses are output in the transfer of one character, and when no transfer is performed the clock is held high. However, when only data reception is performed while the CTS function is disabled, the synchronization clock output is started at the same time when the SCR.RE bit set to 1. The synchronization clock is stopped at the high level when an overrun error occurs or the SCR.RE bit is set to 0.
When only data reception is performed and the CTS function is enabled, the clock output is not started even when the SCR.RE bit set to 1 if the CTSn# pin input is high when the SCR.RE bit is 0. The synchronization clock output is started when the SCR.RE bit is set to 1 and the CTSn# pin input is low. After that, if the CTSn# pin input is high on completion of the frame reception, the synchronization clock output is stopped at the high level. If the CTSn# pin input continues to be low, the synchronization clock is stopped at the high level when an overrun error occurs or the SCR.RE bit is set to 0.

35.5.2 CTS and RTS Functions

In the CTS function, CTSn# pin input is used to control reception/transmission start when the clock source is the internal clock. Setting the SPMR.CTSE bit to 1 enables the CTS function.
When the CTS function is enabled, placing the low level on the CTSn# pin causes reception/transmission to start. Applying the high level to the CTS# pin while reception/transmission are in progress does not affect reception/transmission of the current frame, which continues.
In the RTS function, RTSn# pin output is used to request reception/transmission start when the clock source is an external synchronizing clock. A low level is output when serial communications become possible. Conditions for output of the low and high level are shown below.

(a) SCI0 to SCI9 and SCI12, and SCI10 and SCI11 when FIFO is Disabled

[Conditions for low-level output]
When the following conditions are all satisfied.

- The SCR.RE or SCR.TE bit is 1.
- Transmission or reception is not in progress.
- There are no received data yet to be read (when the SCR.RE bit is 1).
- Untransmitted data is present (when the SCR.TE bit is 1).
- The SSR.ORER flag is 0.

[Condition for high-level output]
The conditions for low-level output have not been satisfied.

(b) SCI10 and SCI11 When FIFO is Enabled

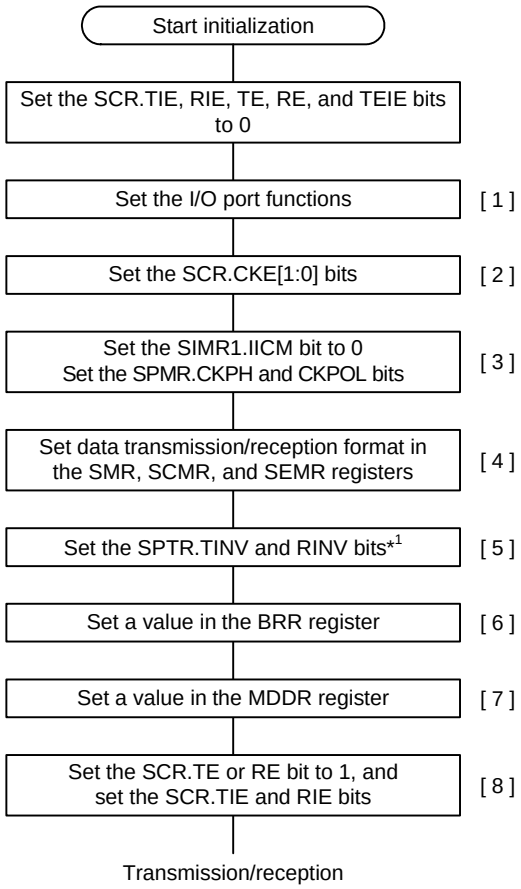
[Conditions for low-level output]
When the following conditions are all satisfied.

- The SCR.RE or SCR.TE bit is 1.
- Transmission or reception is not in progress.
- The number of data stored in the receive FIFO is less than the threshold (FCR.RTRG[3:0] bits) (when the SCR.RE bit is 1).
- Untransmitted data is present (when the SCR.TE bit is 1).
- The SSRFIFO.ORER flag is 0.

[Condition for high-level output]
When the conditions for low-level output are not satisfied.

35.5.3 SCI Initialization (Clock Synchronous Mode)

Before transmitting and receiving data, start by writing the initial value 00h to the SCR register and then continue through the procedure for SCI given in Figure 35.30. Whenever the operating mode or transfer format is changed, the SCR register must be initialized before the change is made.
Note that setting the SCR.RE bit to 0 initializes neither the ORER, FER, and PER flags in the SSR register nor the RDR register.



[1] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[2] Set the clock selection in the SCR register.
When an internal clock is selected, the SCK pin functions as the clock output pin.

[3] Set the SIMR1.IICM bit to 0.
Set the SPMR.CKPH and CKPOL bits.
Step [3] can be skipped if the values have not been changed from the initial values.

[4] Set the data transmission/reception format in the SMR, SCMR, and SEMR registers.

[5] Set the polarities of transmit/receive signals by the SPTR.TINV and RINV bits*1.

[6] Write a value corresponding to the bit rate to the BRR register. This step is not necessary if an external clock is used.

[7] Write the value obtained by correcting a bit rate error in the MDDR register. This step is not necessary if the SEMR.BRME bit is set to 0 or an external clock is used.

[8] Set the SCR.TE or RE bit to 1. At this time, also set the SCR.TIE and RIE bits.
Setting the TE and RE bits allows TXDn and RXDn to be used.

Note: In simultaneous transmit and receive operations, the SCR.TE and RE bits should both be set to 0 or 1 simultaneously.
Note 1. Only for SCI0 to SCI11.

Figure 35.30 Example of SCI Initialization Flowchart (Clock Synchronous Mode)

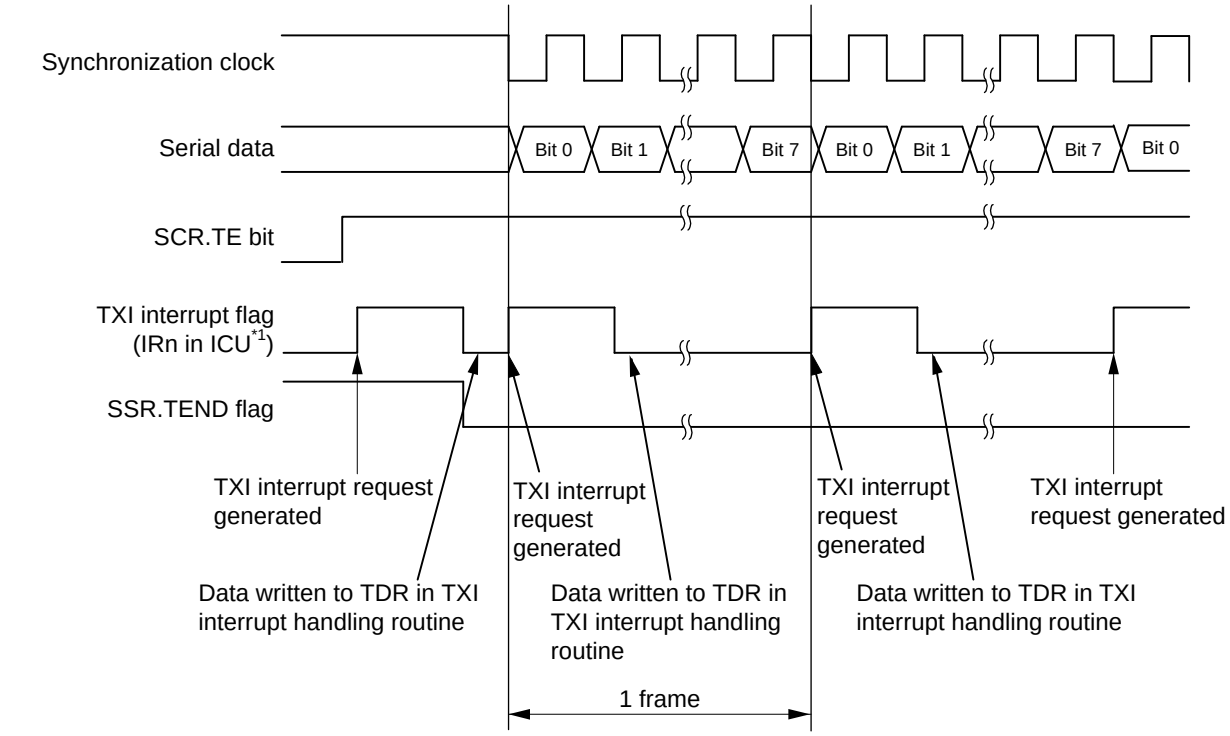
35.5.4 Serial Data Transmission (Clock Synchronous Mode)

(1) SCI0 to SCI9 and SCI12, and SCI10 and SCI11 when FIFO is Disabled

Figure 35.31, Figure 35.32, and Figure 35.33 show an example of the operation for serial transmission in clock synchronous mode.
In serial data transmission, the SCI operates as described below.

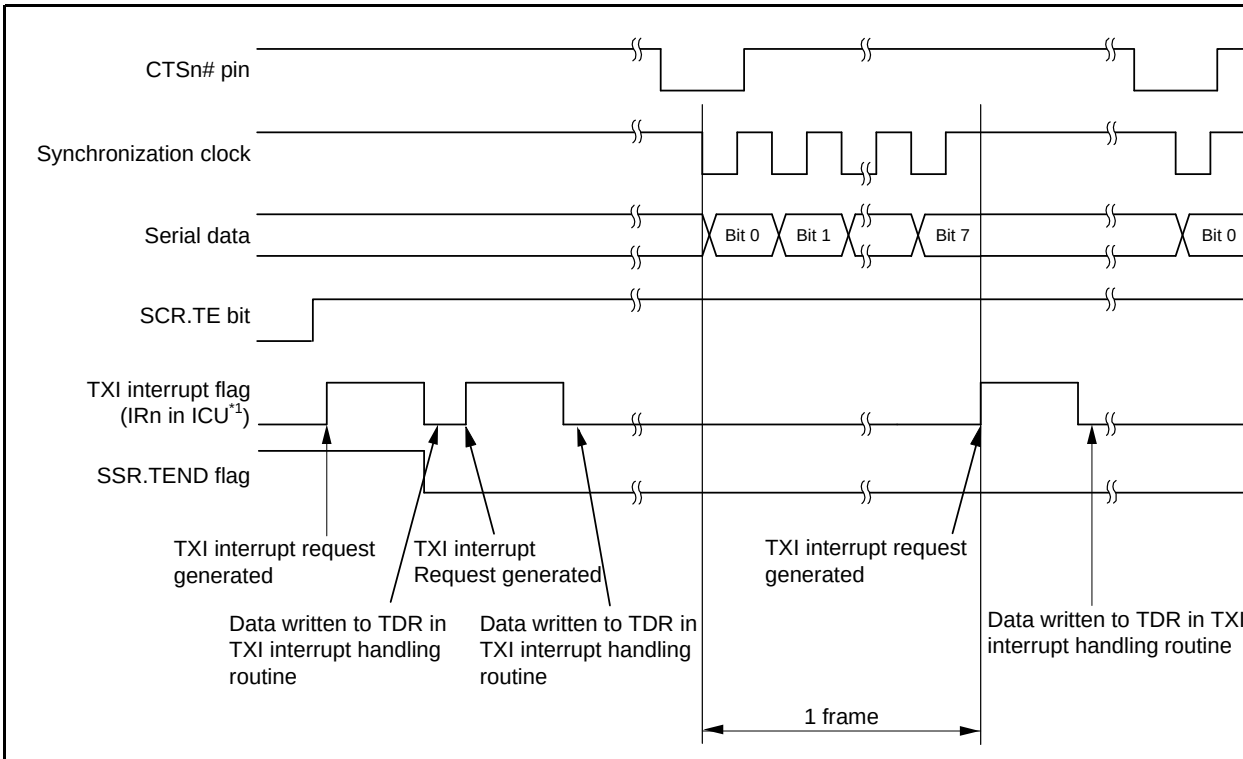
1. The SCI transfers data from the TDR register to the TSR register when data is written to the TDR register in the TXI interrupt handling routine. The TXI interrupt request at the beginning of transmission is generated when the SCR.TE bit is set to 1 after the SCR.TIE bit is set to 1 or when these 2 bits are set to 1 simultaneously by a single instruction.
2. After transferring data from the TDR register to the TSR register, the SCI starts transmission. When the SCR.TIE bit is set to 1 at this time, a TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to the TDR register in this TXI interrupt handling routine before transmission of the current transmit data has finished. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (a TXI interrupt request is disabled) and the SCR.TEIE bit to 1 (a TEI interrupt request is enabled) after the last of the data to be transmitted are written to the TDR register from the handling routine for TXI requests.
3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when clock output mode has been specified and in synchronization with the input clock when use of an external clock has been specified. Output of the clock signal is suspended until the input CTS signal is at the low level while the SPMR.CTSE bit is 1 (CTS function is enabled).
4. The SCI checks for updating of (writing to) the TDR register at the time of the last bit output.
5. When TDR is updated, the next transmit data is transferred from the TDR register to the TSR register, and serial transmission of the next frame is started.
6. If the TDR register is not updated, set the SSR.TEND flag to 1 and the TXDn pin retains the output state of the last bit. If the SCR.TEIE bit is 1 at this time, a TEI interrupt request is generated. The SCKn pin is held high.

Figure 35.34 shows a sample flowchart of serial data transmission.
Transmission will not start while a receive error flag (ORER, FER, or PER in the SSR register) is set to 1. Be sure to set the receive error flags to 0 before starting transmission. Note that setting the SCR.RE bit to 0 does not clear the receive error flags.



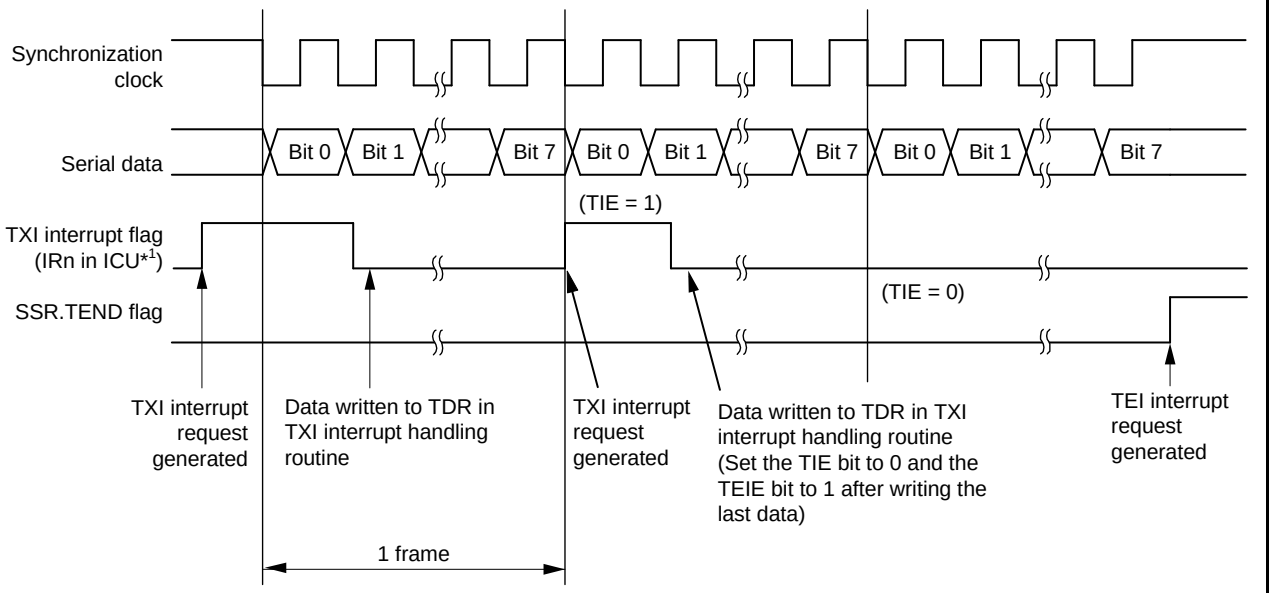
Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 35.31 Example of Serial Data Transmission in Clock Synchronous Mode When the CTS Function is Not Used at the Beginning of Transmission



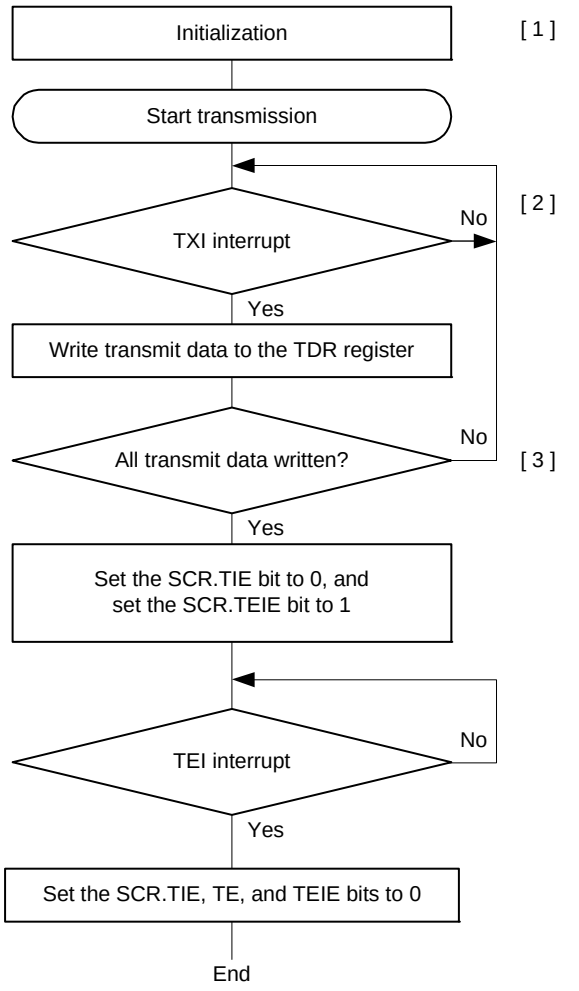
Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 35.32 Example of Serial Data Transmission in Clock Synchronous Mode When the CTS Function is Used at the Beginning of Transmission



Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 35.33 Example of Serial Data Transmission in Clock Synchronous Mode from the Middle of Transmission until Transmission Completion



[1] SCI initialization:
Set data transmission.

[2] Writing transmit data write to the TDR register by a TXI interrupt request:
When transmit data is transferred from the TDR register to the TSR register, a transmit data empty interrupt (TXI) request is generated.
Transmit data is written to the TDR register once from the handling routine for TXI requests.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to the TDR register upon accepting a transmit data empty interrupt (TXI) request. Transmit data can also be written to the TDR register by activating the DMAC or DTC by the TXI interrupt request.
When TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted are written to the TDR register.

Note: When the external clock is in use (the value of the SCR.CKE[1:0] bits is 10b or 11b), the rising edge on the SCK pin for the last bit sets the SSR.TEND flag to 1. Setting the SCR.TE bit to 0 immediately after this may lead to insufficient received-data hold time on the receiver side.

Figure 35.34 Example Flowchart of Serial Transmission in Clock Synchronous Mode

(2) SCI10 and SCI11 When FIFO is Enabled

Set the transmit data in the FTDR register (FTDR.L register) instead of the TDR register. When data is transferred from the transmit FIFO to the TSR register and if the number of data stored in the transmit FIFO is less than or equal to the threshold (FCR.TTRG[3:0] bits), a transmit data empty interrupt (TXI) request is generated.
Up to 16 minus FDR.T[4:0] frames of transmit data can be set within the TXI interrupt processing routine. When the settings of all transmit data are completed, set the SSRFIFO.TDFE flag to 0.
When the transmit data is set by using the DMAC or DTC, the TDFE flag is automatically set to 0.
When transmitting serial data, the SCI operates as follows.

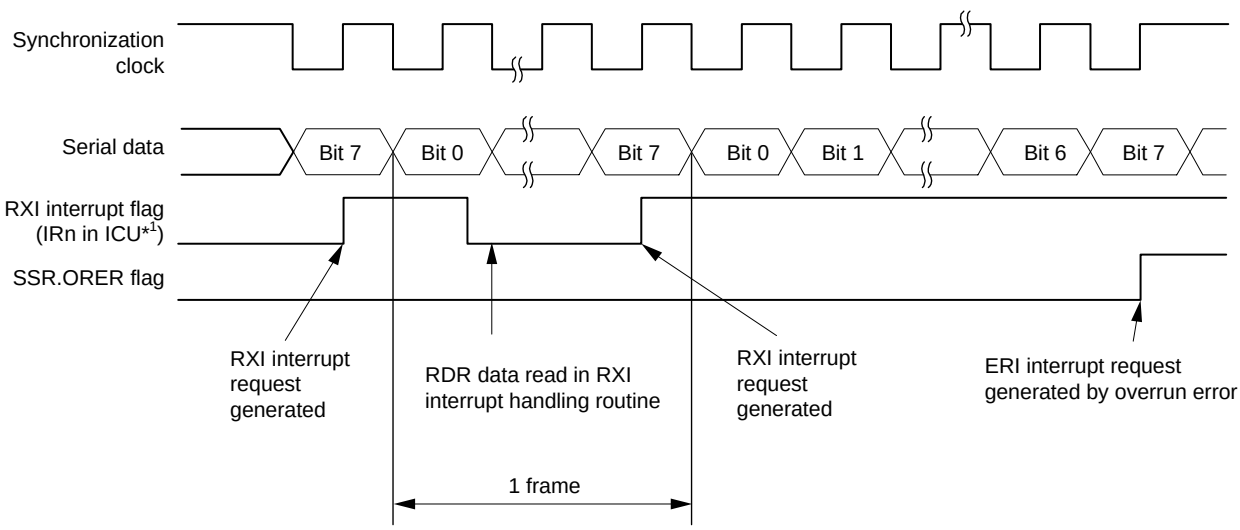
1. When the SCR.TIE and SCR.TE bits are simultaneously set to 1, a TXI interrupt request is generated. Up to 16 minus FDR.T[4:0] bytes of transmit data can be set in the FTDR register (FTDR.L register) within the TXI interrupt processing routine.
2. After the transmit data is written to the FTDR register, the SCI starts transmission by transferring the data to the TSR register in sequence starting from the top of the transmit FIFO. When the number of untransmitted data in the transmit FIFO is less than or equal to the specified threshold (FCR.TTRG[3:0] bits), the SSRFIFO.TDFE flag is set to 1. At this time, if the SCR.TIE bit is 1, a TXI interrupt request is generated. Writing the transmit data to the FTDR register by using the TXI interrupt processing routine before the transmit FIFO becomes empty enables sequence transmission. When a TEI interrupt request is used, the SCI writes the last transmit data to the FTDR register by using the TXI interrupt processing routine, and then sets the SCR.TIE bit to 0 and the SCR.TEIE bit to 1.
3. The SCI outputs 8-bits of data from the TXDn pin in synchronization with the clock for output when the SCR.CKE[1] bit is 0 (internal clock) and in synchronization with the clock for input when the SCR.CKE[1] bit is 1 (external clock). When the SPMR.CTSE bit is 1 (with the CTS function enabled), the SCI waits until the low is input to the CTSn# pin and starts transmission.
4. The SCI checks whether any untransmitted data is in the transmit FIFO when it transmits the last bit.
5. When any untransmitted data is in the transmit FIFO, the SCI transfers the data to the TSR register from the transmit FIFO and starts transmission of the next frame data.
6. When no untransmitted data remains in the transmit FIFO, the SCI sets the SSRFIFO.TEND flag to 1 and stops shifting by the TSR register. At this time, if the SCR.TEIE bit is 1, a TEI interrupt request is generated. The TXDn pin is fixed to the value of the last bit of the last transmitted data and the SCKn pin is fixed to high.

35.5.5 Serial Data Reception (Clock Synchronous Mode)

(1) SCI0 to SCI9 and SCI12, and SCI10 and SCI11 when FIFO is Disabled

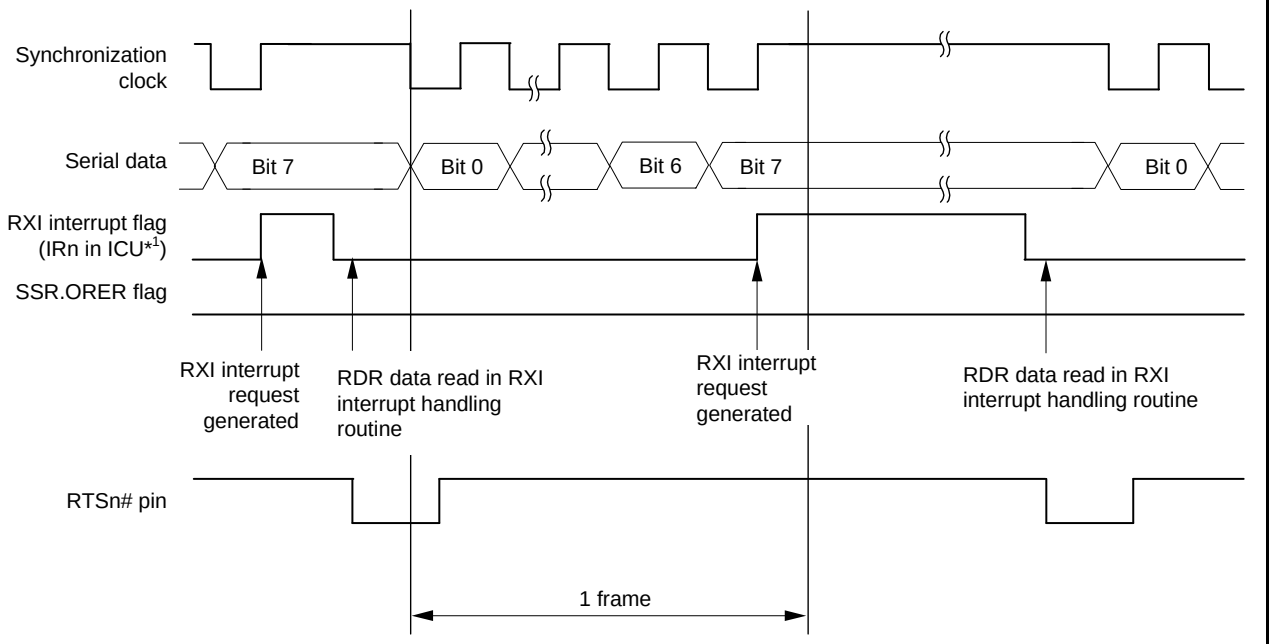
Figure 35.35 and Figure 35.36 show an example of SCI operation for serial reception in clock synchronous mode. In serial data reception, the SCI operates as described below.

1. The value of the SCR.RE bit becoming 1 places the signal output on the RTSn# pin at the low level (when the RTS function is in use).
2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores the receive data in the RSR register.
3. If an overrun error occurs, the SSR.ORER flag is set to 1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to the RDR register.
4. When reception finishes successfully, receive data is transferred to the RDR register. If the SCR.RIE bit is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the RDR register in this RXI interrupt handling routine before reception of the next receive data is completed. Reading out the received data that have been transferred to the RDR register causes the RTSn# pin to output the low level (when the RTS function is in use).



Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 35.35 Example of Operation for Serial Reception in Clock Synchronous Mode (1) (When RTS Function is Not Used)

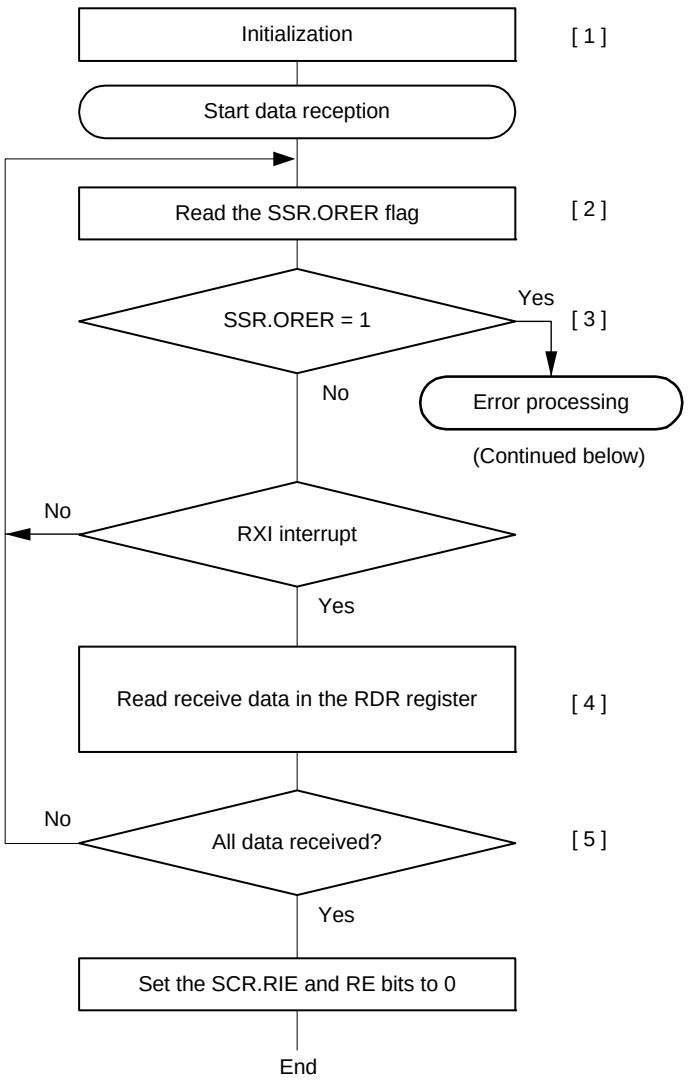


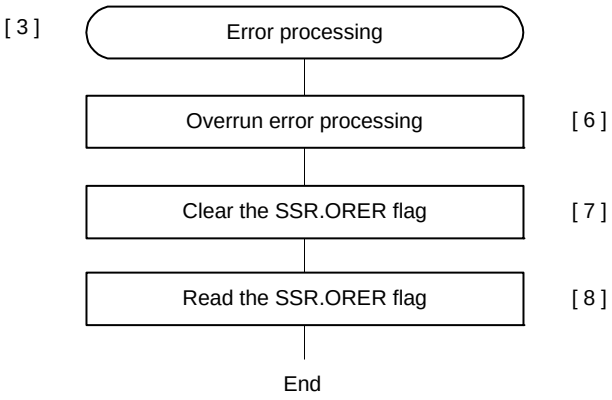
Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 35.36 Example of Operation for Serial Reception in Clock Synchronous Mode (2) (When RTS Function is Used)

Data transfer cannot be resumed while a receive error flag is 1. Accordingly, clear the ORER, FER, and PER flags in the SSR register before resuming reception. Moreover, be sure to read the RDR register during overrun error processing. When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read the RDR register because received data which has not yet been read may be left in the RDR register.

Figure 35.37 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 SSR.ORER flag, perform the relevant error processing, and then set the ORER flag to 0. Data reception cannot be resumed while the ORER flag is 1.

[4] Read the receive data in the RDR register once in the receive data full interrupt (RXI) request handling routine.

[5] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, finish reading the receive data in the RDR register. The RDR data can also be read by activating the DMAC or DTC by an RXI interrupt request.

[6] Processing in response to an overrun error:
Read the RDR register. In combination with step [7], this will make correct reception of the next frame possible.

[7] Clearing the error flag:
Write 0 to the error flag.

[8] Confirming that the error flag is cleared:
Read the error flag to confirm that its value is 0.

Figure 35.37 Example Flowchart of Serial Reception in Clock Synchronous Mode

(2) SCI10 and SCI11 When FIFO is Enabled

The received data and status flag are read from the FRDR register instead of the RDR register. When reading in byte units, read the FRDR.H register and then the FRDR.L register in this order. Reading the FRDR.L register updates the FRDR.RDAT[7:0] bits. The value of the RDF and ORER flags in the FRDR register is the same as that of the SSRFIFO register.
When receiving serial data, the SCI operates as follows.

1. If the SCR.RE bit is set to 1 and the RTSn# function is in use, the SCI drives the RTSn# pin low.
2. The SCI starts receiving data in synchronization with the clock input or output and stores the received data in the RSR register.
3. When there is no space in the receive FIFO, an overrun error occurs. When an overrun error occurs, the SCI sets the SSRFIFO.ORER flag to 1. At this time, if the SCR.RIE bit is 1, an ERI interrupt request is generated. The received data is not transferred to the receive FIFO.
4. When the data is successfully received, the SCI transfers the received data to the receive FIFO. If the number of data stored in the receive FIFO is equal to or greater than the threshold (FCR.RTRG[3:0] bits), the SCI sets the SSRFIFO.RDF flag to 1. At this time, if the SCR.RIE bit is 1, an RXI interrupt request is generated. Using the RXI interrupt processing routine allows reading of the received data in the RDRF register before an overrun error occurs, thus enabling the sequence reception of data. When the received data that have been transferred to the receive FIFO are read and the number of unread data is less than the value of the FCR.RSTRG[3:0] bits, the SCI drives the RTSn# pin low.

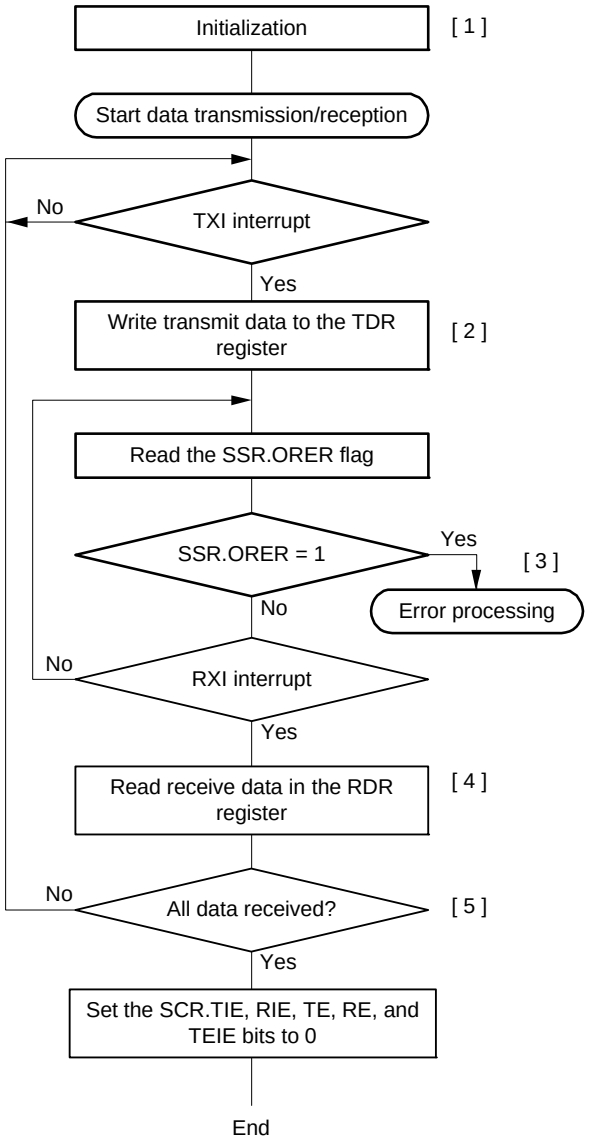
35.5.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode)

Figure 35.38 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 SSR.TEND flag is 1, and then initialize the SCR register. Then set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously by a single instruction.

To switch from receive mode to simultaneous transmit and receive mode, check that the SCI has finished reception, and then set the RIE and RE bits to 0. Then check that the receive error flags (ORER, FER, and PER in the SSR register) are 0, and then set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously by a single instruction.



[1] SCI initialization:
The TXDn pin can act as the output pin for transmitted data and the RXDn pin can act as the input pin for received data at the same time.

[2] Transmit data write:
Write transmit data to the TDR register once in the TXI interrupt handling routine.

[3] Receive error processing:
If a receive error occurs, read the SSR.ORER flag, perform the relevant error processing, and then set the ORER flag to 0. Data reception cannot be resumed while the ORER flag is 1.

[4] Reading receive data:
Read the receive data in the RDR register once in the RXI interrupt request handling routine.

[5] Serial transmission/reception continuation procedure:
To continue serial transmission and reception, before the MSB (bit 7) of the current frame is received, finish reading the receive data in the RDR register by the RXI interrupt. Also, before the MSB (bit 7) of the current frame is transmitted, write data to the TDR register by the TXI interrupt. Transmit data can also be written to the TDR register by activating the DMAC or DTC by a transmit data empty interrupt (TXI) request. Similarly, the RDR data can also be read by activating the DMAC or DTC by a receive data full interrupt (RXI) request.

Note: When switching from transmit or receive operation to simultaneous transmit and receive operations first set the TIE, RIE, TE, RE, and TEIE bits in the SCR register to 0, and then set TIE, RIE, TE, and RE bits to 1 simultaneously.

Figure 35.38 Example Flowchart of Simultaneous Serial Transmission and Reception in Clock Synchronous Mode

35.6 Operation in Smart Card Interface Mode

The SCI supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards), as an extended function of the SCI.
Smart card interface mode can be selected using the appropriate register.

35.6.1 Sample Connection

Figure 35.39 shows a sample connection between a smart card (IC card) and this MCU.
As in the figure, since this MCU communicates with an IC card using a single transmission line, interconnect the TXDn and RXDn pins and pull up the data transmission line to VCC using a resistor.
Setting the TE and RE bits in the SCR register to 1 with an IC card disconnected enables closed transmission/reception allowing self-diagnosis.
To supply an IC card with the clock pulses generated by the SCI, input the SCKn pin output to the CLK pin of an IC card.
The output port of the this MCU can be used to output a reset signal.

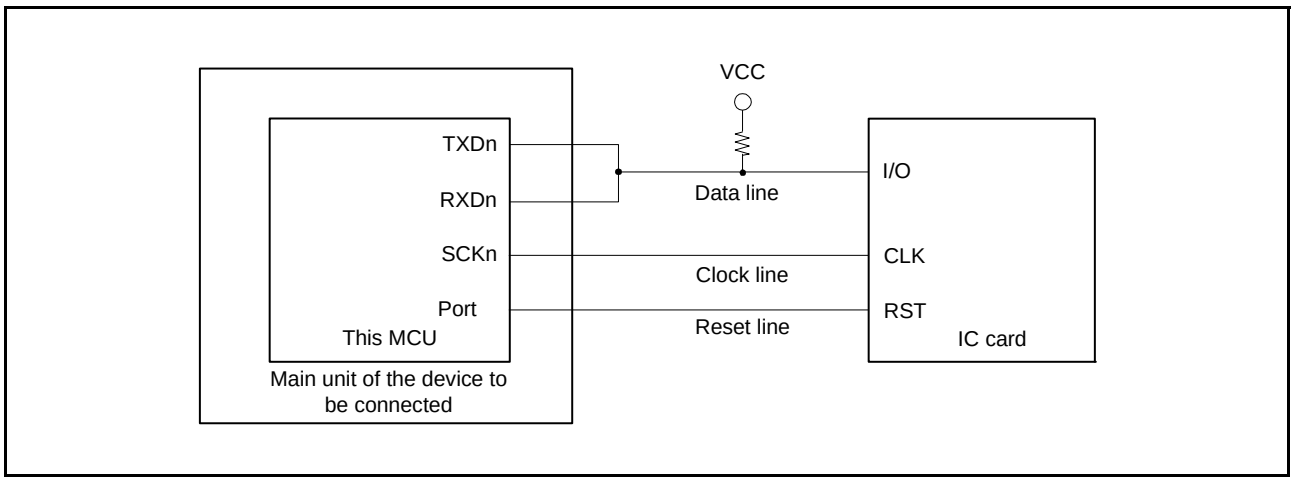


Figure 35.39 Sample Connection with a Smart Card (IC Card)

35.6.2 Data Format (Except in Block Transfer Mode)

Figure 35.40 shows the data transfer formats in smart card interface mode.

- One frame consists of 8-bit data and a parity bit in asynchronous mode.
- During transmission, at least 2 etu (elementary time unit: time required for transferring 1 bit) is secured as a guard time from the end of the parity bit until the start of the next frame.
- If a parity error is detected during reception, a low-level error signal is output for 1 etu after 10.5 etu has passed from the start bit.
- If an error signal is sampled during transmission, the same data is automatically retransmitted after at least 2 etu.

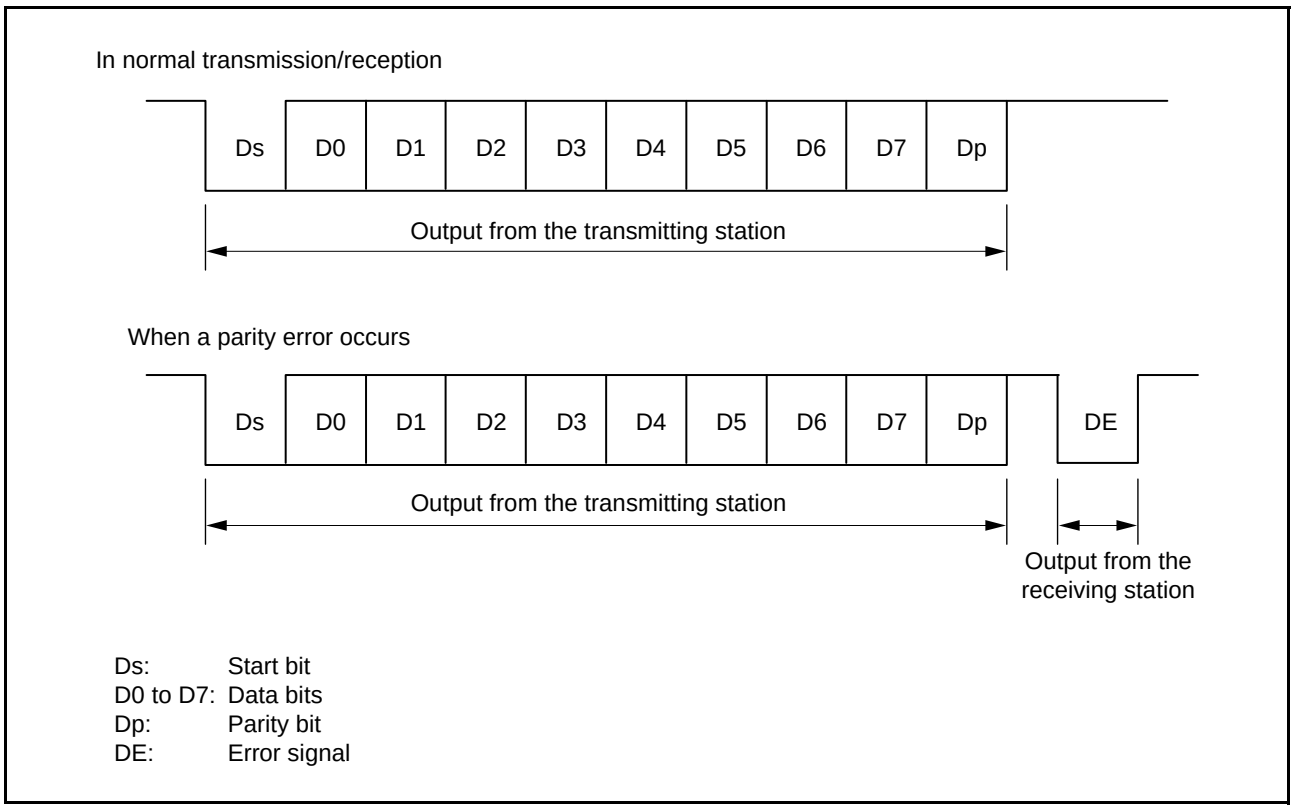


Figure 35.40 Data Formats in Smart Card Interface Mode

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 35.41. Therefore, data in the start character in the figure is 3Bh. When using the direct convention type, write 0 to both the SDIR and SINV bits in the SCMR register. Write 0 to the SMR.PM bit in order to use even parity, which is prescribed by the smart card standard.

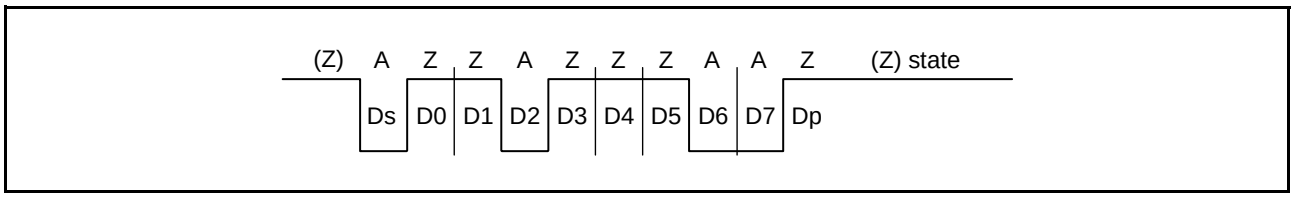


Figure 35.41 Direct Convention (SCMR.SDIR bit = 0, SCMR.SINV bit = 0, SMR.PM bit = 0)

(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 35.42. Therefore, data in the start character in the figure is 3Fh. When using the inverse convention type, write 1 to both the SDIR and SINV bits in the SCMR register. The parity bit is logic level 0 to produce even parity, which is prescribed by the smart card standard, and corresponds to state Z. Since the SINV bit of the this MCU only inverts data bits D7 to D0, write 1 to the SMR.PM bit to invert the parity bit for both transmission and reception.

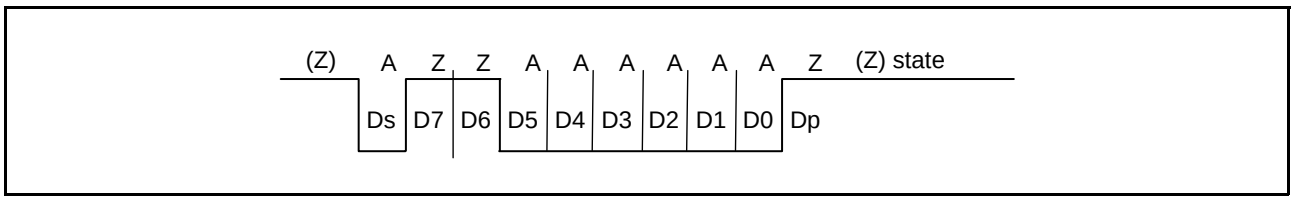


Figure 35.42 Inverse Convention (SCMR.SDIR bit = 1, SCMR.SINV bit = 1, SMR.PM bit = 1)

35.6.3 Block Transfer Mode

Block transfer mode is different from non-block transfer mode in the following respects.

- Even if a parity error is detected during reception, no error signal is output. Since the SSR.PER flag is set by error detection, clear the PER flag before receiving the parity bit of the next frame.
- During transmission, at least 1 etu is secured as a guard time from the end of the parity bit until the start of the next frame.
- Since the same data is not retransmitted during transmission, the SSR.TEND flag is set 11.5 etu after transmission start.
- In block transfer mode, the SSR.ERS flag indicates the error signal status as in non-block transfer mode, but the flag is read as 0 because no error signal is transferred.

35.6.4 Receive Data Sampling Timing and Reception Margin

Only the base clock generated by the on-chip baud rate generator can be used as a transmit/receive 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 SCMR.BCP2 bit and the SMR.BCP[1:0] bits.

For data reception, the falling edge of the start bit is sampled with the base clock to perform 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 35.43. The reception margin here is determined by the following formula.

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ (\%)$$

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 × 372)} × 100 (%) = 49.866 (%)

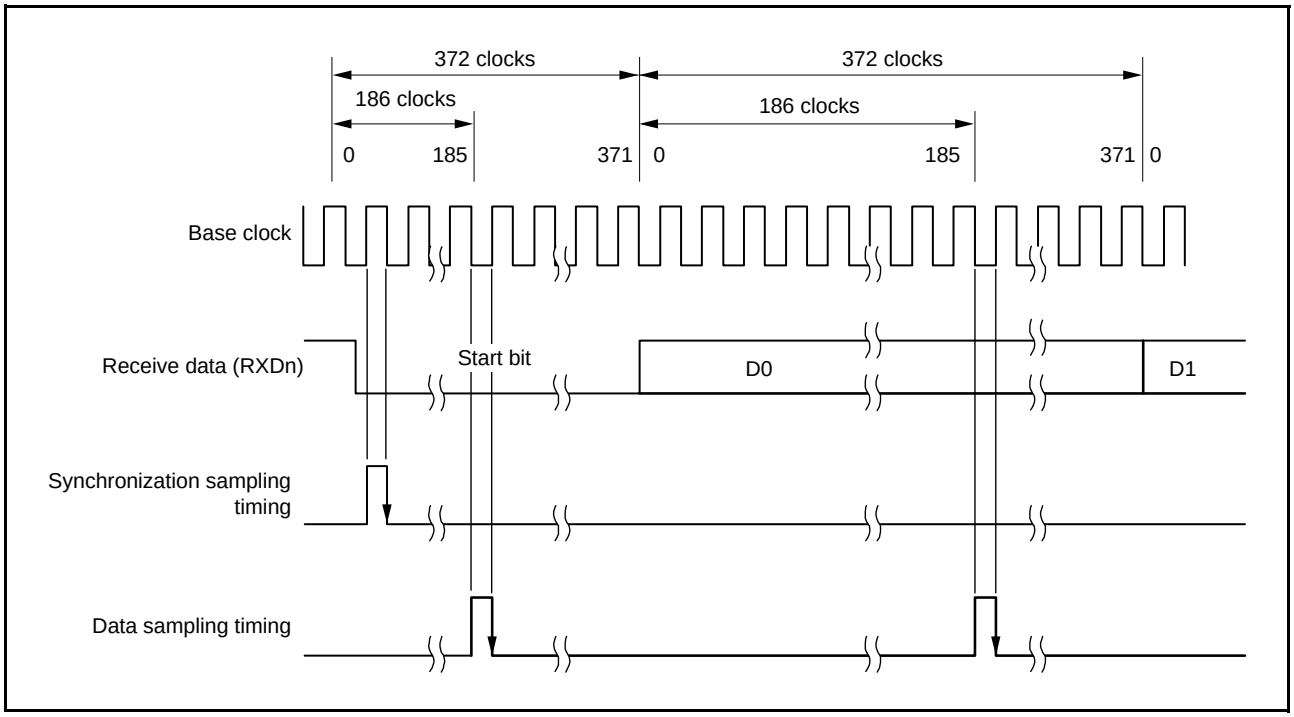


Figure 35.43 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)

35.6.5 SCI Initialization (Smart Card Interface Mode)

Initialize the SCI following the example of flowchart shown in Figure 35.44.
Initialize the SCR and SSR registers before switching from transmit mode to receive mode and vice versa. When not changing the bit rate, it is not necessary to set the CKE[1:0] bits to 00b. Even if the RE bit is set to 0, the RDR register is not initialized.

To change receive mode to transmit mode, first check that reception has completed, and then execute steps [1] and [3] in Figure 35.44. Set TE = 1 and RE = 0 at step [11]. Reception completion can be verified by reading the RXI request, SSR.ORER, or SSR.PER flag.
To change transmit mode to receive mode, first check that transmission has completed, and then execute steps [1] and [3] in Figure 35.44. Set TE = 0 and RE = 1 at step [11]. Transmission completion can be verified by reading the SSR.TEND flag.

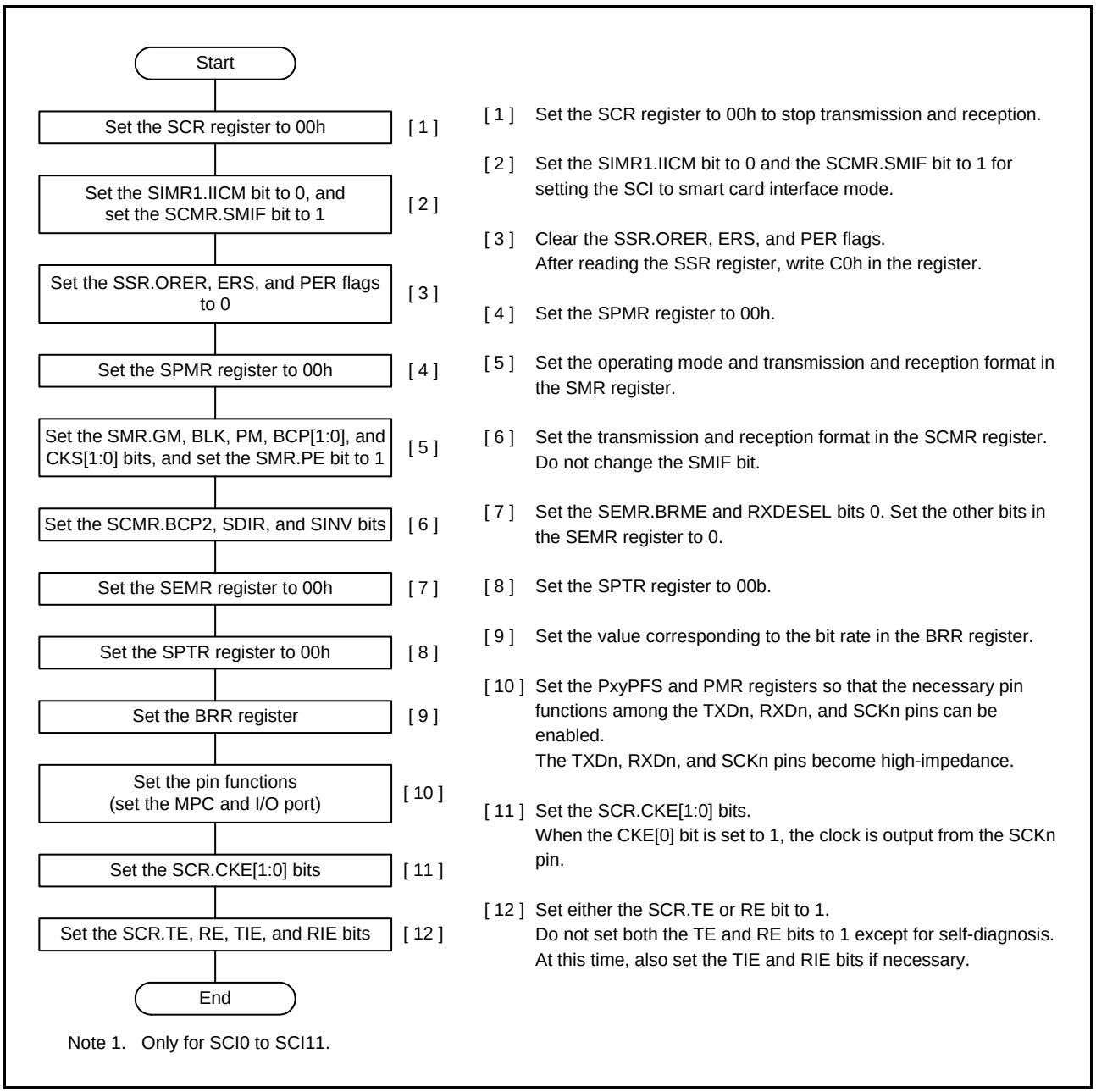


Figure 35.44 Example of SCI Initialization Flowchart (Smart Card Interface Mode)

Figure 35.45 shows an example of data transmission when the SCI is set to smart card interface mode according to the flow described in Figure 35.44 after a reset. When the pin functions are set to the SCK and TXD pins, they are still high-impedance because the SCR.CKE[0] and SCR.TE bits are 0. When the CKE[0] bit is set to 1, clock is output from the SCK pin. When the transmit data is written after setting the TE bit to 1, a data transmission starts. After the TE bit is set to 1, one frame of high-impedance is output from TXD pin (internal wait time) and then the data transmission starts. In smart card interface mode, the clock is continuously output while the CKE[0] bit is set to 1 (clock output) even if both the TE and RE bits are set to 0.

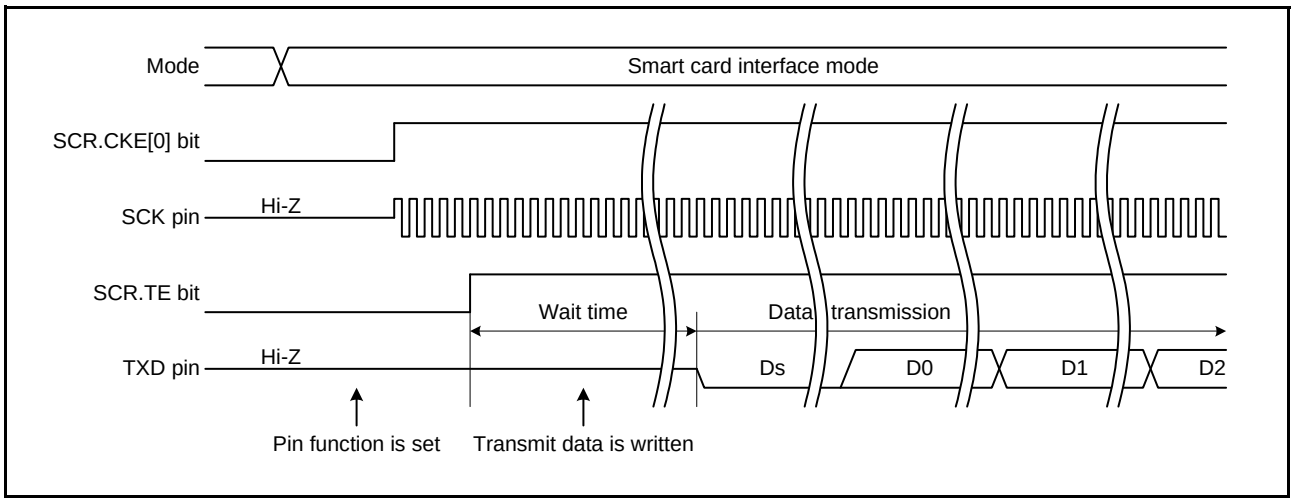


Figure 35.45 Example of Data Transmission Timing in Smart Card Interface Mode

35.6.6 Serial Data Transmission (Except in Block Transfer Mode)

Serial data transmission in smart card interface mode (except in block transfer mode), in that an error signal is sampled and data can be retransmitted, is different from that in non-smart card interface mode. Figure 35.46 shows the data retransmit operation during transmission.

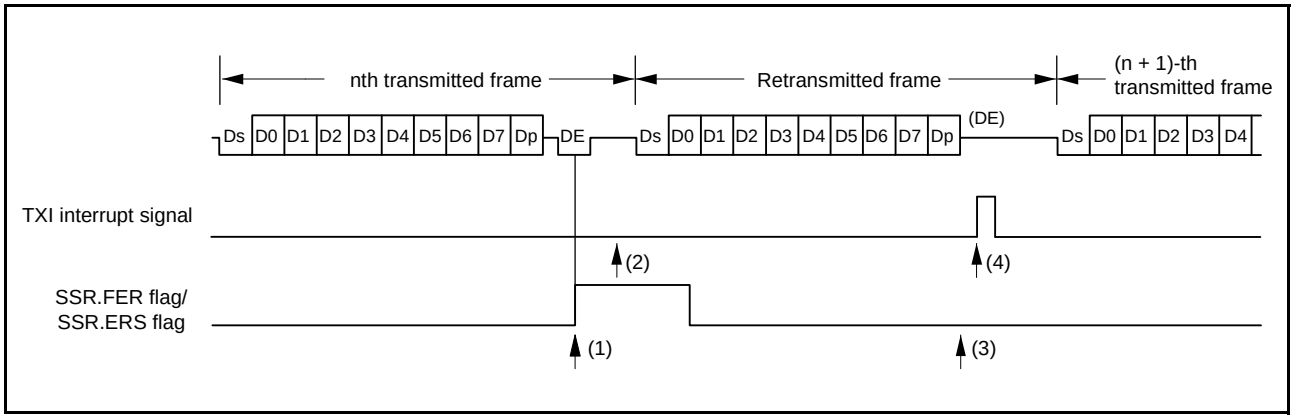


Figure 35.46 Data Retransmit Operation in SCI Transmit Mode

(1) When an error signal from the receiver end is sampled after one-frame data has been transmitted, the SSR.ERS flag is set to 1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated. Clear the ERS flag before the next parity bit is sampled.

(2) For a frame in which an error signal is received, the SSR.TEND flag is not set. Data is retransferred from the TDR register to the TSR register allowing automatic data retransmission.

(3) If no error signal is returned from the receiver, the ERS flag is not set to 1.

(4) In this case, the SCI judges that transmission of one-frame data (including retransmission) has been completed, and the TEND flag is set. If the SCR.TIE bit is 1 at this time, a TXI interrupt request is generated. Writing transmit data to the TDR register starts transmission of the next data.

Figure 35.47 shows a sample flowchart of serial transmission.

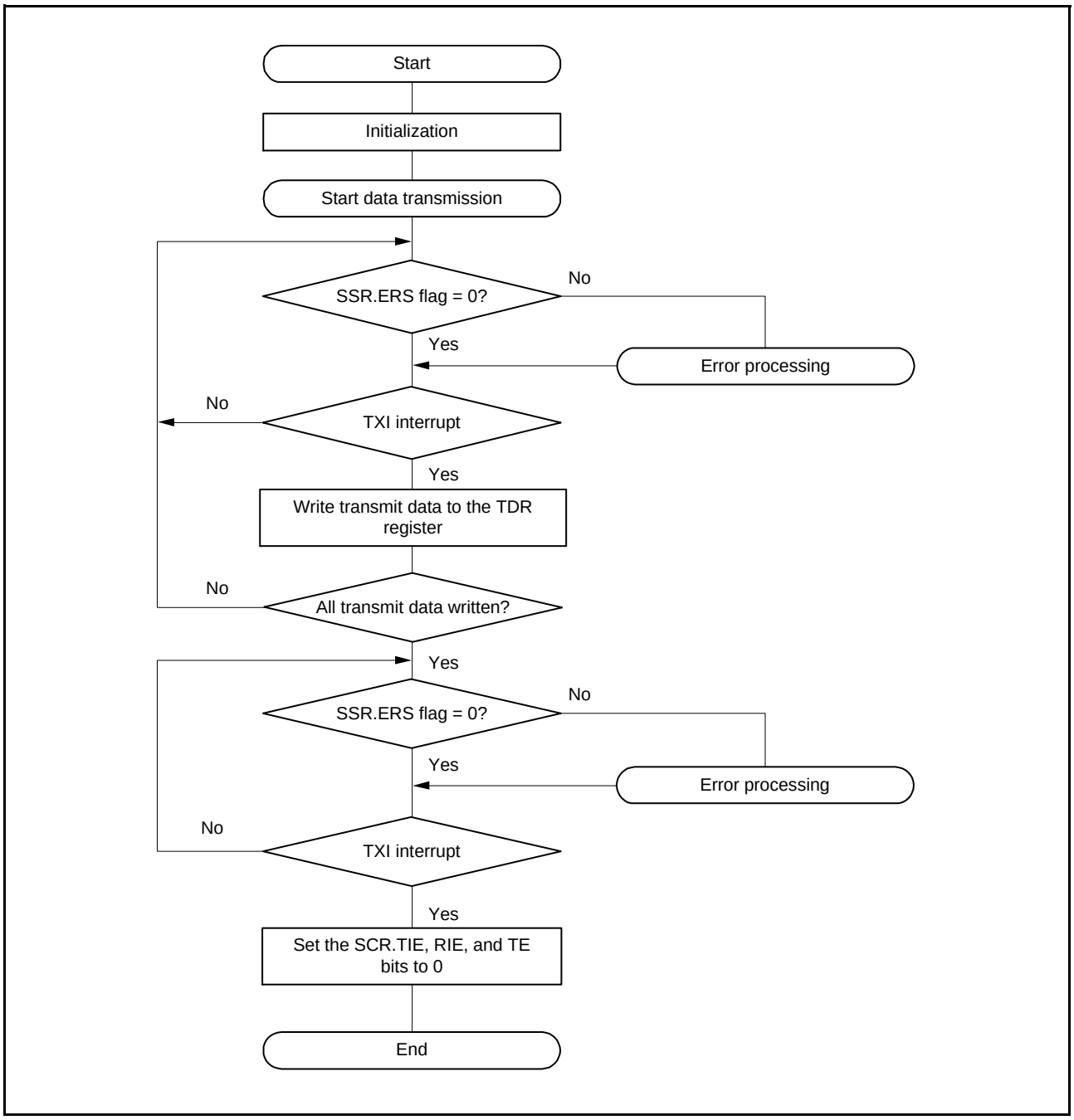


Figure 35.47 Sample Smart Card Interface Transmission Flowchart

All the processing steps are automatically performed using a TXI interrupt request to activate the DTC or DMAC. When the SSR.TEND flag is set to 1 in transmission, if the SCR.TIE bit is 1, a TXI interrupt request is generated. The DTC or DMAC is activated by a TXI interrupt request if the TXI interrupt request is specified as a source of DTC or DMAC activation beforehand, allowing transfer of transmit data. The TEND flag is automatically set to 0 when the DTC or DMAC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During this retransmission, the TEND flag is kept to 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, since the ERS flag is not automatically cleared, set the RIE bit to 1 beforehand to enable an ERI interrupt request to be generated at error occurrence, and clear the ERS flag.

When transmitting/receiving data using the DTC or DMAC, be sure to make settings to enable the DTC or DMAC before making SCI settings.
For DTC or DMAC settings, refer to section 20, Data Transfer Controller (DTCb), section 18, DMA Controller (DMACAb).

Note that the SSR.TEND flag is set in different timings depending on the SMR.GM bit setting. Figure 35.48 shows the TEND flag generation timing.

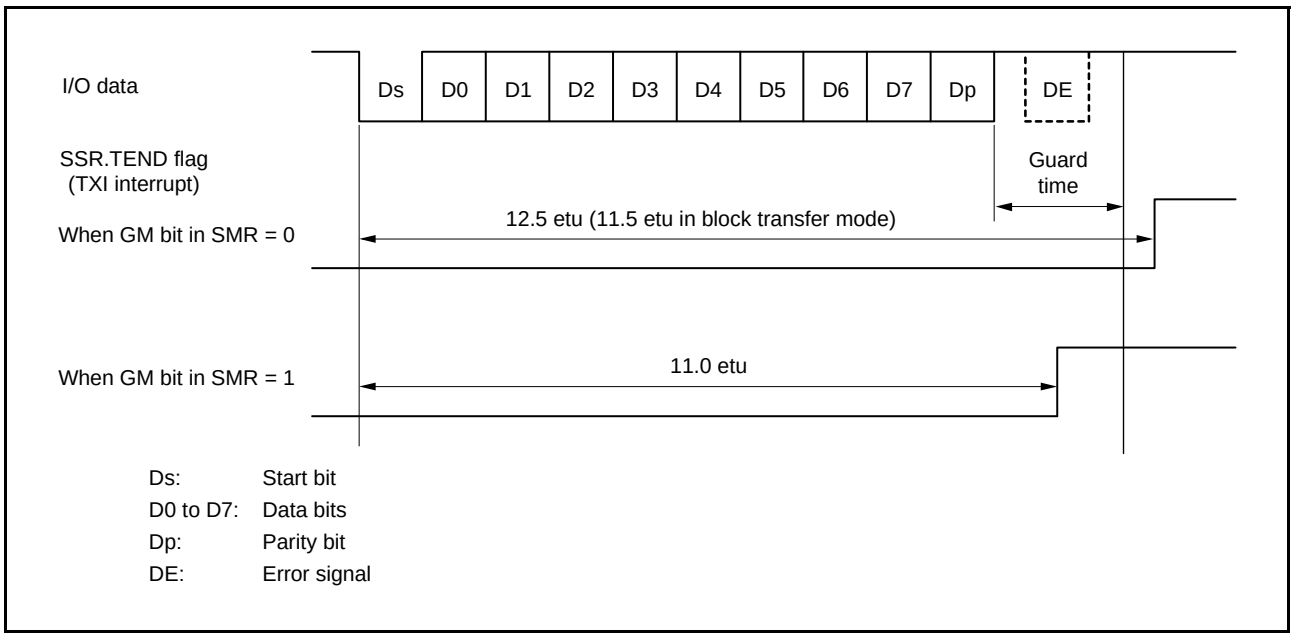


Figure 35.48 SSR.TEND Flag Generation Timing during Transmission

35.6.7 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is similar to that in non-smart card interface mode. Figure 35.49 shows the data retransmit operation in receive mode.

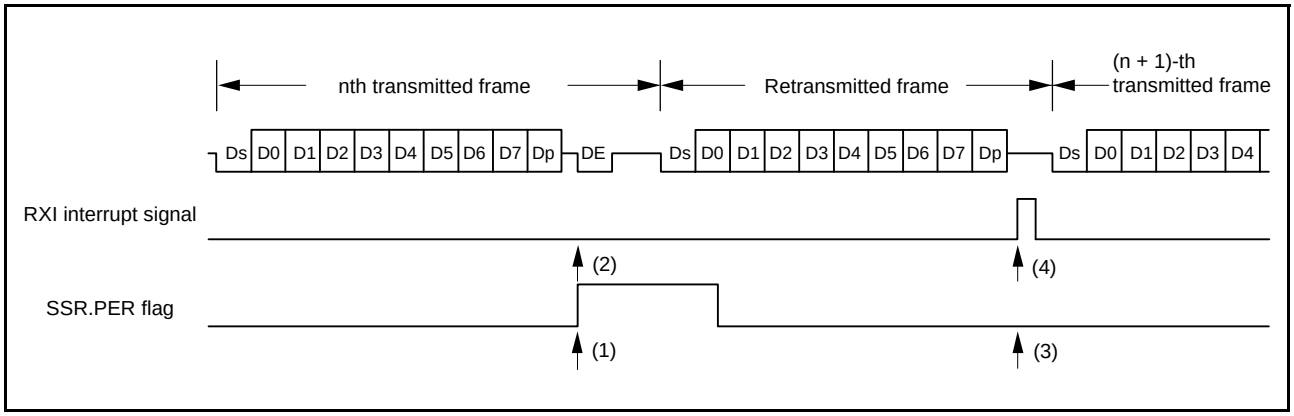


Figure 35.49 Data Retransmit Operation in SCI Receive Mode (Data Retransmit Operation during Reception)

(1) If a parity error is detected in receive data, the SSR.PER flag is set to 1. When the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated. Clear the PER flag 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 SSR.PER flag is not set to 1.
(4) In this case, data is determined to have been received successfully. When the SCR.RIE bit is 1, an RXI interrupt request is generated.

Figure 35.50 shows a sample flowchart for serial data reception.

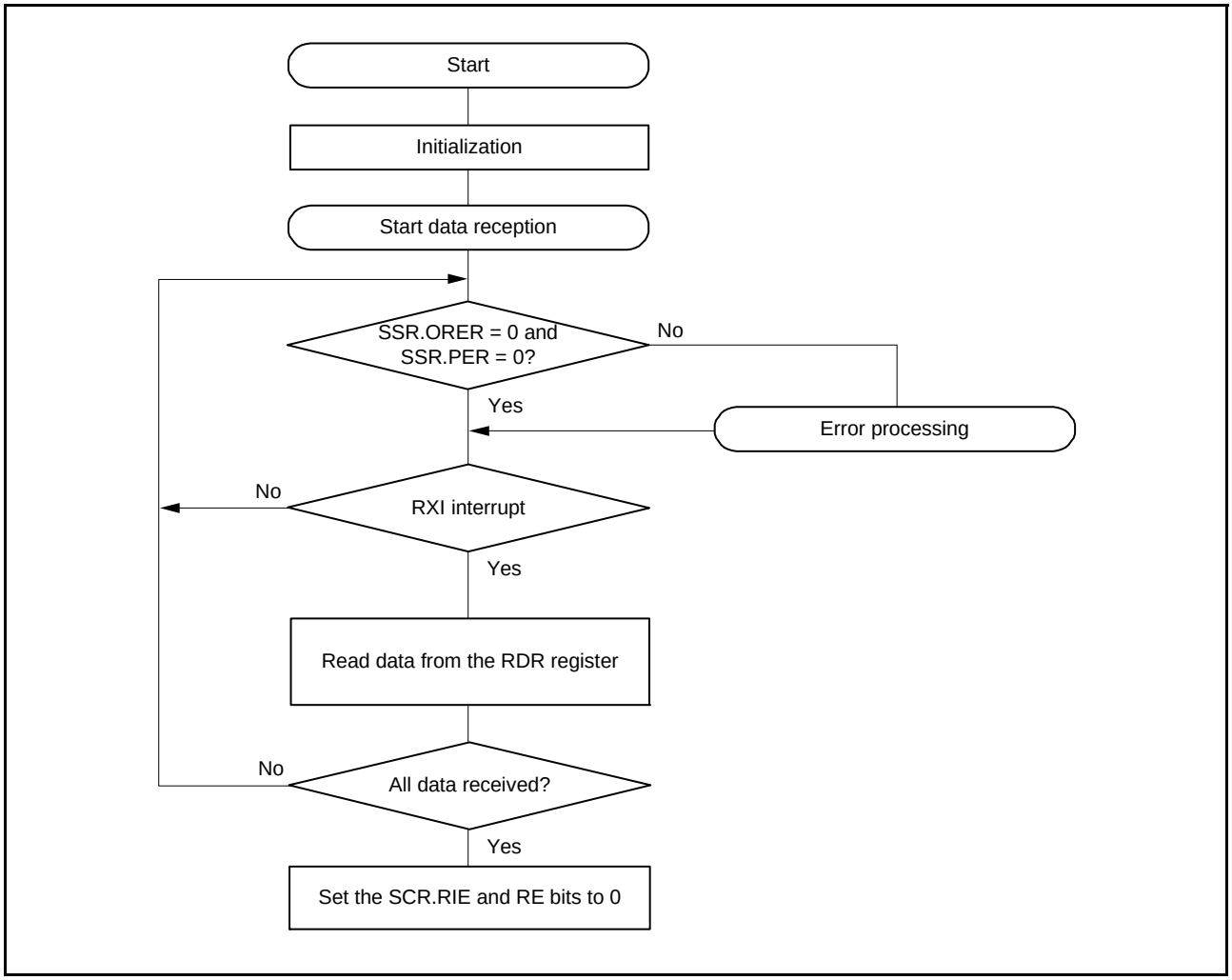


Figure 35.50 Sample Smart Card Interface Reception Flowchart

All the processing steps are automatically performed using an RXI interrupt request to activate the DTC or DMAC.
In reception, setting the RIE bit to 1 allows an RXI interrupt request to be generated. The DTC or DMAC is activated by an RXI interrupt request if the RXI interrupt request is specified as a source of DTC or DMAC activation beforehand, allowing transfer of receive data.
If an error occurs during reception and either the ORER or PER flag in the SSR register is set to 1, a receive error interrupt (ERI) request is generated. Clear the error flag after the error occurrence. If an error occurs, the DTC or DMAC is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DTC or DMAC is transferred.
Even if a parity error occurs and the PER flag is set to 1 during reception, receive data is transferred to the RDR register, thus allowing the data to be read.
When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read the RDR register because the received data which has not yet been read may be left in the RDR register.

Note 1. For operations in block transfer mode, refer to section 35.3, Operation in Asynchronous Mode.

35.6.8 Clock Output Control

Clock output can be fixed to high or low using the SCR.CKE[1:0] bits when the SMR.GM bit is 1. When the CKE[1:0] bits are set to 01b (clock output), the base clock is output from the SCK pin. For the settings of the base clock frequency (bit rate), refer to section 35.2.13, Bit Rate Register (BRR). When the CKE[1:0] bits are set to 00b (output fixed low) or 10b (output fixed to high), the SCK pin can be fixed to low or high.

Figure 35.51 shows a timing chart when the clock output is controlled.

If changing the CKE[1:0] bits while the SMR.GM bit is 0 (non-GSM mode), a pulse of unexpected width may output from SCK pin because the result is immediately reflected to the SCK pin.

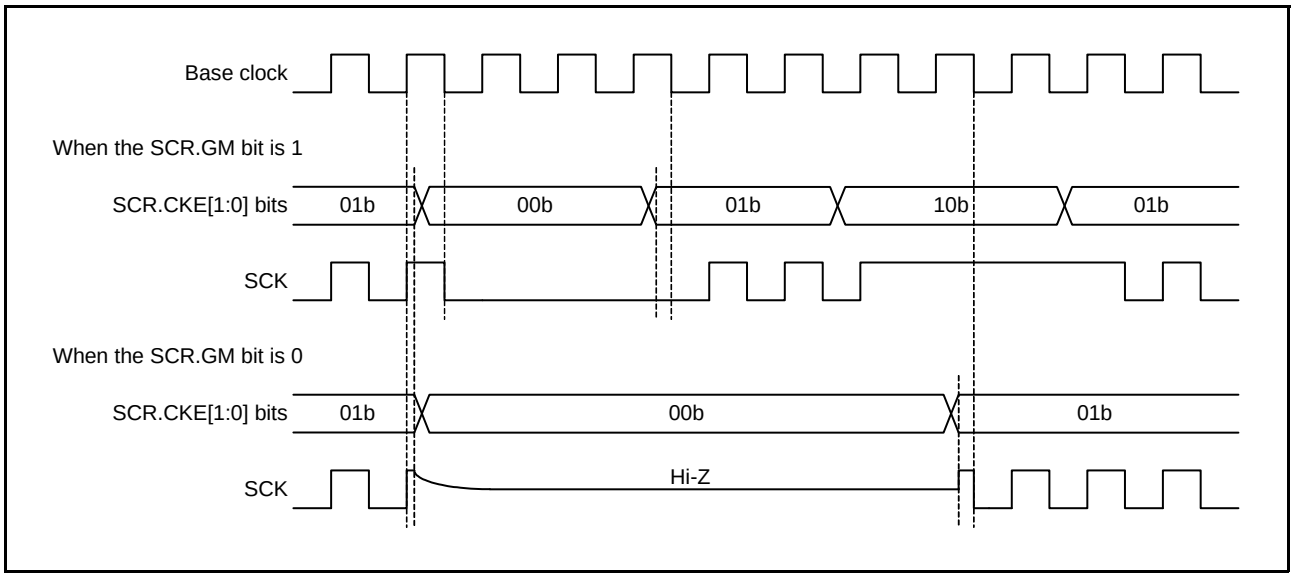


Figure 35.51 Clock Output Control

35.7 Operation in Simple I2C Mode

Simple I2C-bus format is composed of 8 data bits and an acknowledge bit. By continuing into a slave-address frame after a start condition or restart condition, a master device is able to specify a slave device as the partner for communications. The currently specified slave device remains valid until a new slave device is specified or a stop condition is satisfied.
The 8 data bits in all frames are transmitted in order from the MSB.
The I2C-bus format and timing of the I2C-bus are shown in Figure 35.52 and Figure 35.53.

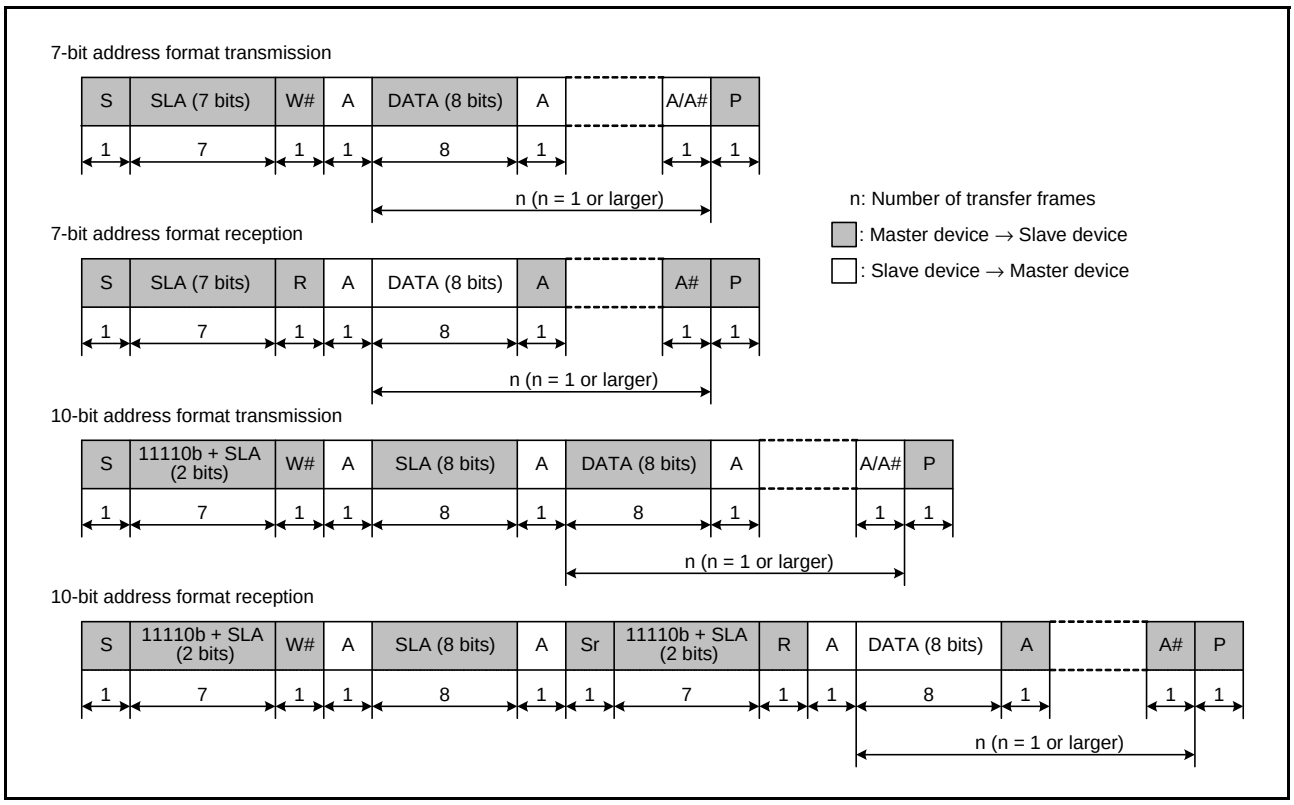


Figure 35.52 I2C-bus Format

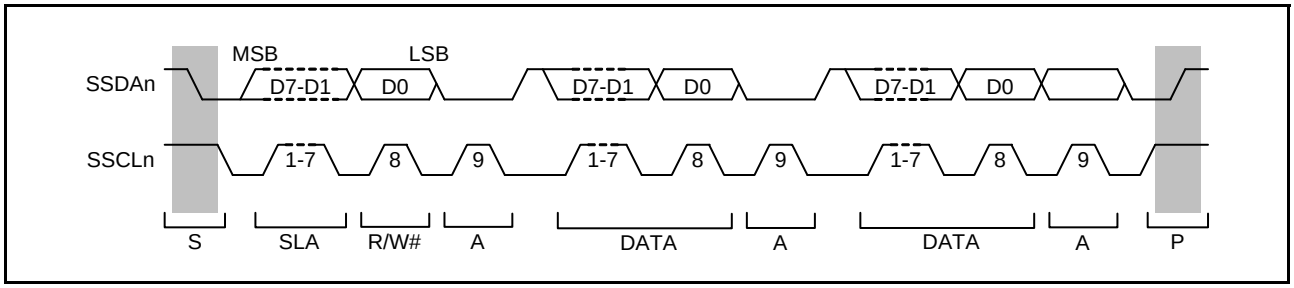


Figure 35.53 I2C-bus Timing (When SLA is 7 Bits)

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.

35.7.1 Generation of Start, Restart, and Stop Conditions

Writing 1 to the SIMR3.IICSTAREQ bit 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 SIMR3.IICSTAREQ bit is set (to 0), and a start-condition generated interrupt is output.

Writing 1 to the SIMR3.IICRSTAREQ bit 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 SIMR3.IICRSTAREQ bit is set (to 0), and a restart-condition generated interrupt is output.

Writing 1 to the SIMR3.IICSTPREQ bit 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 SIMR3.IICSTPREQ bit is set (to 0), and a stop-condition generated interrupt is output.

Figure 35.54 shows the timing of operations in the generation of start, restart, and stop conditions.

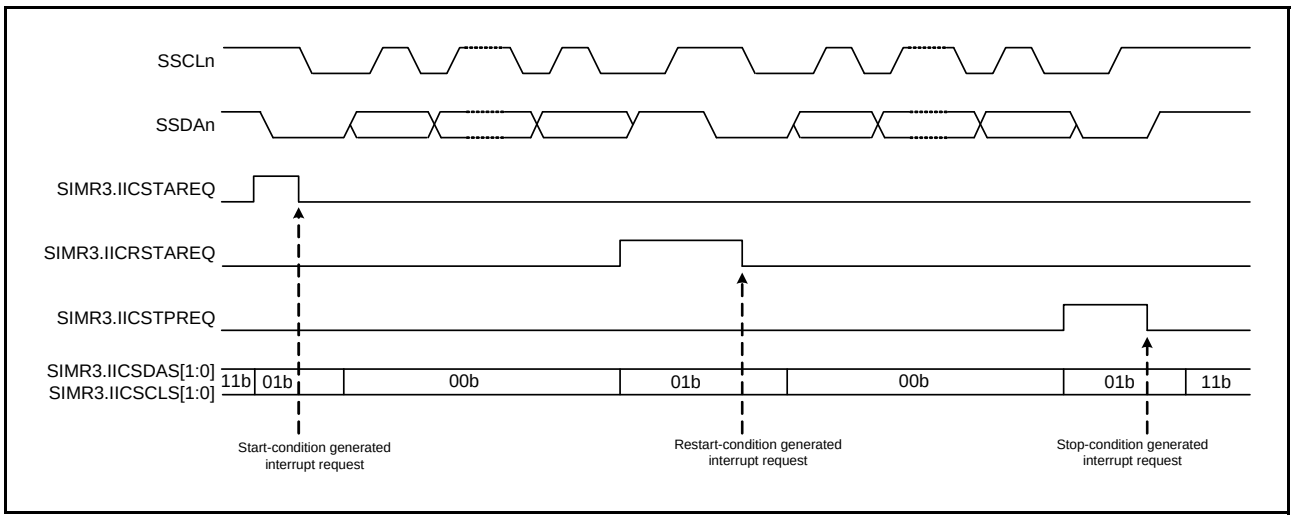


Figure 35.54 Timing of Operations in the Generation of Start, Restart, and Stop Conditions

35.7.2 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 SIMR2.IICCSC bit 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 SIMR2.IICCSC bit is set to 1, the level of the internal SSCLn clock signal changes from low to high, counting to determine the period at high level is stopped while the low level is being input on the SSCLn pin, and counting to determine the period at high level starts after the transition of the input on the SSCLn pin to the high level. The interval from this time until counting to determine the period at high level starts on the transition of the SSCLn pin to the high level is the total of the delay of SSCLn output, delay for noise filtering of the input on the SSCLn pin (2 or 3 cycles of sampling clock for the noise filter), and delay for internal processing (1 or 2 cycles of PCLK). The period at high level of the internal SSCLn clock is extended even if other devices are not placing the low level on the SSCLn line. If the SIMR2.IICCSC bit 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 SIMR2.IICCSC bit 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 35.55 shows an example of operations to synchronize the clocks.

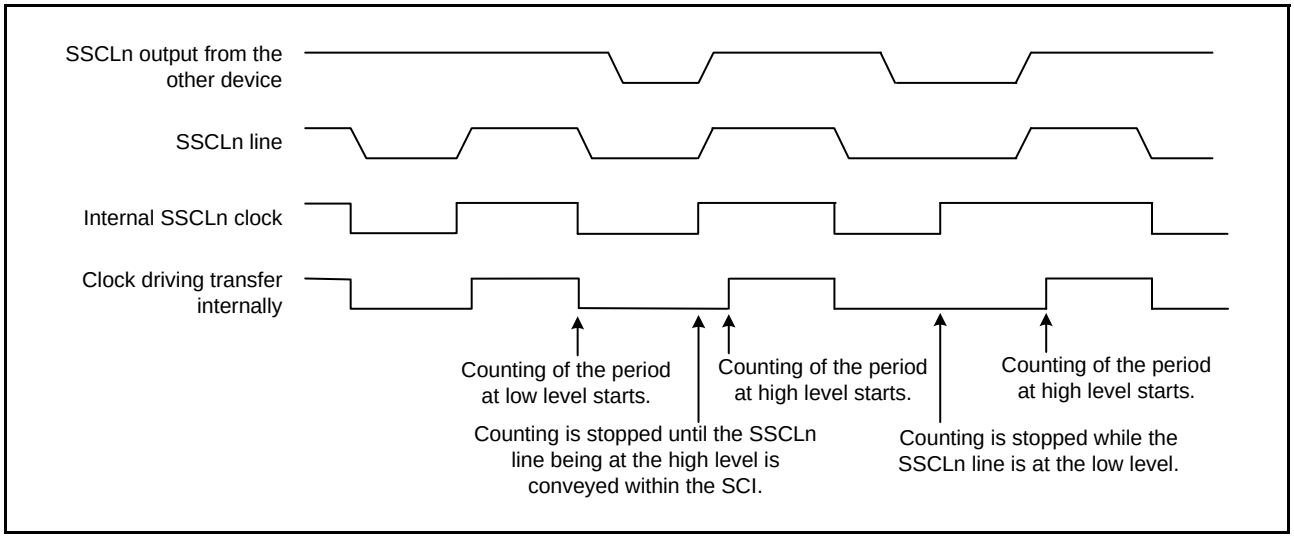


Figure 35.55 Example of Operations for Clock Synchronization

35.7.3 SSDA Output Delay

The SIMR1.IICDL[4:0] bits can be used to set a delay for output on the SSDAn pin relative to falling edges of output on the SSCLn pin. Delay-time settings from 0 to 31 are selectable, representing periods of the corresponding numbers of cycles of the clock signal from the on-chip baud rate generator (derived by frequency-dividing the base clock, PCLK, by the divisor selected by the SMR.CKS[1:0] bits). A delay for output on the SSDAn pin is for the start condition/restart condition/stop condition signal, 8-bit transmit data, and an acknowledge bit.

If the SSDA output delay is shorter than the time for the level on the SSCLn pin to fall, the change of the output on the SSDAn pin will start while the output level on the SSCLn pin is falling, creating a possibility of erroneous operation for slave devices. Ensure that settings for the delay of output on the SSDAn pin are for times greater than the time output on the SSCLn pin takes to fall (300 ns for I^2C-bus in normal mode and fast mode).

Figure 35.56 shows the timing of delays in SSDA output.

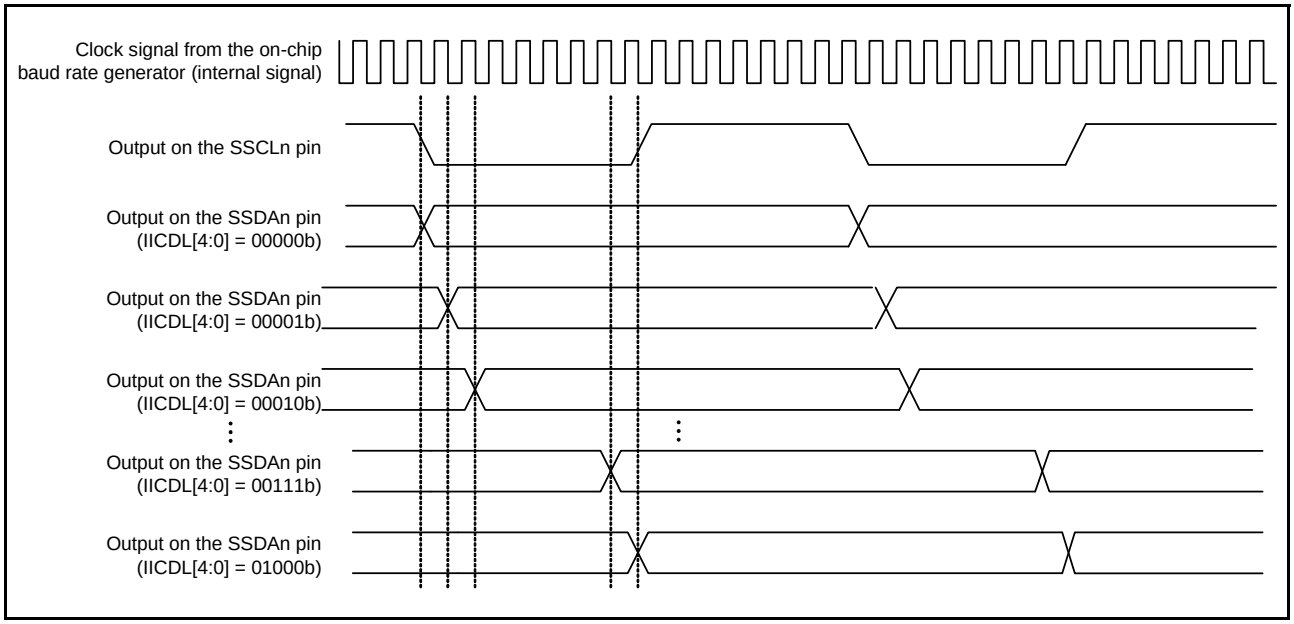


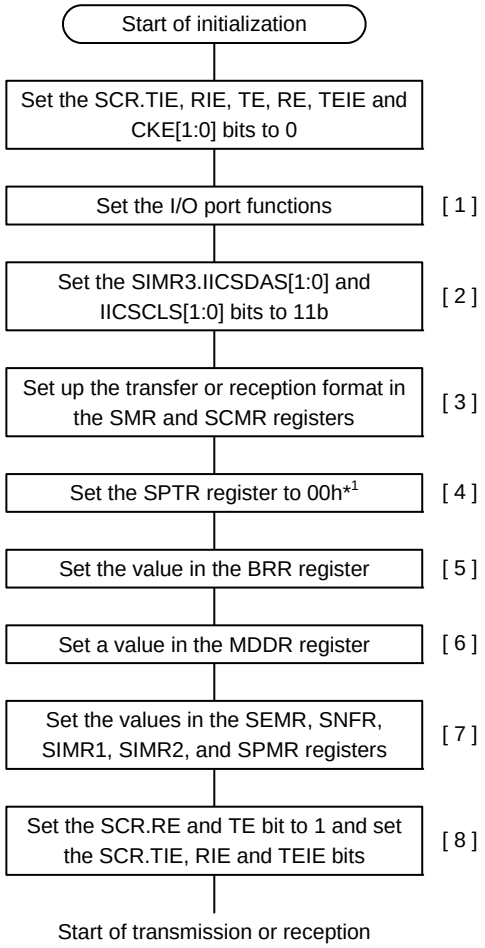
Figure 35.56 Timing of Delays in SSDA Output

35.7.4 SCI Initialization (Simple I2C Mode)

Before transferring data, write the initial value (00h) to the SCR register and initialize the interface following the example shown in Figure 35.57.

When changing the operating mode, transfer format, and so on, be sure to set the SCR register 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 the SMR and SCMR registers.
In the SMR register, set the CKS[1:0] bits to the desired value and set the other bits to 0.
In the SCMR register, set the SDIR bit to 1 and the SINV and SMIF bits to 0.

[4] Set the SPTR register to 00h*1.

[5] Write the value for the desired bit rate to the BRR register.

[6] Write the value obtained by correcting a bit rate error in the MDDR register. This step is not necessary if the SEMR.BRME bit is set to 0.

[7] Set the values in the SEMR, SNFR, SIMR1, SIMR2, and SPMR registers.
In the SEMR register, set the NFEN and BRME bits.
In the SNFR register, set the NFCS[2:0] bits.
In the SIMR1 register, set the IICM bit to 1 and the IICDL[4:0] bits as required.
In the SIMR2 register, set the IICACKT and IICCSC bits to 1 and the IICINTM bits as required.
In the SPMR register, set all the bits to 0.

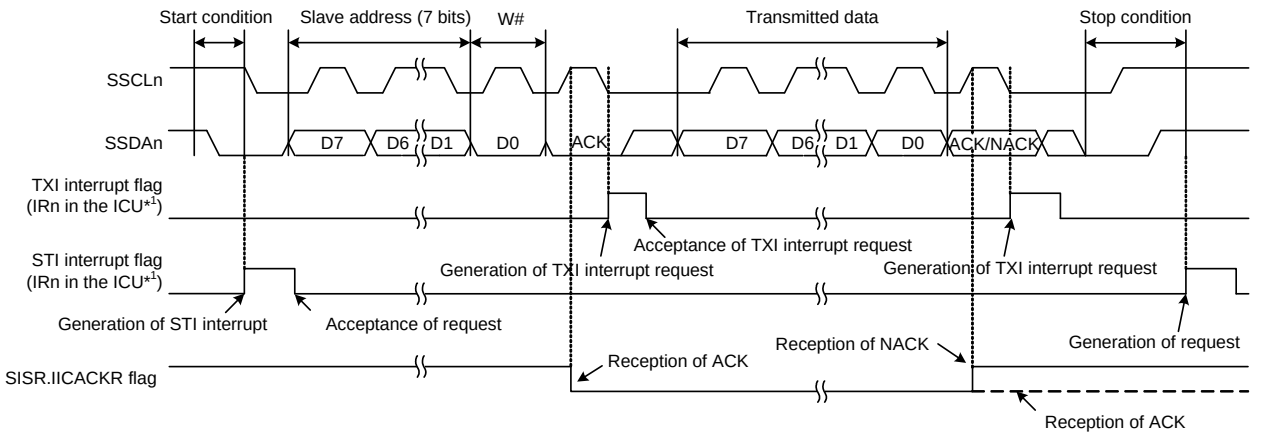
[8] Set the SCR.RE and TE bits to 1. At this time, also set the SCR.TIE, RIE, and TEIE bits (for transmission and when the SIMR2.IICINTM bit is 1, set the RIE bit to 0).
Setting the TE and RE bits to 1 makes the SSCLn and SSDAn pin functions available.

Note: Set the SCR.RE and TE bits to 0 or 1 at the same time.
Note 1. Only for SCI0 to SCI11.

Figure 35.57 Example of the Flowchart of SCI Initialization (for Simple I2C Mode)

35.7.5 Operation in Master Transmission (Simple I2C Mode)

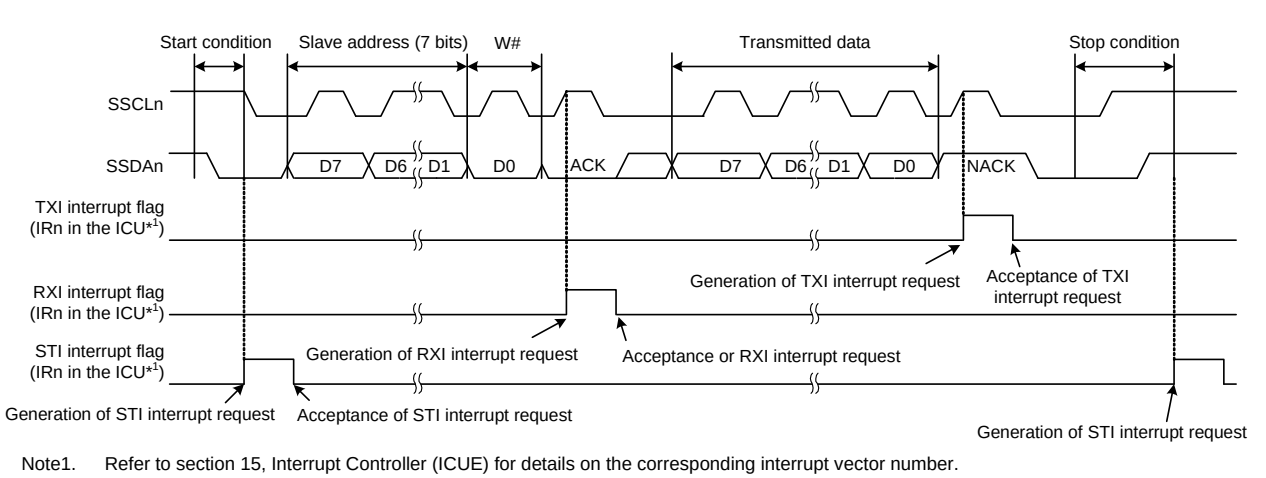
Figure 35.58 and Figure 35.59 show examples of operations in master transmission and Figure 35.60 is a flowchart showing the procedure for data transmission. Refer to Table 35.41 for more information on the STI interrupt.
When 10-bit slave addresses are in use, steps [3] and [4] in Figure 35.60 are repeated twice.
In simple I2C mode, the transmit data empty interrupt (TXI) is generated when communication of one frame is completed, unlike the TXI interrupt request generation timing during clock synchronous transmission.



Note1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

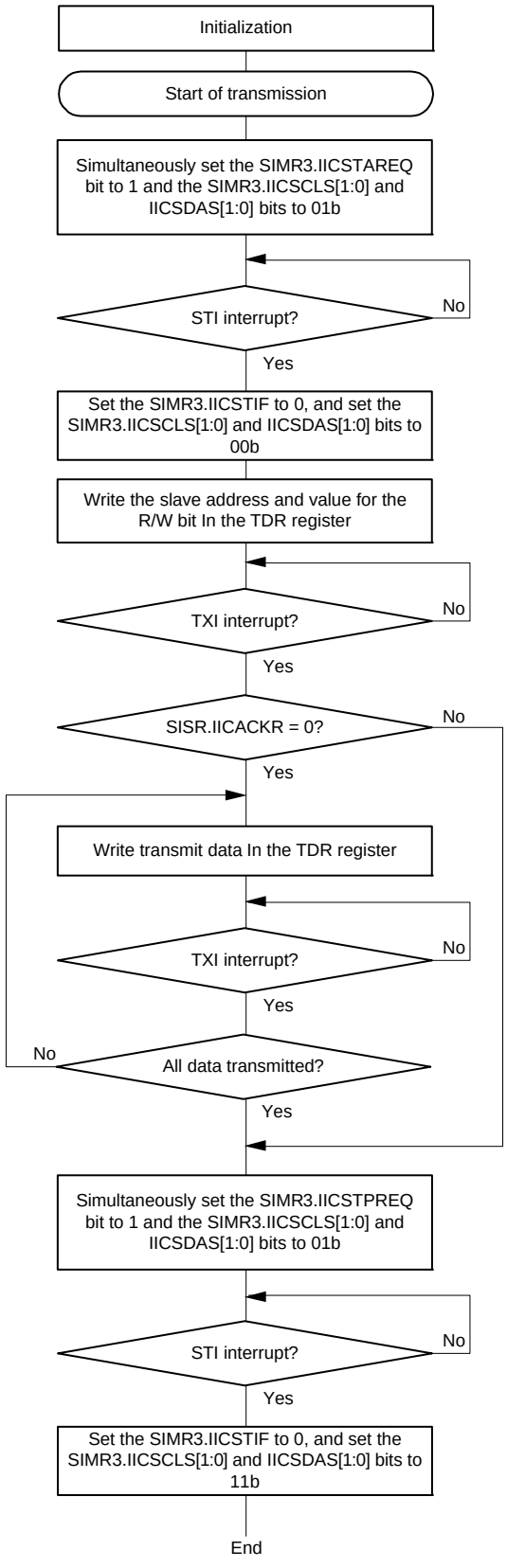
Figure 35.58 Example 1 of Operations for Master Transmission in Simple I2C-bus Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)

When the SIMR2.IICINTM bit is set to 0 (use ACK/NACK interrupts) during master transmission, the DTC or DMAC is activated by the ACK interrupt as the trigger and necessary number of data bytes are transmitted. When the NACK is received, error processing, such as transmission stop and retransmission, is performed by the NACK interrupt as the trigger.



Note1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 35.59 Example 2 of Operations for Master Transmission in Simple I2C-bus Mode (with 7-Bit Slave Addresses, ACK Interrupts, and NACK Interrupts in Use)



[1] [1] Initialization for simple I²C mode
For transmission, set the SCR.RIE bit to 0 (RXI and ERI interrupts requests are disabled)

[2] Generate a start condition.

[3] Writing to the TDR register:
[2] Writing the slave address and value for the R/W bit to the TDR register.

[4] Confirming ACK response from the slave address:
Check the SISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

[5] Steps for continuing with serial transmission:
When transmission is to continue, write further transmit data to the TDR register.
Except for the first data to be transmitted, a TXI request can activate the DMAC or DTC to handle writing of data to the TDR register. In this case, ACK/NACK cannot be confirmed.

[3]

[6] Generation of a stop condition.

[4] If 10-bit slave addresses are in use, processing of [3] and [4] is repeated twice.

[5]

[6]

Note: In simple I²C mode, the TXI interrupt request is generated when communication is completed, unlike the timing during clock synchronous transmission.

Figure 35.60 Example of the Procedure for Master Transmission Operations in Simple I²C Mode (with Transmission Interrupts and Reception Interrupts in Use)

35.7.6 Master Reception (Simple I2C Mode)

Figure 35.61 shows an example of operations in simple I2C mode master reception and Figure 35.62 is a flowchart showing the procedure for master reception.

The value of the SIMR2.IICINTM bit is assumed to be 1 (use reception and transmission interrupts).

In simple I2C mode, the transmit data empty interrupt (TXI) is generated when communication of one frame is completed, unlike the TXI interrupt request generation timing during clock synchronous transmission.

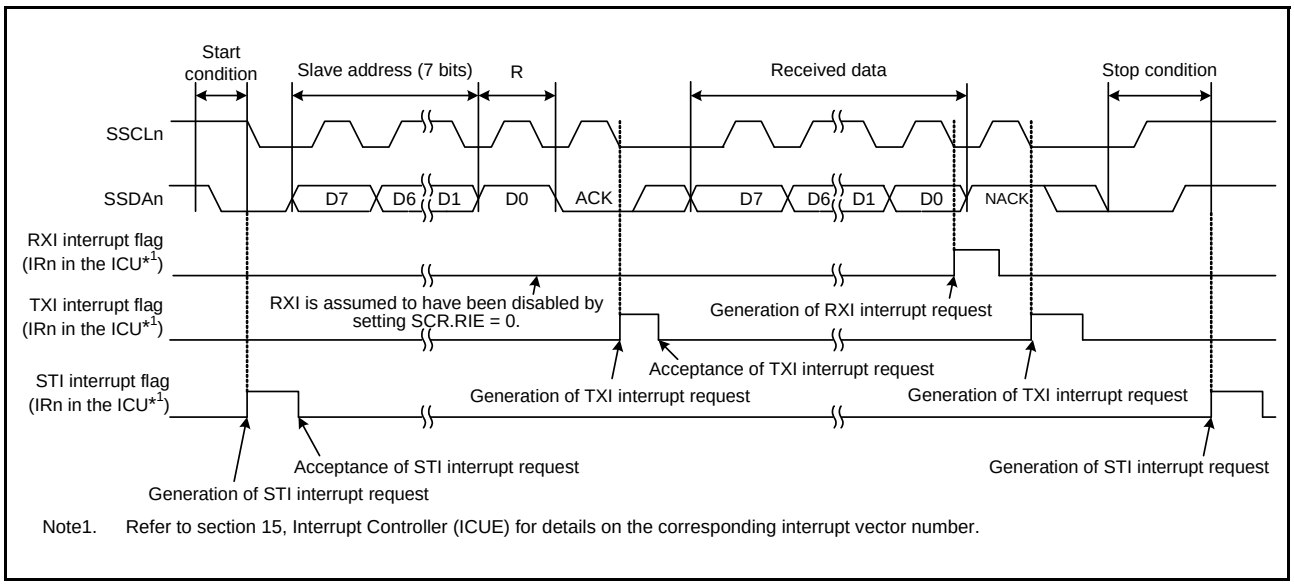


Figure 35.61 Example of Operations for Master Reception in Simple I2C-bus Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)

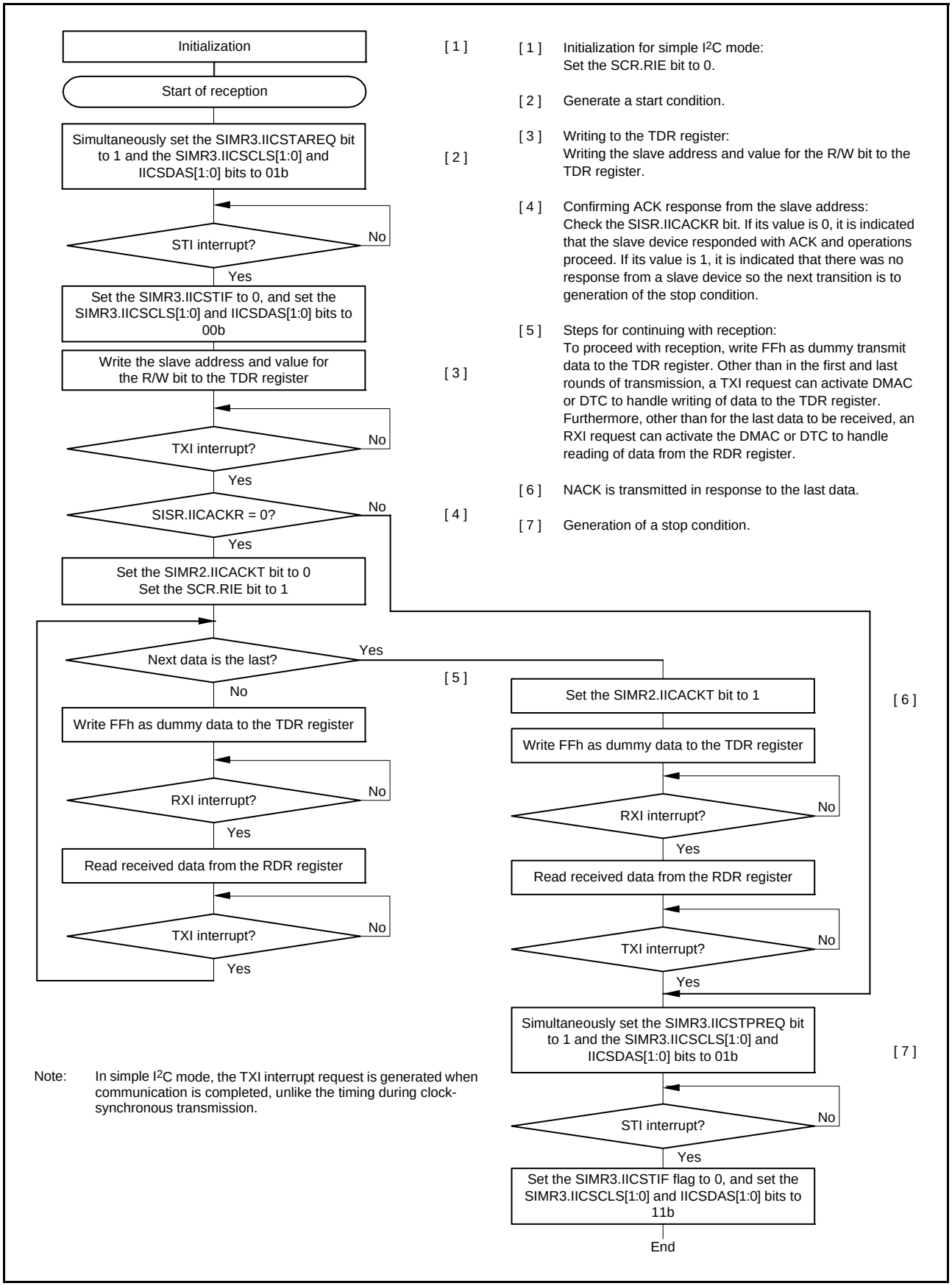


Figure 35.62 Example of the Procedure for Master Reception Operations in Simple I²C Mode (with Transmission Interrupts and Reception Interrupts in Use)

35.7.7 Recovery from Bus Hang-up

If the bus is stuck by an abnormal state in SCI because of the communication error, reset the SCI according to the following steps and release the bus.

(1) Set the SCR.TE and RE bit to 0 at the same time to reset SCI.
(2) Set the SIMR3 register to F0h to release the bus.
(3) If the SSR.RDRF flag is 1, dummy-read the RDR register to clear the flag.
(4) Set the SCR.TE and RE bit to 1 at the same time.

35.8 Operation in Simple SPI Mode

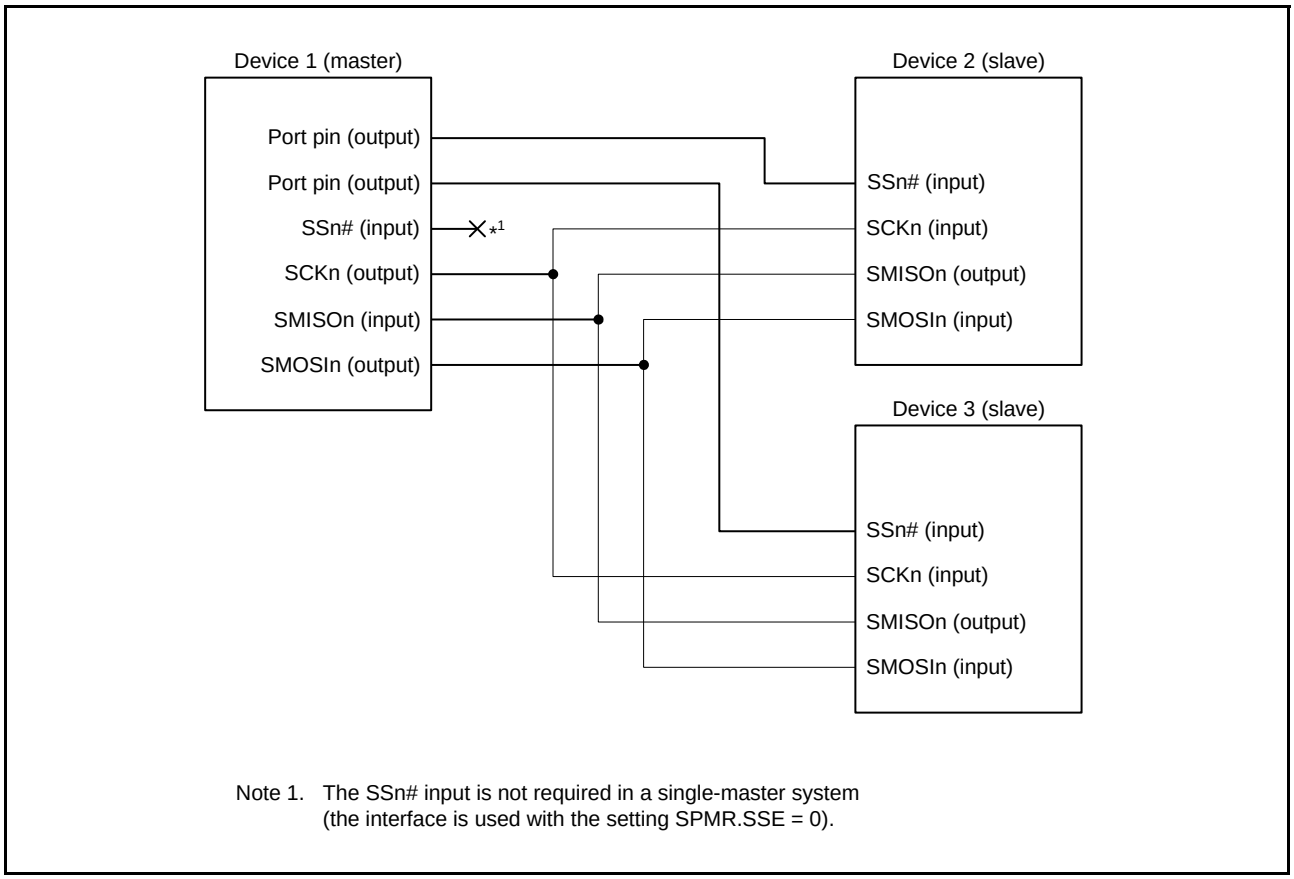
As an extended function, the SCI supports a simple SPI mode that handles transfer among one or multiple master devices and multiple slave devices.

Making the settings for clock synchronous mode (SCMR.SMIF = 0, SIMR1.IICM = 0, SMR.CM = 1) plus setting the SPMR.SSE bit 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 SPMR.SSE bit to 0 in such cases.

Figure 35.63 shows an example of connections for simple SPI mode. Control a general port pin to produce the SS output signal from the master.

In simple SPI mode, data are transferred in synchronization with clock pulses in the same way as in clock synchronous mode. One character of data for transfer consists of 8 bits of data, and parity bits cannot be appended to this. The data can be inverted by setting the SCMR.SINV bit to 1.

Since the receiver and transmitter are independent of each other within the SCI module, full-duplex communications are possible, with a common clock signal. Furthermore, since both the transmitter and receiver have a double-buffered structure, writing of further transmit data while transmission is in progress and reading of previously received data while reception is in progress are both possible. Continuous transfer is thus possible.



Note 1. The SSn# input is not required in a single-master system (the interface is used with the setting SPMR.SSE = 0).

Figure 35.63 Example of Connections via a Simple SPI Mode (In Single Master Mode, SPMR.SSE Bit = 0)

35.8.1 States of Pins in Master and Slave Modes

The direction (input or output) of pins for the simple SPI mode interface differs according to whether the device is a master (SCR.CKE[1:0] = 00b or 01b and SPMR.MSS = 0) or slave (SCR.CKE[1:0] = 10b or 11b and SPMR.MSS = 1). Table 35.36 lists the states of pins according to the mode and the level on the SSn# pin.

Table 35.36 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 cannot proceed)	Input for received data (but disabled)	High-impedance	Clock input (but disabled)
	Low level (transfer can proceed)	Input for received data	Output for data transmission	Clock input

Note 1. When there is only a single master (SPMR.SSE = 0), transfer is possible regardless of the input level on the SSn# pin (this is equivalent to input of a high level on the SSn# pin). Since the SSn# pin function is not required, the pin is available for other purposes.
Note 2. The SMOSIn pin output is in the high-impedance state when serial transmission is disabled (SCR.TE bit = 0).
Note 3. The SCKn pin output is in the high-impedance state when serial transmission is disabled (SCR.TE and RE bits = 00b) in a multi-master configuration (SPMR.SSE = 1).

35.8.2 SS Function in Master Mode

Setting the SCR.CKE[1:0] bits to 00b and the SPMR.MSS bit to 0 selects master operation. The SSn# pin is not used in single-master configurations (SPMR.SSE = 0), so transmission or reception can proceed regardless of the value of the SSn# pin.

When the level on the SSn# pin is high in a multi-master configuration (SPMR.SSE = 1), a master device outputs clock signals from the SCKn pin before starting transmission or reception to indicate that there are no other masters or another master is performing reception or transmission. When the level on the SSn# pin is low in a multi-master configuration (SPMR.SSE = 1), there are other masters, and this indicates that transmission or reception is in progress. At this time the SMOSIn output and SCKn pins will be placed in the high-impedance state and starting transmission or reception will not be possible. Furthermore, the value of the SPMR.MFF bit will be 1, indicating a mode fault error. In a multi-master configuration, start error processing by reading SPMR.MFF flag. Also, even if a mode fault error occurs while transmission or reception is in progress, transmission or reception will not be stopped, but the SMOSIn and SCKn pin output will be placed in the high-impedance state after the completion of the transfer.

Control a general port pin to produce the SS output signal from the master.

35.8.3 SS Function in Slave Mode

Setting the SCR.CKE[1:0] bits to 10b and the SPMR.MSS bit to 1 selects slave operation. When the level on the SSn# pin is high, the SMISOn output pin will be in the high-impedance state and clock input through the SCKn pin will be ignored. When the level on the SSn# pin is low, clock input through the SCKn pin will be effective and transmission or reception can proceed.

If the input on the SSn# pin changes from low to high level during transmission or reception, the SMISOn output pin will be placed in the high-impedance state. Meanwhile, the internal processing for transmission or reception will continue at the rate of the clock input through the SCKn pin until processing for the character currently being transmitted or received is completed, after which it stops. At that time, an interrupt (the appropriate one from among TXI, RXI, and TEI) will be generated.

35.8.4 Relationship between Clock and Transmit/Receive Data

The CKPOL and CKPH bits in the SPMR register 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 35.64. The relation is the same for both master and slave operation.

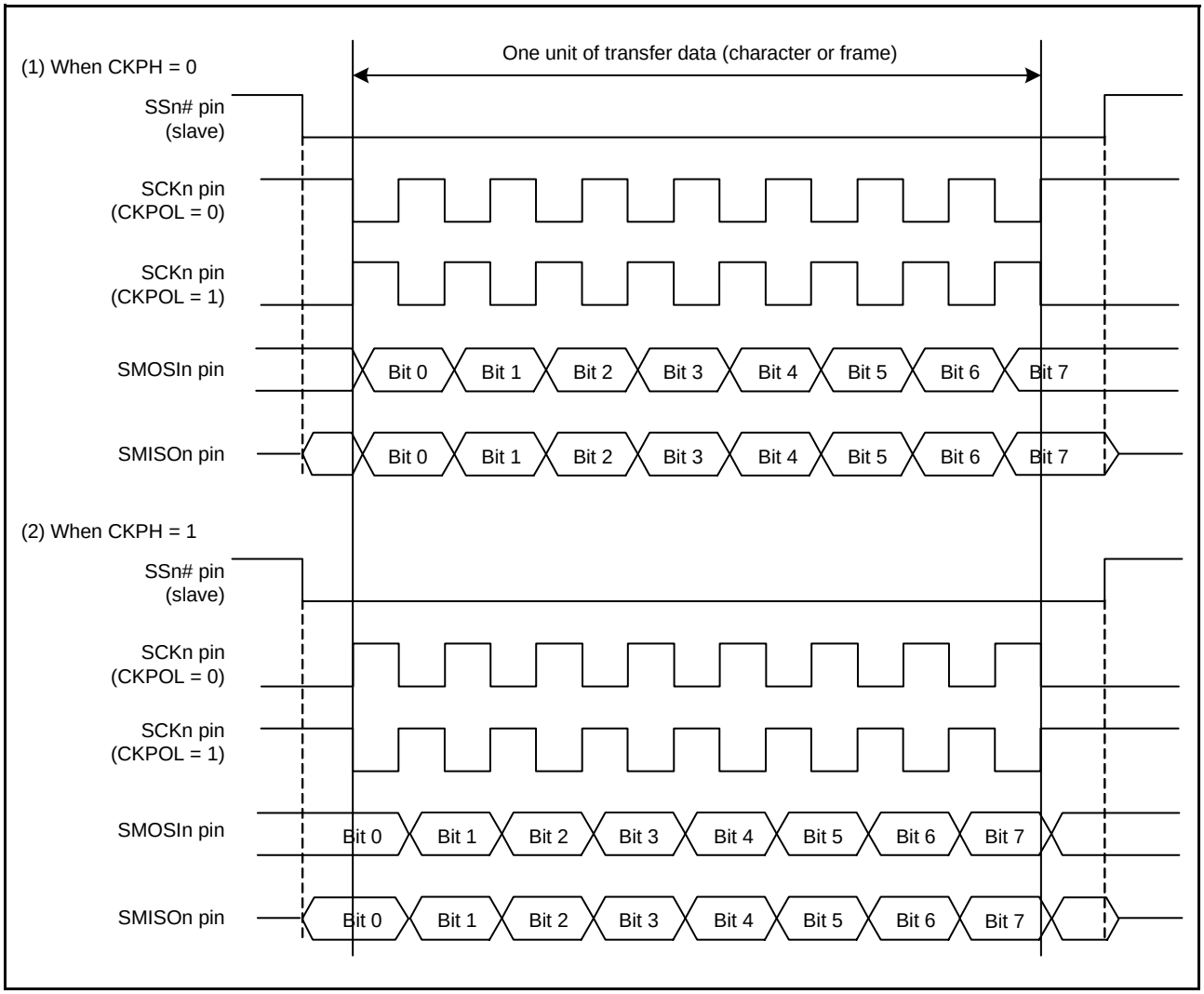


Figure 35.64 Relation between Clock Signal and Transmit/Receive Data in Simple SPI Mode

35.8.5 SCI Initialization (Simple SPI Mode)

The procedure is the same as for initialization in clock synchronous mode Figure 35.30, Sample SCI Initialization Flowchart. The CKPOL and CKPH bits in the SPMR register must be set to ensure that the kind of clock signal they select is suitable for both master and slave devices.
For initialization, changes to the operating mode, changes to the transfer format, and so on, initialize the SCR register before proceeding with changes.
As well as setting the RE bit to 0, note that the SSR.ORER, FER, and PER flags, as well as the RDR register, are not initialized.

35.8.6 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.

35.9 Bit Rate Modulation Function

The bit rate modulation function corrects the bit rate by thinning out the specified amount of clocks from those input to the baud rate generator.
When the SEMR.BRME bit is 1, the baud rate generator validates and counts the average interval of the number of clocks set in the MDDR register out of the total 256 clocks input.
Figure 35.65 assumes the SCI is in asynchronous mode, bits SMR.CKS[1:0] are 00b, the BRR register is 00h, and the MDDR register is 160. In this example, the cycle of the base clock is evenly corrected to 256/160, and the bit rate is corrected to 160/256. Note that there is an imbalance in thinning out the internal clock, and expansion and contraction occur in the pulse width of the base clock.

Note: Do not use this function in the highest speed settings (SMR.CKS[1:0] = 00b, SCR.CKE[1] = 0, and BRR = 0) in clock synchronous mode and simple SPI mode.

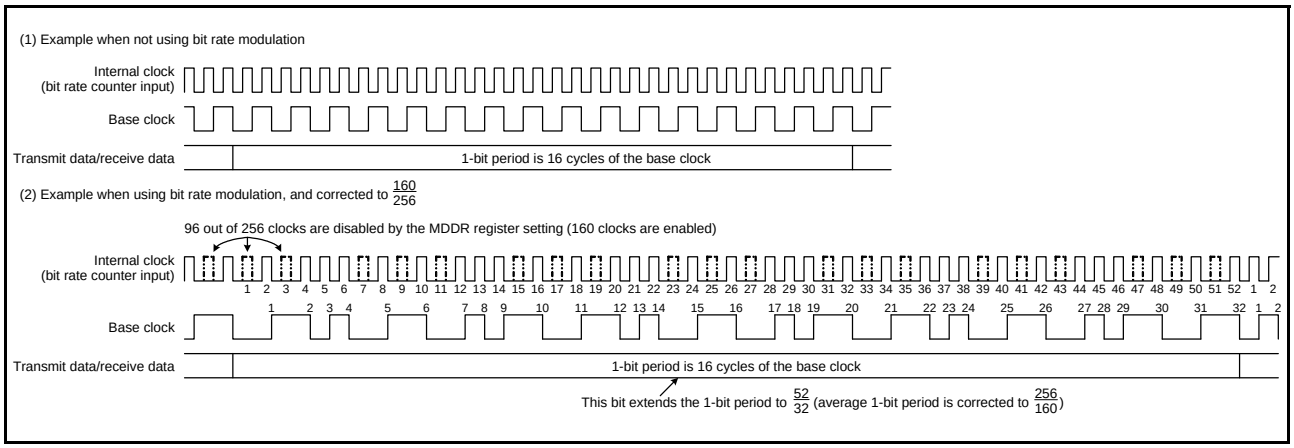


Figure 35.65 Example of the Base Clock When the Bit Rate Modulation Function is Used

The input of a clock signal with a shorter period to the baud rate generator reduces difference in the generated base clock period and, since the division ratio of the baud rate generator also becomes larger, reduces difference in the length of the 1-bit period.

35.10 Extended Serial Mode Control Section: Description of Operation

35.10.1 Serial Transfer Protocol

The extended serial mode control section of the SCI12 can realize the serial transfer protocol composed of Start Frames and Information Frames that is shown in Figure 35.66.
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.

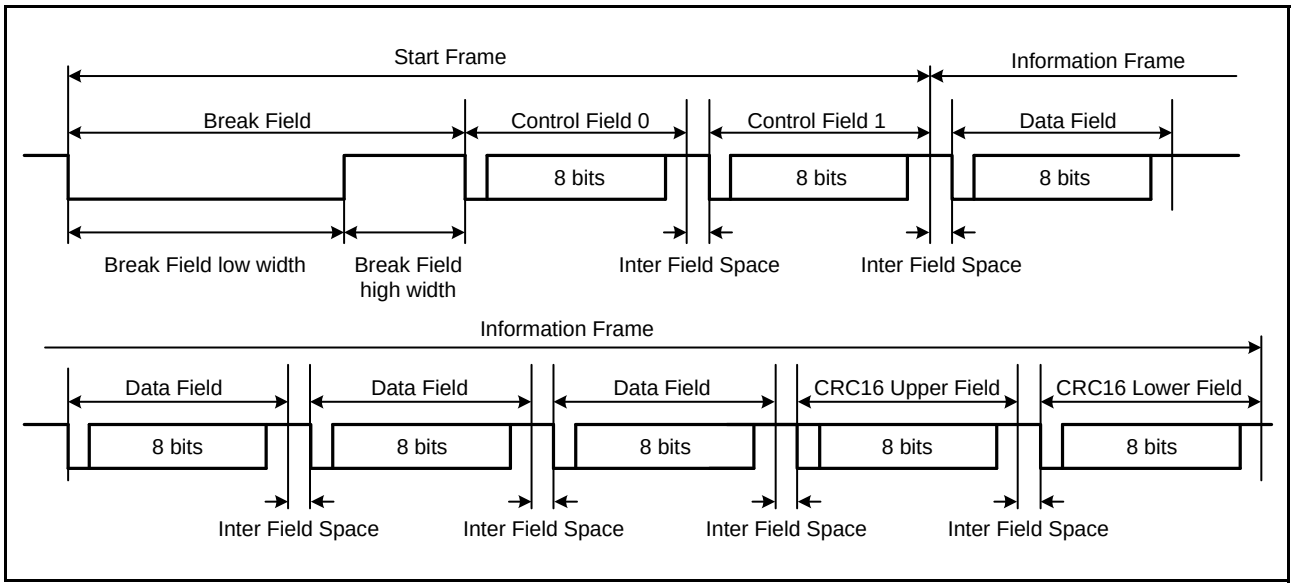


Figure 35.66 Protocol for Serial Transfer by the Extended Serial Mode Control Section

35.10.2 Transmitting a Start Frame

Figure 35.67 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 35.68 and Figure 35.69 are flowcharts for the transmission of a Start Frame.
Operations when the extended serial mode control section is to be used to transmit a Start Frame are as listed below. Be sure to use the SCI12 in asynchronous mode.

(1) With Break Field low width output mode as the operating mode for the timer, writing 1 to the TCR.TCST bit starts counting by the timer, and the low level will be output from the TXDX12 pin over the period corresponding to registers TCNT and TPRE settings.
(2) The output on the TXDX12 pin is inverted when the timer counter underflows, and the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1.
(3) Write 0 to the TCR.TCST bit to stop counting by the timer, and send the data for Control Field 0. After the Break Field low width output, stop counting before the next underflow occurs.
(4) When the data for Control Field 0 have been transmitted, the data for Control Field 1 is transmitted.
(5) When the data for Control Field 1 have been transmitted, an Information Frame is transmitted.

Omit the Break Field and Control Field 0 to suit the structure of the Start Frame.

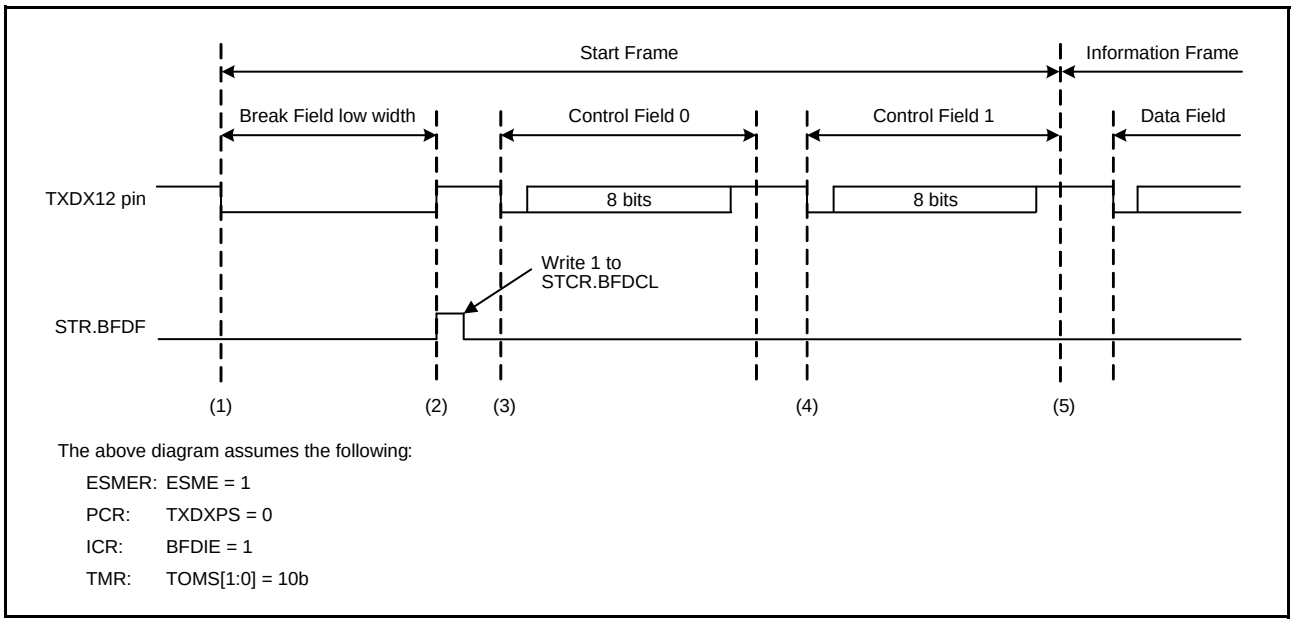


Figure 35.67 Example of Operations When Transmitting a Start Frame

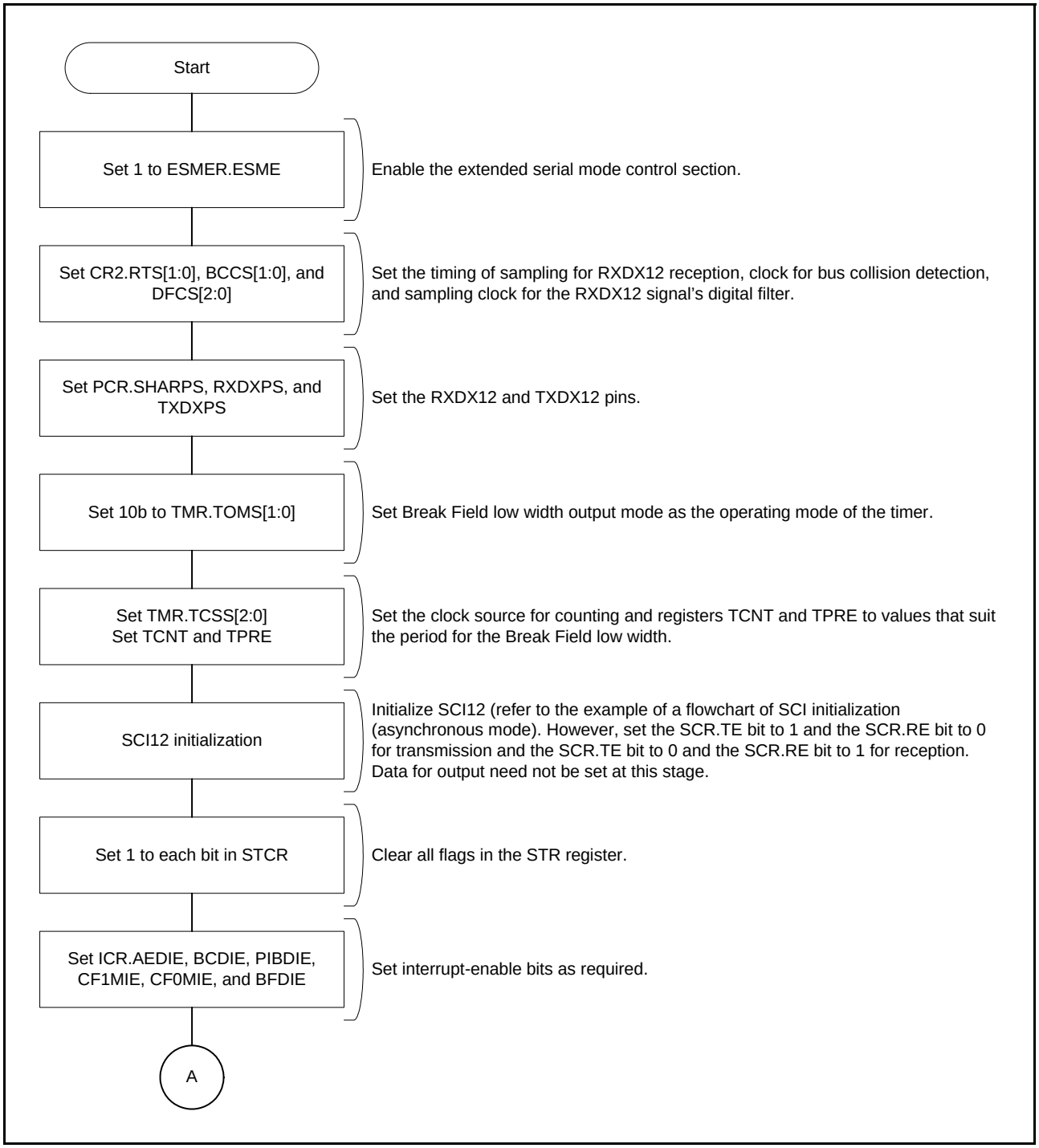


Figure 35.68 Example of Start Frame Transmission (1/2)

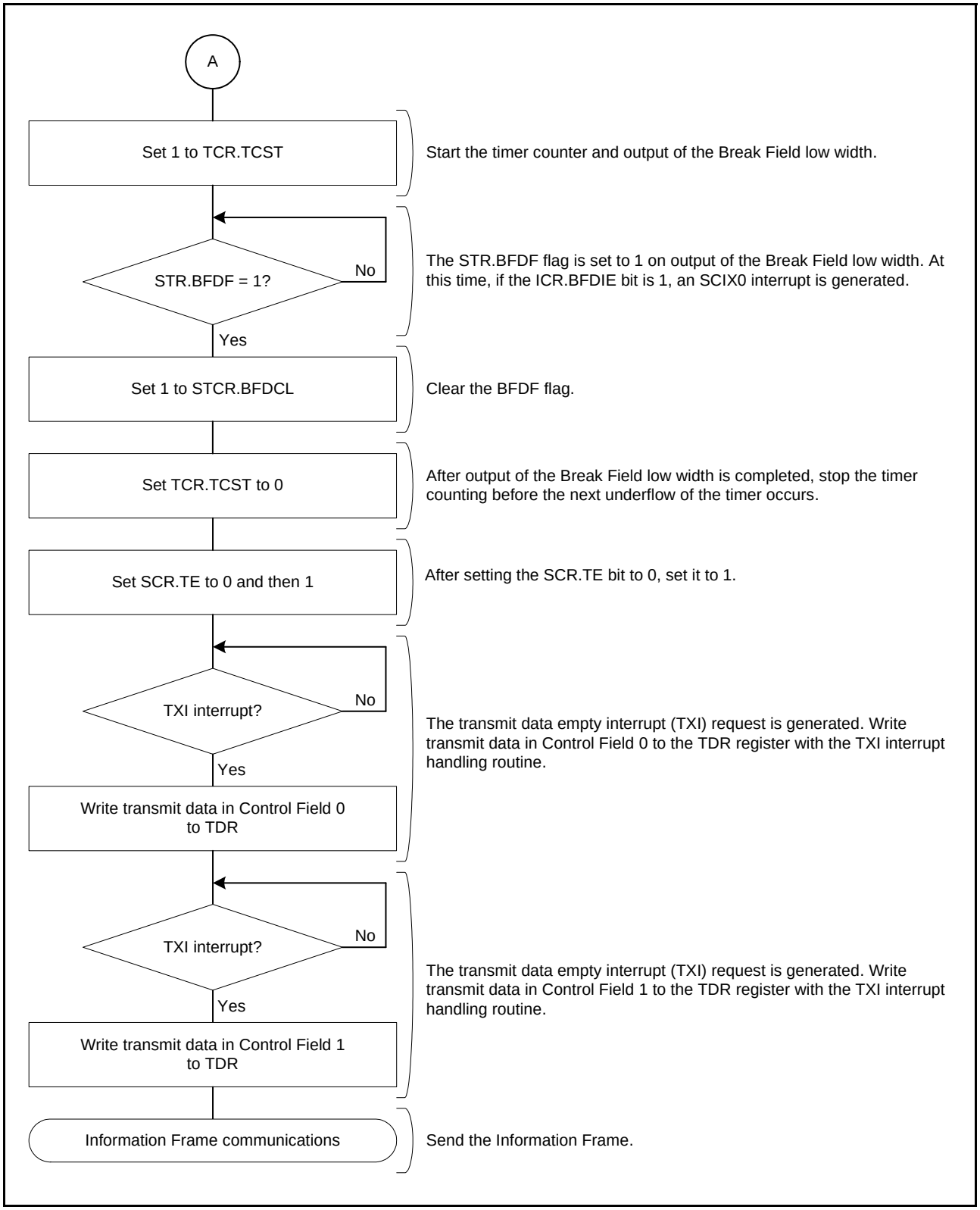


Figure 35.69 Example of Start Frame Transmission (2/2)

35.10.3 Receiving a Start Frame

The extended serial mode control section is capable of receiving Start Frames with the structures listed in Table 35.37.

Table 35.37 Structures of Start Frames

Bit Setting		Structures of Start Frames
BFE	CF0RE	
0	0	Control Field 1 (8 bits) → Information Frame
0	1	Control Field 0 (8 bits) → Control Field 1 (8 bits) → Information Frame
1	0	Break Field (low width) → Control Field 1 (8 bits) → Information Frame
1	1	Break Field (low width) → Control Field 0 (8 bits) → Control Field 1 (8 bits) → Information Frame

Figure 35.70 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 35.71 and Figure 35.72 are flowcharts for the reception of a Start Frame, and Figure 35.73 is a state transition diagram when receiving a Start Frame.

Operations when the extended serial mode control section is to be used to receive a Start Frame are as listed below. Be sure to use the SCI12 in asynchronous mode.

(1) With Break Field low width detection mode as the operating mode for the timer, writing 1 to the CR3.SDST bit enables detection of the Break Field low width.

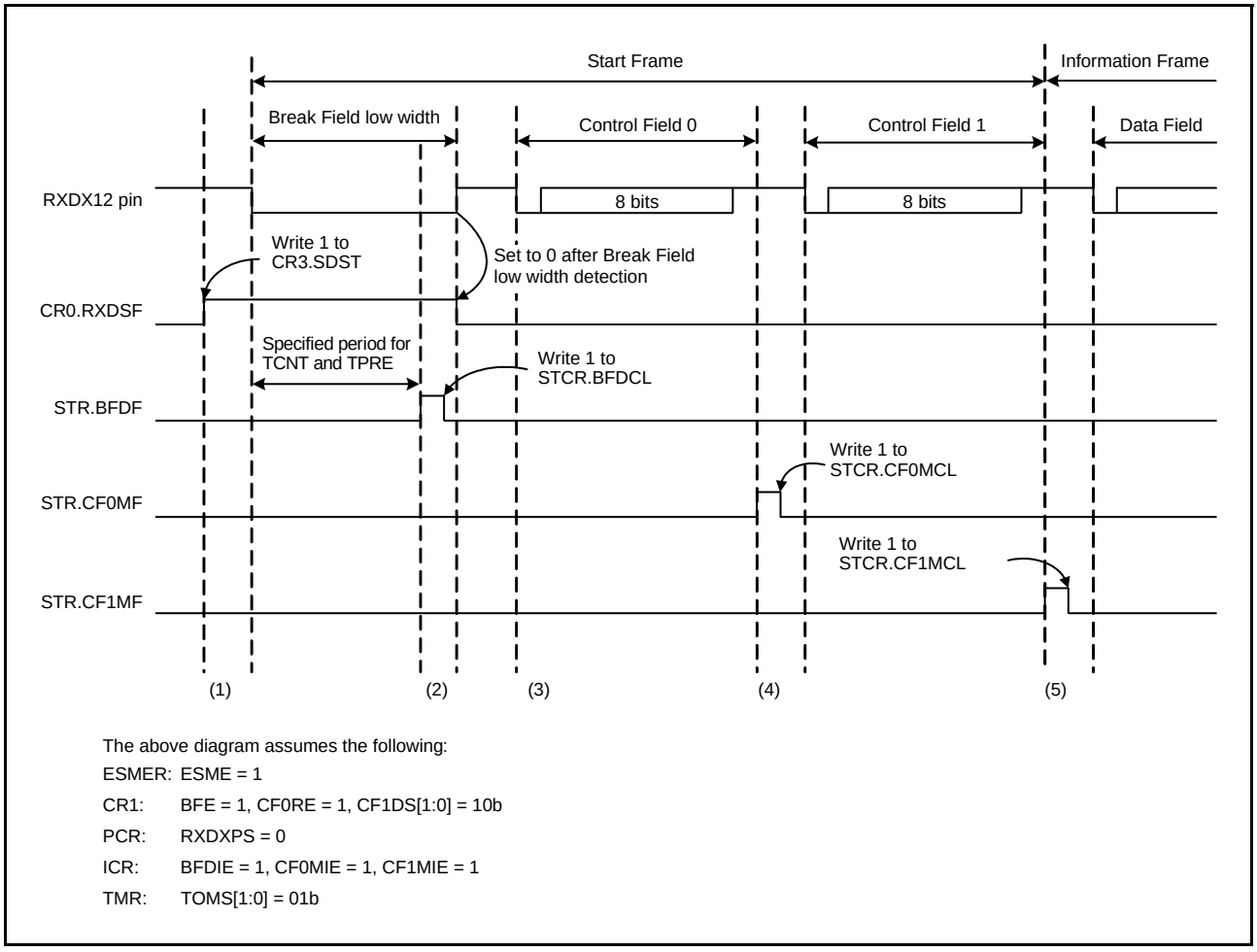
(2) Low-level input on the RXDX12 pin continuing over a period longer than that corresponding to the settings of registers TCNT and TPRE is detected as the Break Field low width. At this time, the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1.

(3) When the input from the RXDX12 pin goes high after the Break Field low width, the CR0.RXDSF flag becomes 0 and reception of Control Field 0 starts.

(4) If the data received in Control Field 0 match the data set in the CF0DR register, the STR.CF0MF flag is set to 1. An SCIX1 interrupt is also generated if the value of the ICR.CF0MIE bit is 1. Reception of Control Field 1 starts after that. If the data received in Control Field 0 do not match the data set in the CF0DR register, a transition to the state prior to Break Field low width detection proceeds.

(5) If the data received in Control Field 1 match the data set in registers PCF1DR and SCF1DR, the STR.CF1MF flag is set to 1. An SCIX1 interrupt is also generated if the value of the ICR.CF1MIE bit is 1. Transfer of the Information Frame starts after that. If the data received in Control Field 1 do not match the data set in either or both of registers PCF1DR and SCF1DR, a transition to the state prior to Break Field low width detection proceeds.

Omit the Break Field and Control Field 0 to suit the structure of the Start Frame.



The above diagram assumes the following:

ESMER: ESME = 1

CR1: BFE = 1, CF0RE = 1, CF1DS[1:0] = 10b

PCR: RXDXPS = 0

ICR: BFDIE = 1, CF0MIE = 1, CF1MIE = 1

TMR: TOMS[1:0] = 01b

Figure 35.70 Example of Operations at the Time of Start Frame Reception

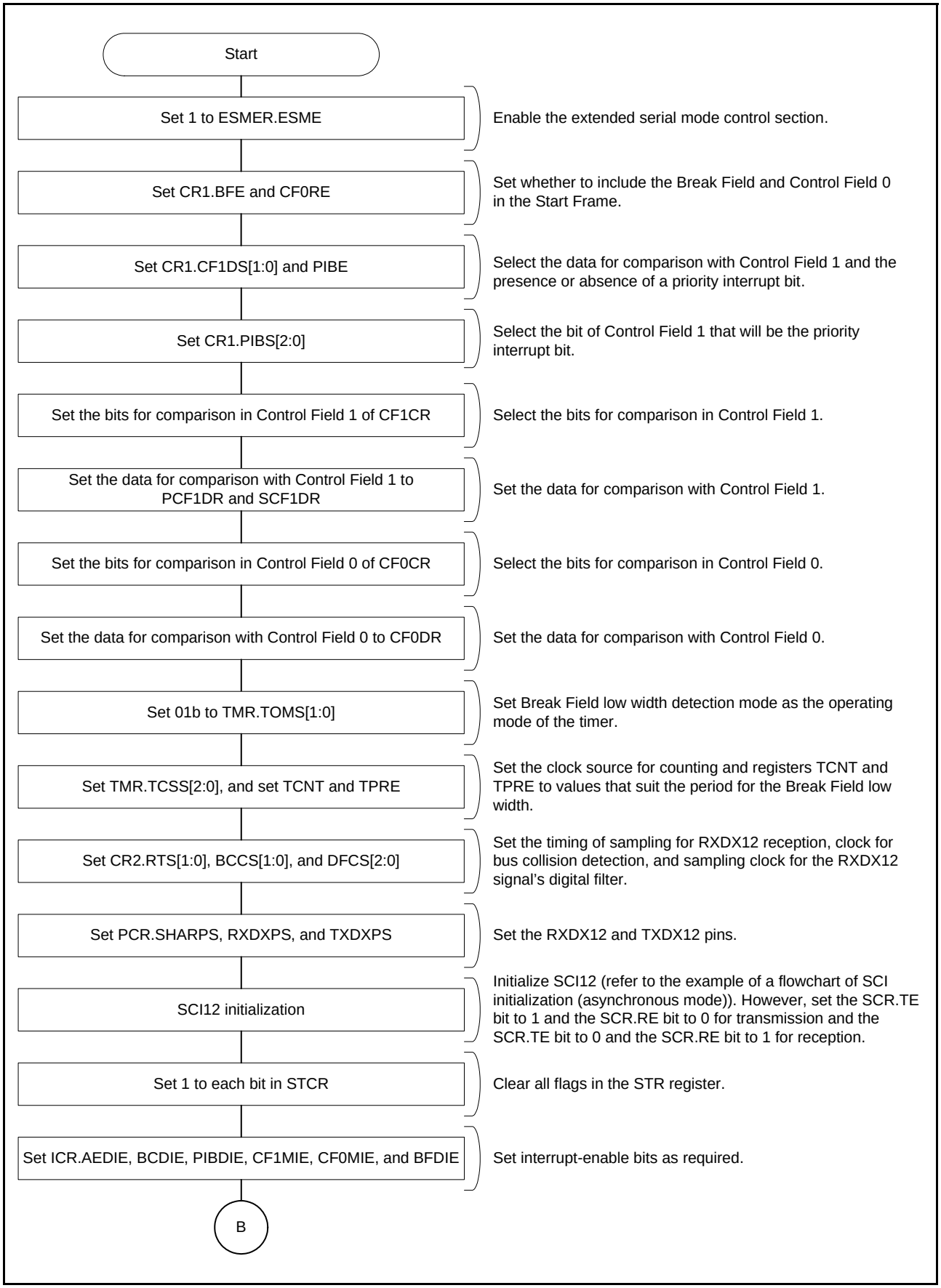


Figure 35.71 Sample Flowchart for Reception of a Start Frame (1)

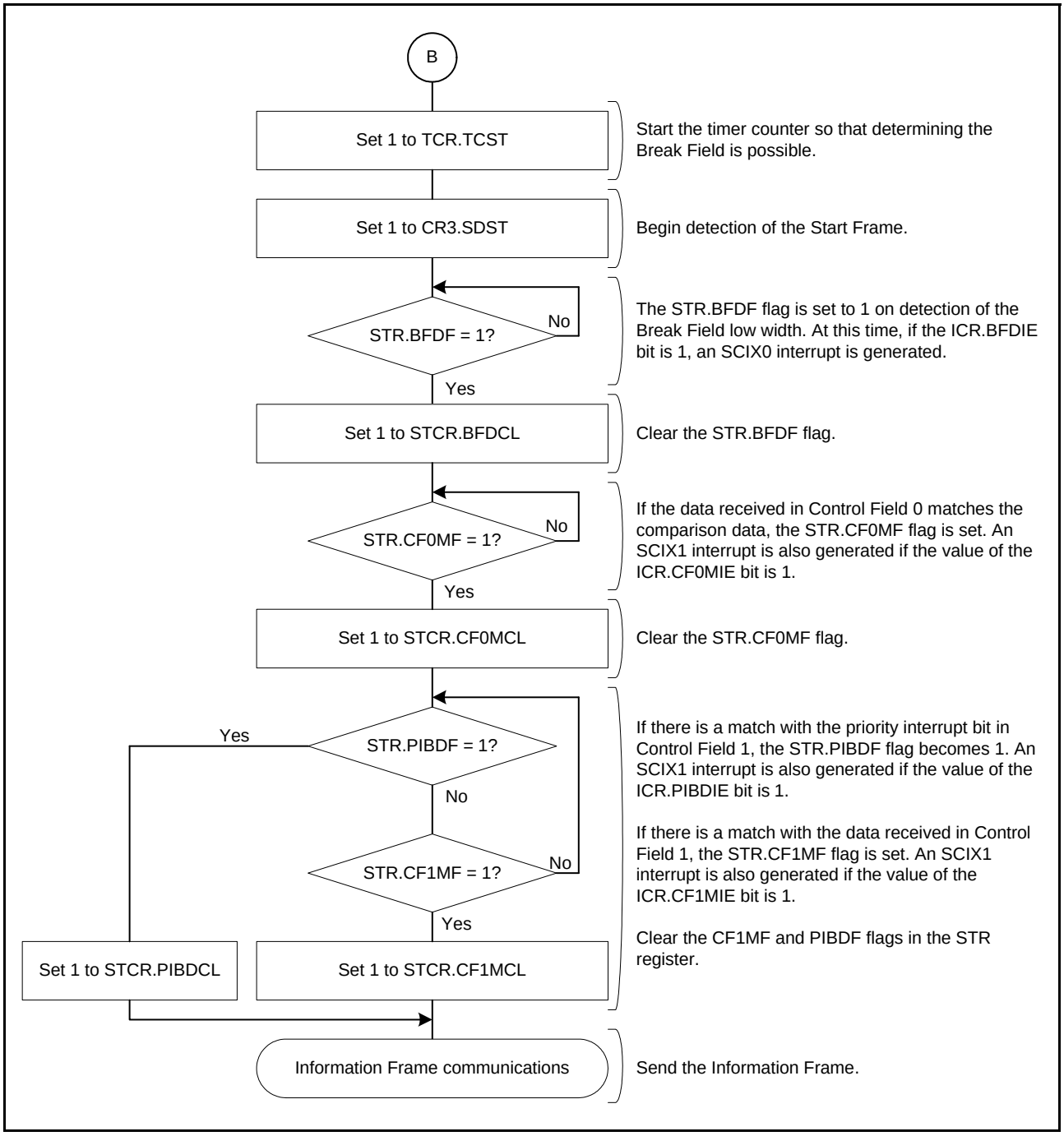


Figure 35.72 Sample Flowchart for Reception of a Start Frame (2)

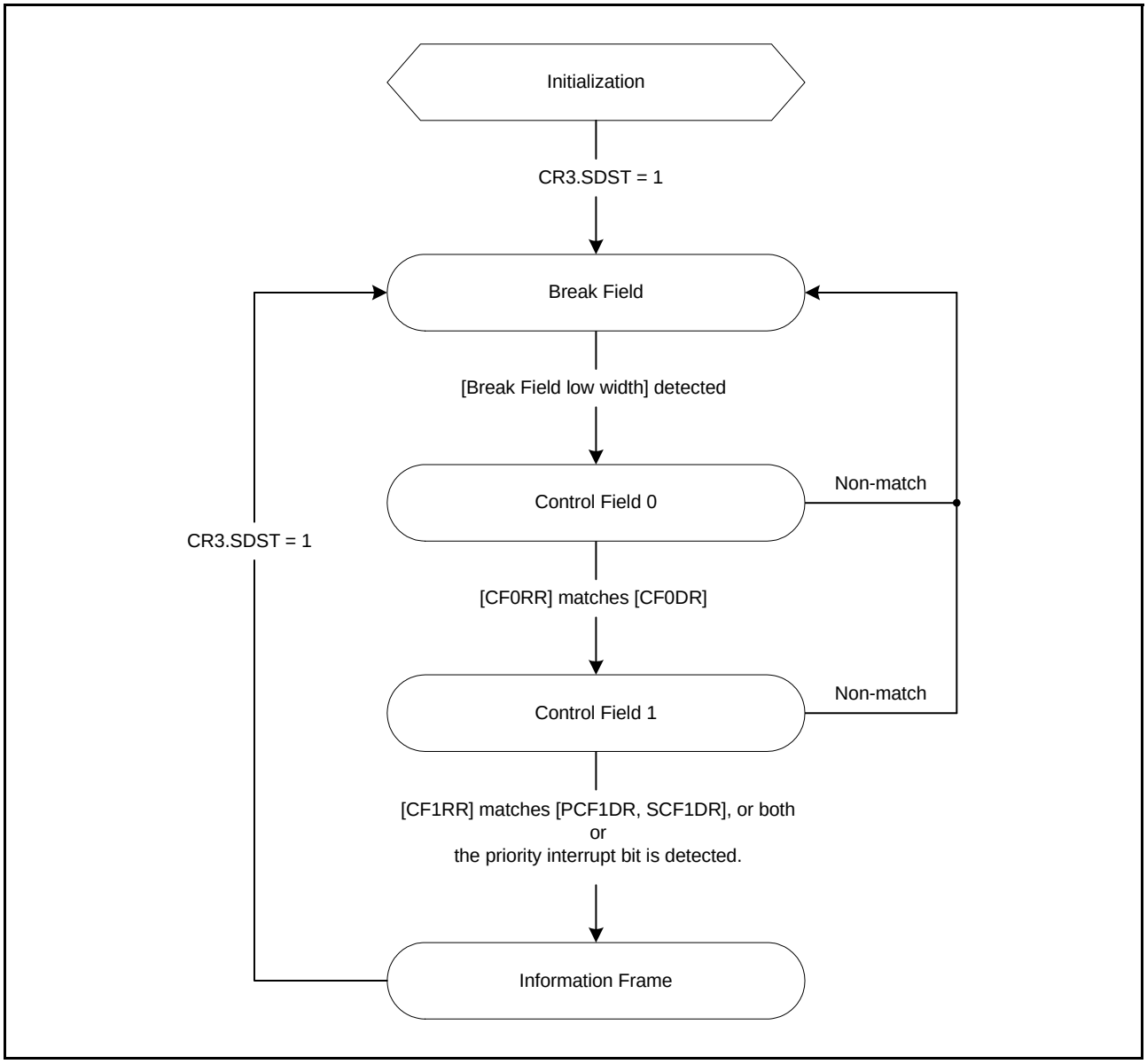


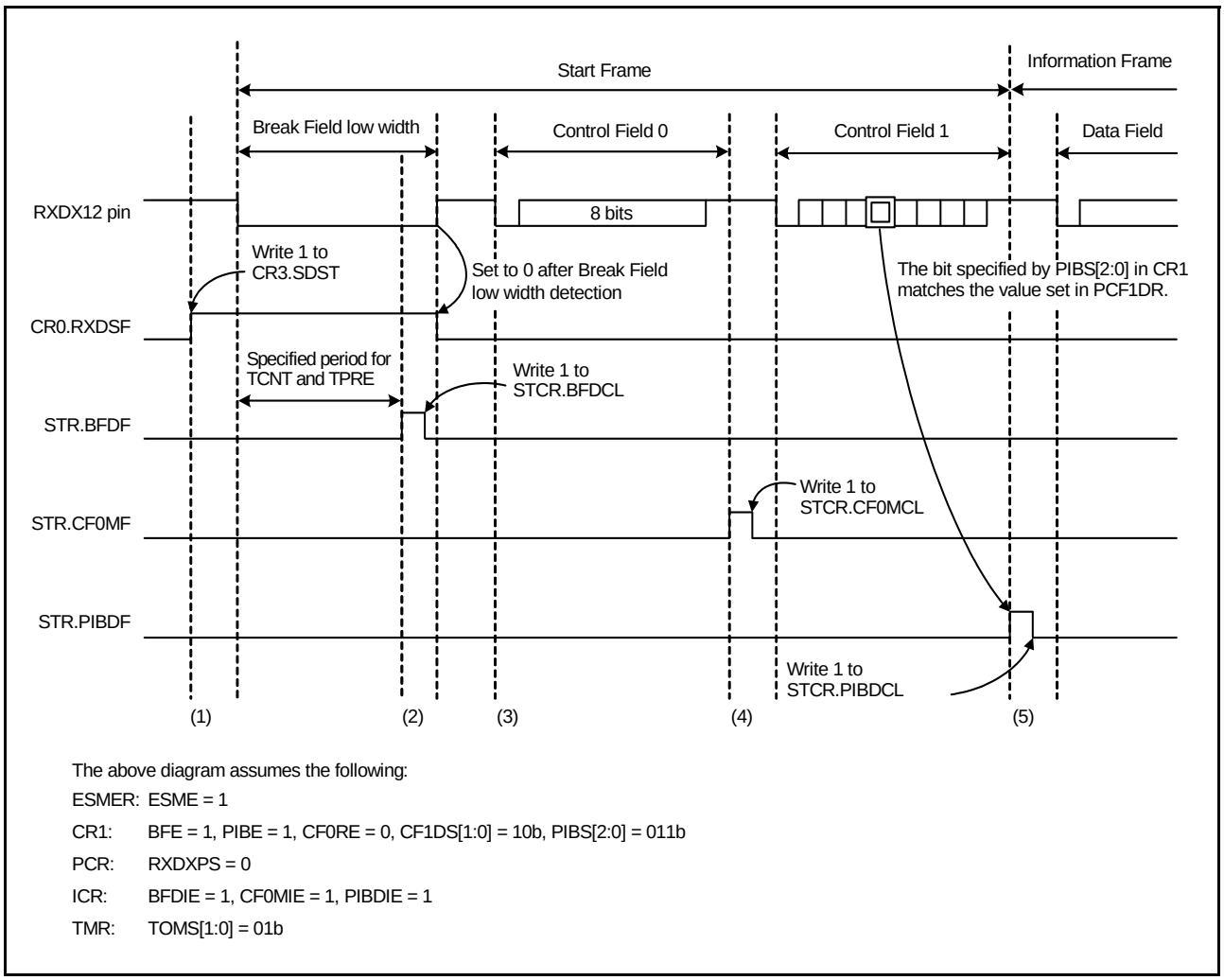
Figure 35.73 State Transitions When Receiving a Start Frame

35.10.3.1 Priority Interrupt Bit

Figure 35.74 shows an example of operation in Start Frame reception where a priority interrupt bit is in use. Setting the CR1.PIBE bit to 1 enables the use of a priority interrupt bit.
Operations of the extended serial mode control section in start Frame reception where a priority interrupt bit is in use are as described below.

Steps (1) to (4) are the same as in Figure 35.70, for Start Frame reception.

(5) If the value of the bit selected by the CR1.PIBS[2:0] bits matches the corresponding bit in the PCF1DR register, the STR.PIBDF flag is set to 1. An SCIX1 interrupt is also generated if the value of the ICR.PIBDIE bit is 1. Transfer of the Information Frame starts after that. If the data received in Control Field 1 do not match the data set in either or both of registers PCF1DR and SCF1DR and the priority interrupt bit is not detected, a transition to the state prior to Break Field low width detection proceeds.



The above diagram assumes the following:

ESMER: ESME = 1
CR1: BFE = 1, PIBE = 1, CF0RE = 0, CF1DS[1:0] = 10b, PIBS[2:0] = 011b
PCR: RXDXPS = 0
ICR: BFDIE = 1, CF0MIE = 1, PIBDIE = 1
TMR: TOMS[1:0] = 01b

Figure 35.74 Example of Operations When Receiving a Start Frame While the CR1.PIBE Bit is 1

35.10.4 Detection of Bus Collisions

Detection of bus collisions operate for cases where output of the Break Field low width and transmission of data are in progress when the ESMER.ESME bit and the SCI.TE bit are set to 1.

Figure 35.75 shows an example of operations with bus collision detection. Signals output through TXDX12 and input through RXDX12 are sampled with the bus collision detection clock set with the CR2.BCCS[1:0] bits as the sampling clock, and the STR.BCDF flag is set to 1 if the signals fail to match three times in a row. An SCIX2 interrupt is also generated if the value of the ICR.BCDIE bit is 1.

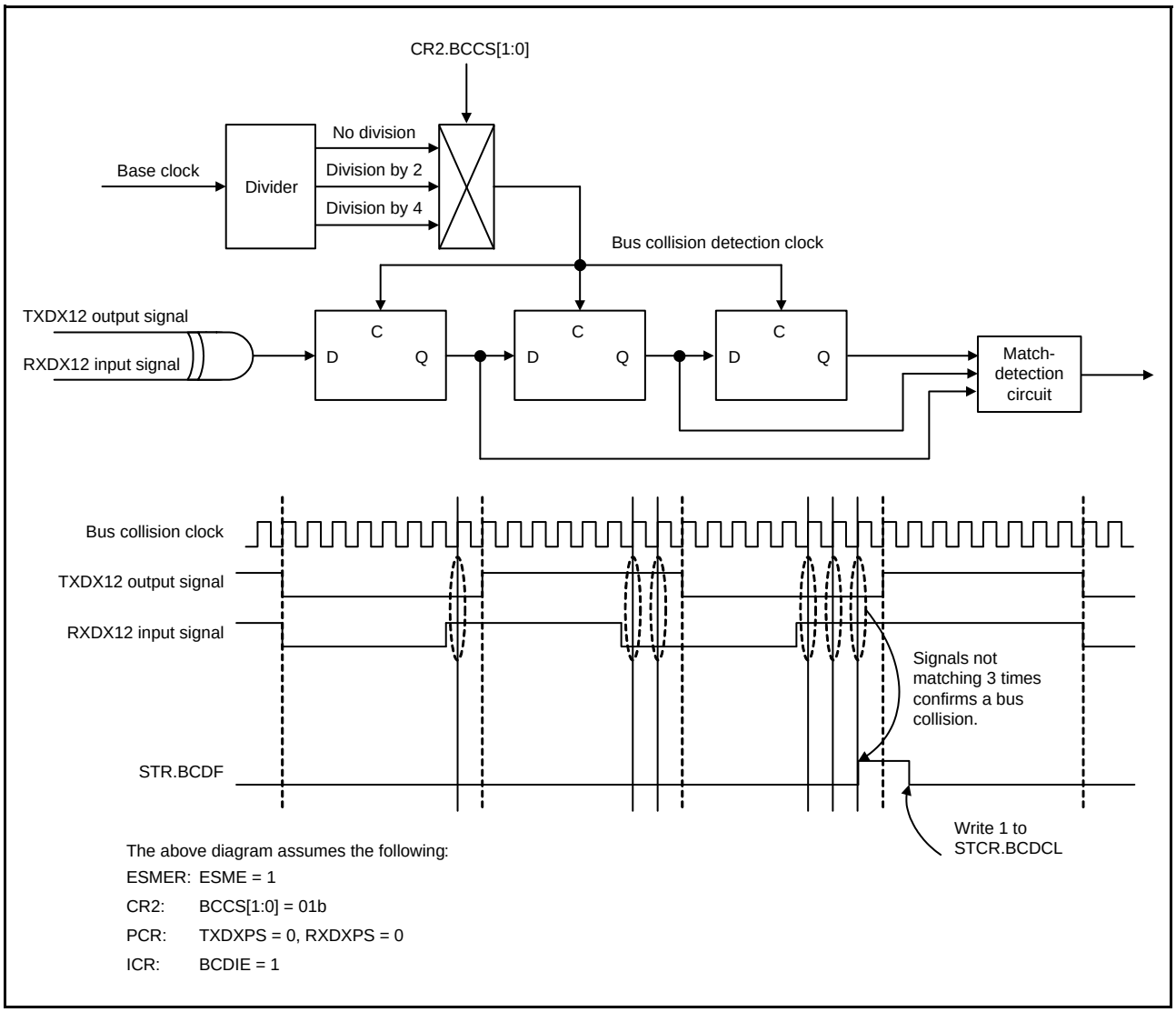


Figure 35.75 Example of Operations with Bus Collision Detection

35.10.5 Digital Filter for Input on the RXDX12 Pin

Signals input through the RXDX12 pin can be passed through a digital filter before they are conveyed to the internal circuits. The digital filter consists of three flip-flop circuit stages connected in series and a match-detecting circuit. The CR2.DFCS[2:0] bits select the sampling clock for the RXDX12 pin input signals. If the outputs of all three latches match, the given level is conveyed to subsequent circuits. If the levels do not match, the previous value is retained. In other words, levels are confirmed as being the signal if they are retained for at least three cycles of the sampling clock but judged to be noise rather than changes in the signal level if they change within three cycles of the sampling clock. Figure 35.76 shows an example of operations with the digital filter.

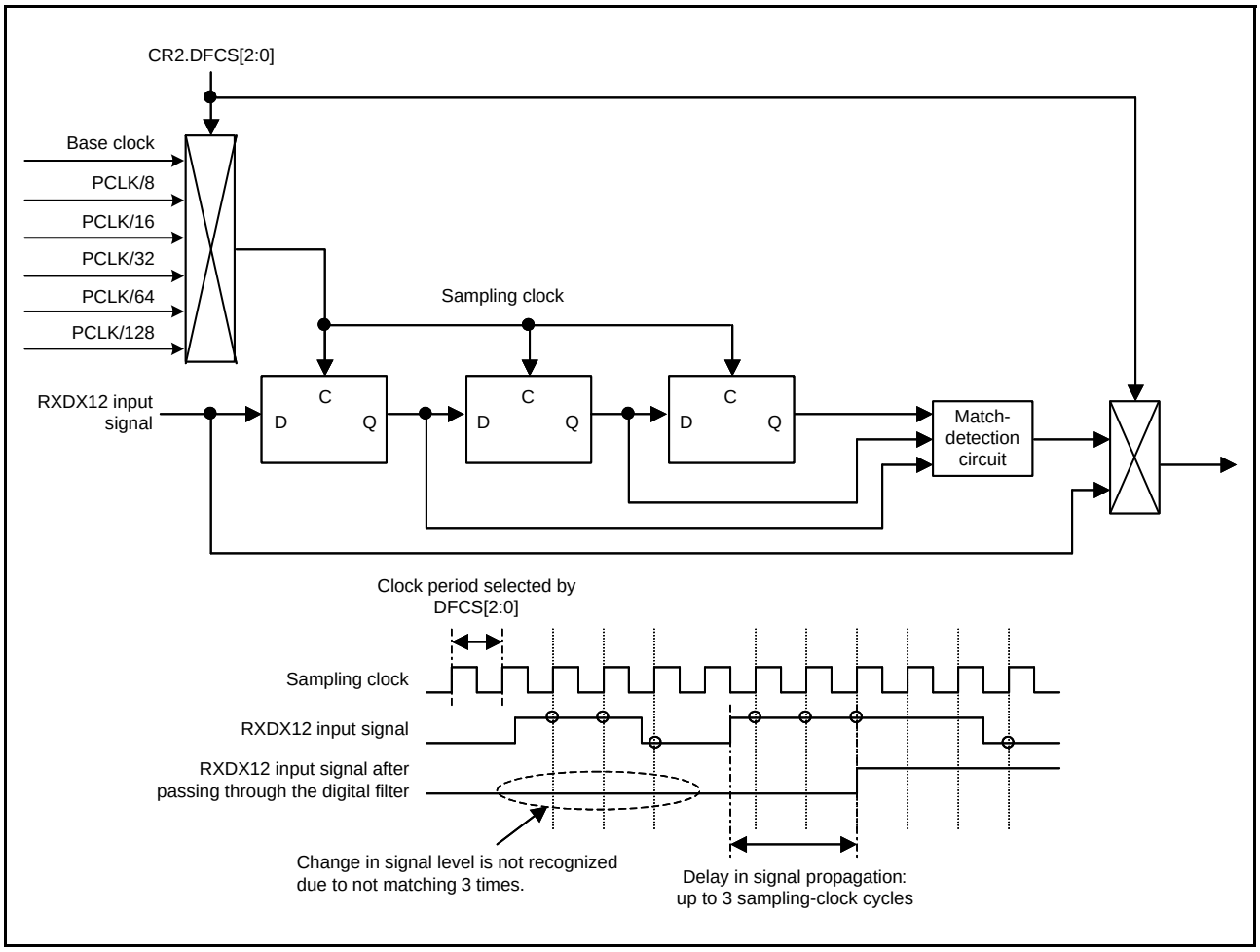
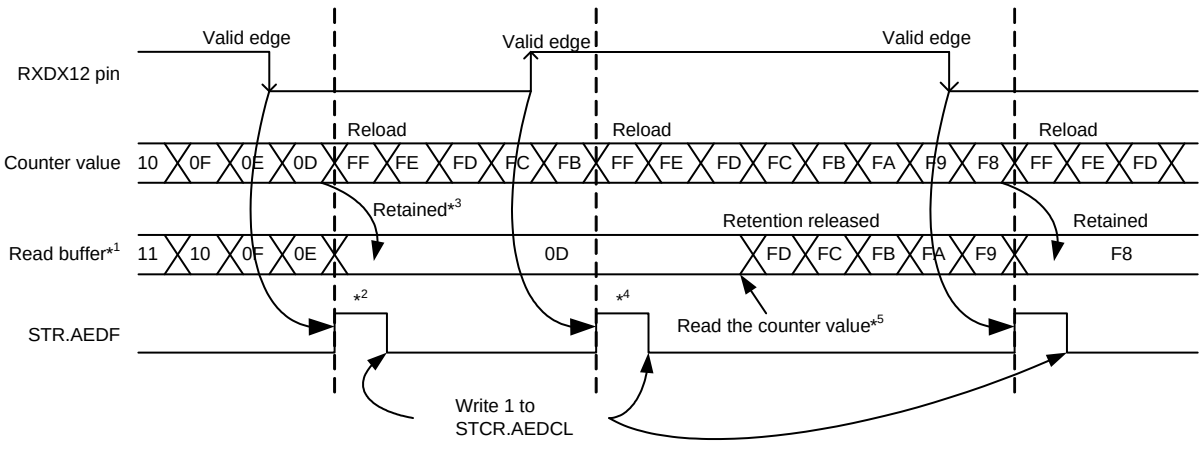


Figure 35.76 Example of Operations with the Digital Filter

35.10.6 Bit Rate Measurement

The bit rate measurement function measures the intervals between rising and falling edges and between falling and rising edges of the signal input from the RXDX12 pin. Figure 35.77 shows an example of operations for bit rate measurement.

(1) Writing 1 to the CR0.BRME bit enables bit rate measurement. Only set the BRME bit to 1 when you wish to proceed with bit rate measurement. Furthermore, bit rate measurement will not proceed during a Break Field, even if the BRME bit is set to 1.
(2) After detection of the Break Field low width, bit rate measurement starts when the level input on the RXDX12 pin becomes high.
(3) Once bit rate measurement has started, counter values from the timer are retained in the read buffers on the input of valid edges from the RXDX12 pin (rising and falling edges) and the counter is reloaded. An SCIX3 interrupt is also generated if the value of the ICR.AEDIE bit is 1. Retention by registers TCNT and TPRE is released by reading these registers.
(4) The bit rate as calculated from the values counted during intervals between valid edges can be used for adjusting the rate by changing the settings of the BRR register. To disable the bit rate measurement after a match with Control Field 1, write 0 to the CR0.BRME bit.



Note 1. The TCNT or TPRE register indicates the contents of the read buffers.
Note 2. The STR.AEDF flag becomes 1 on detection of a valid edge.
Note 3. When an effective edge is detected, the current value of the counter is stored in the read buffer.
Note 4. The value retained in the read buffers due to one valid edge is not updated to the new value even if a further valid edge is detected.
Note 5. Retention of a value in a read buffer due to a valid edge is released by reading the TCNT or TPRE register.

Figure 35.77 Example of Operations for Bit Rate Measurement

35.10.7 Selectable Timing for Sampling Data Received through RXDX12

The extended serial mode control section provides a way of adjusting the timing for the sampling of data received through the RXDX12 pin by setting the CR2.RTS[1:0] bits to select the rising edges of 8th, 10th, 12th, or 14th cycle of the base clock. If the value of the SEMR.ABCS bit is 1, the bits select the rising edges of 4th, 5th, 6th, or 7th cycle of the base clock. Figure 35.78 shows timing for the sampling of data received through RXDX12.

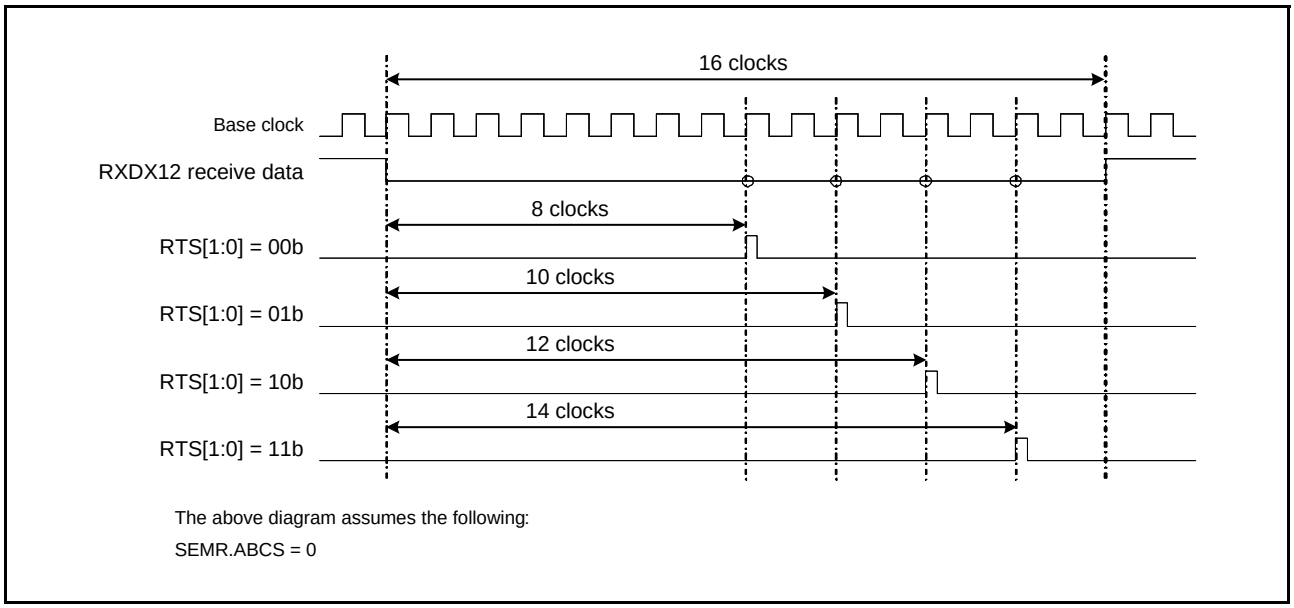


Figure 35.78 Timing for Sampling of Data Received through RXDX12

35.10.8 Timer

The timer has the following operating modes.

(1) Break Field Low Width Output Mode

This mode is for output through the TXDX12 pin of the low level over the Break Field low width at the transmission of a Start Frame. Setting the TMR.TOMS[1:0] bits to 10b switches operation to Break Field low width output mode. The TMR.TCSS[2:0] bits select the clock source for the counter. When the TCR.TCST bit is set to 1, the output on the TXDX12 pin goes to the low level and counting starts. When the timer underflows, the output on the TXDX12 pin goes to the high level and the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1. When 0 is written to the TCR.TCST bit, counting stops after reloading of registers TPRE and TCNT. After output of the Break Field low width is completed, stop the timer before it underflows again. Figure 35.79 shows an example of operations in Break Field low width output mode.

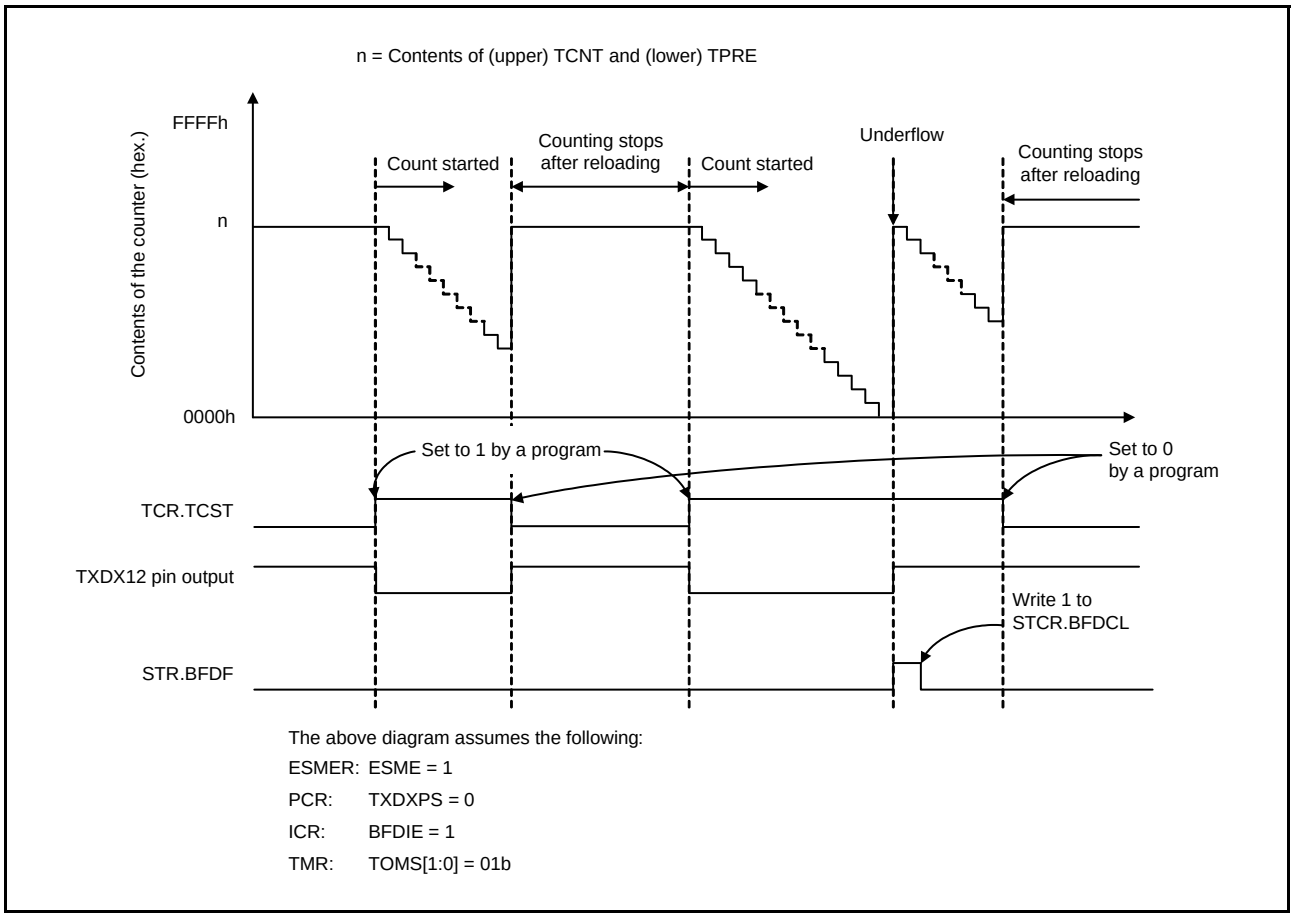


Figure 35.79 Example of Operations in Break Field Low Width Output Mode

(2) Break Field Low Width Determination Mode

This mode is for determining the Break Field low width in the input signal on the RXDX12 pin at the reception of a Start Frame. Setting the TMR.TOMS[1:0] bits to 01b switches operation to Break Field low width determination mode. The TMR.TCSS[2:0] bits select the clock source for the counter. When the TCR.TCST bit is set to 1, the interface enters the Break Field low width determinable state. Determination starts when a low level is input from the RXDX12 pin. When a high level is then input on the RXDX12 pin, registers TPRE and TCNT are reloaded and the interface enters the Break Field low width determinable state. When the timer underflows during Break Field low width determination, the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1. If an underflow of the timer during data transfer cause a problem in the form of interrupt generation, stop the timer after Break Field low width determination. Figure 35.80 shows an example of operations in Break Field low width output mode.

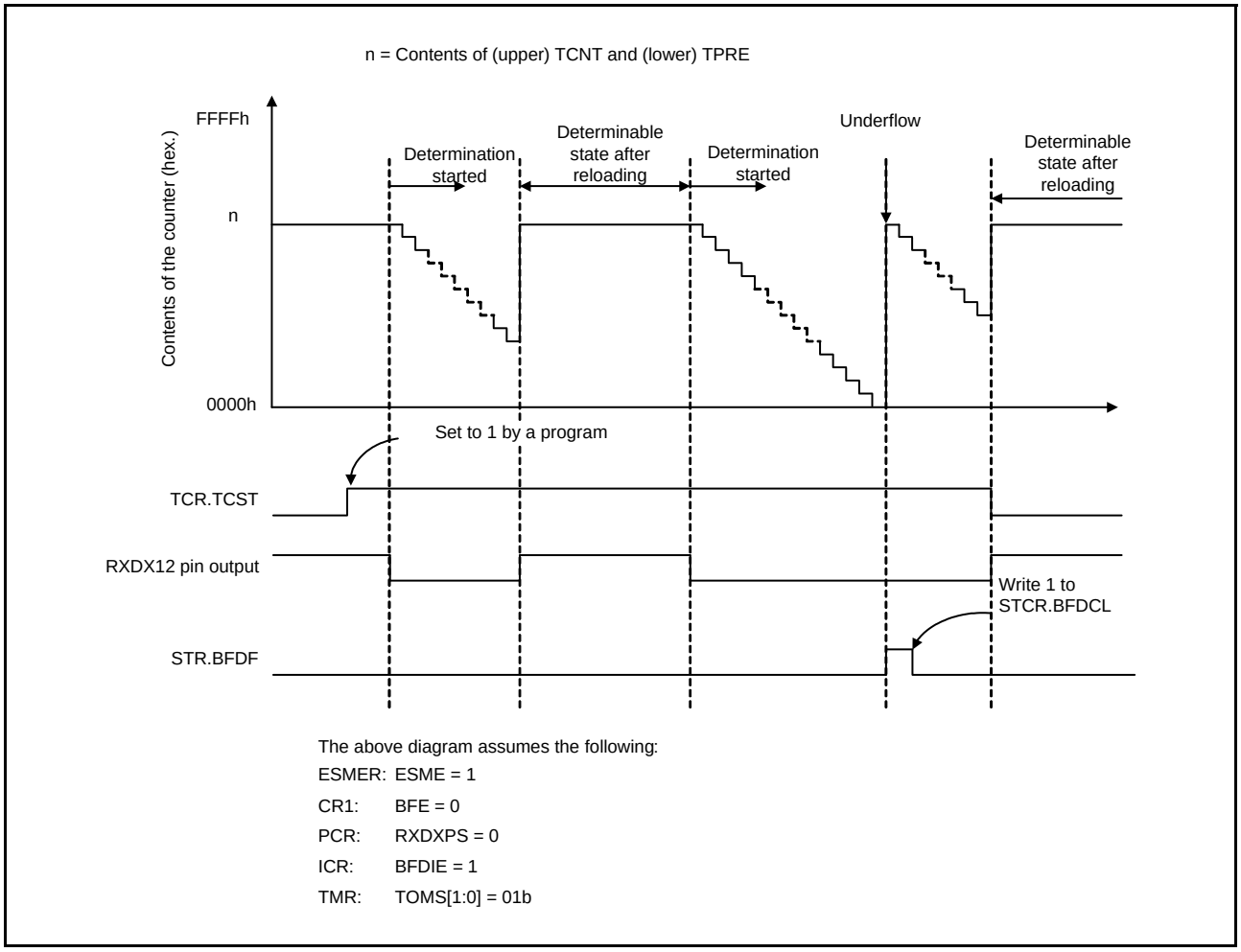


Figure 35.80 Example of Operations in Break Field Low Width Determination Mode

(3) Timer Mode

This mode is for counting cycles of the internal clock as the clock source. Setting the TMR.TOMS[1:0] bits to 00b switches operation to timer mode. The TMR.TCSS[2:0] bits select the clock source for the counter. Counting starts when 1 is written to the TCR.TCST bit and stops when 0 is written to the TCST bit. Registers TPRE and TCNT both count down. The TPRE register counts cycles of the clock source for counting, and underflows of the TPRE register provide the clock source for counting by the TCNT register. When the timer underflows, the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1.

35.11 Noise Cancellation Function

Figure 35.81 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.ABCSE = 0 and SEMR.ABCS = 0, 1/8th of a bit-period when SEMR.ABCSE = 0 and SEMR.ABCS = 1, and 1/6th of a bit-period when SEMR.ABCSE = 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.

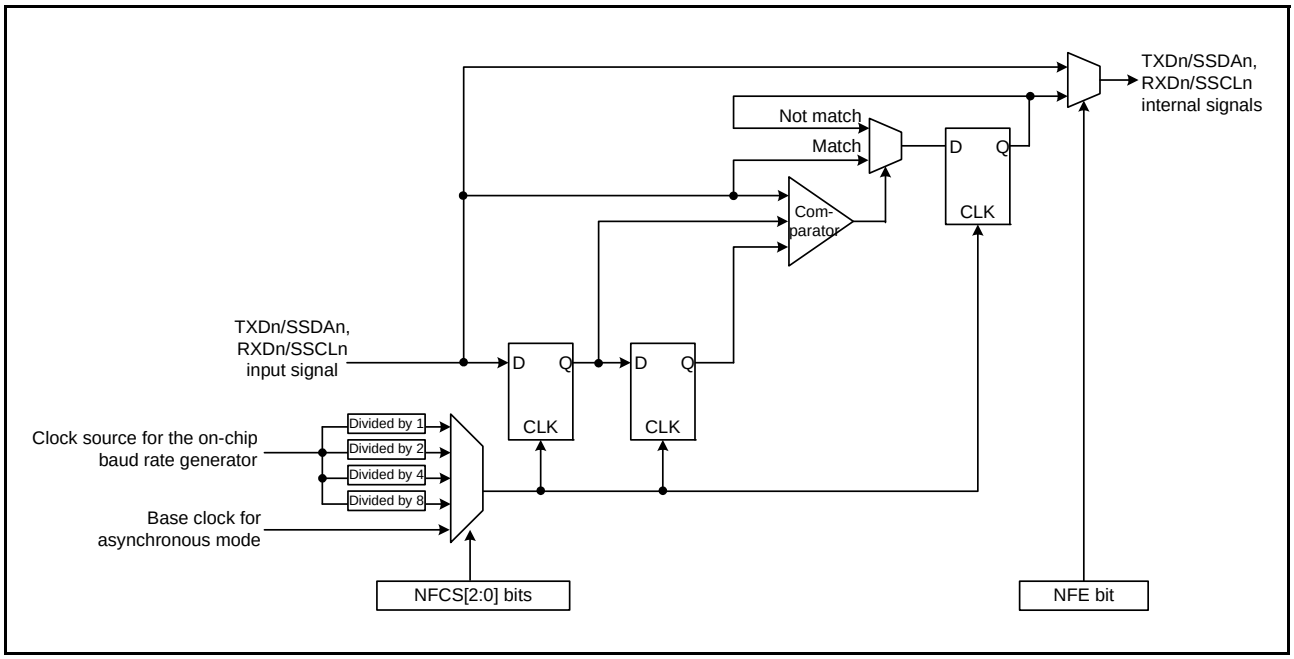


Figure 35.81 Block Diagram of Digital Noise Filter

35.12 Interrupt Sources

35.12.1 Buffer Operations for TXI and RXI Interrupts

(1) SCI0 to SCI9 and SCI12, and SCI10 and SCI11 when FIFO is Disabled

If the conditions for a TXI and RXI interrupt are satisfied while the interrupt status flag in the interrupt controller is 1, the SCI does not output the interrupt request but retains it internally (with a capacity for retention of one request per source). When the value of the interrupt status flag in the interrupt controller becomes 0, the interrupt request retained within the SCI is output. The internally retained interrupt request is automatically discarded once the actual interrupt is output. Clearing of the corresponding interrupt enable bit (the TIE or RIE bit in the SCR register) can also be used to discard an internally retained interrupt request.

(2) SCI10 and SCI11 When FIFO is Enabled

When the FCR.FM bit is 1, the SCI does not retain the interrupt request internally. When the interrupt status flag in the interrupt controller is 1 and the condition for interrupt generation is satisfied, the status flag is updated, but an interrupt request is not generated.

35.12.2 Interrupts in Asynchronous Mode, Clock Synchronous Mode, and Simple SPI Mode

(1) SCI0 to SCI9 and SCI12, and SCI10 and SCI11 when FIFO is Disabled

Table 35.38 lists interrupt sources in asynchronous mode, clock synchronous mode, and simple SPI mode. Individual interrupt sources can be enabled or disabled with the enable bits in the SCR register.

If the SCR.TIE bit is 1, a TXI interrupt request is generated when transmit data is transferred from the TDR or TDRL register*1 to the TSR register. A TXI interrupt request can also be generated by setting the SCR.TE bit to 1 after setting the SCR.TIE bit to 1 or by using a single instruction to set the SCR.TE and SCR.TIE bit to 1 at the same time. A TXI interrupt request can activate the DTC or DMAC to handle data transfer.

A TXI interrupt request is not generated by setting the SCR.TE bit to 1 while the setting of the SCR.TIE bit is 0 or by setting the SCR.TIE bit to 1 while the setting of the SCR.TE bit is 1.*2

Note that setting the SCR.TE bit to 0 while the SCR.TIE bit is 1 leads to the generation of a TXI interrupt.

When new data is not written by the time of transmission of the last bit of the current transmit data and the setting of the SCR.TEIE bit is 1, the SSR.TEND flag becomes 1 and a TEI interrupt request is generated. Furthermore, when the setting of the SCR.TE bit is 1, the SSR.TEND flag retains the value 1 until further transmit data are written to the TDR or TDRL register*1, and setting the SCR.TEIE bit to 1 leads to the generation of a TEI interrupt request.

Writing data to the TDR or TDRL register*1 leads to clearing of the SSR.TEND flag and, after a certain time, discarding of the TEI interrupt request.

If the SCR.RIE bit is 1, an RXI interrupt request is generated when received data is stored in the RDR register. 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 register to 1 while the SCR.RIE bit is 1 leads to the generation of an ERI interrupt request. An RXI interrupt request is not generated at this time. Clearing all three flags (ORER, FER, and PER) leads to discarding of the ERI interrupt request.

Note 1. In the case where asynchronous mode and 9-bit data length are selected

Note 2. To temporarily disable TXI interrupts at the time of transmission of the last of the data and so on when you wish a new round of transmission to start after handling of the transmission-completed interrupt, control disabling and enabling of the interrupt by using the interrupt request enable bit in the interrupt controller rather than using the SCR.TIE bit. This can prevent the suppression of TXI interrupt requests in the transfer of new data.

Table 35.38 Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation
ERI	Receive error	ORER, FER, PER, DFER*1, or DPER*1	Not possible	Not possible
RXI	Receive data full	RDRF	Possible	Possible
	Data match*1	DCMF*1		
TXI	Transmit data empty	TDRE	Possible	Possible
TEI	Transmit end	TEND	Not possible	Not possible

Note 1. Available in SCI0 to SCI11 only.

(2) SCI10 and SCI11 When FIFO is Enabled

Table 35.39 lists the interrupt sources in asynchronous mode and clock synchronization mode when the FIFO is enabled. Each interrupt source is enabled independently by using the enable bit in the SCR register.

When the SCR.TIE bit is 1, the number of data to be transmitted in the transmit FIFO is less than or equal to the threshold (FCR.TTRG[3:0] bits), a TXI interrupt request is generated. A TXI interrupt request is generated when the SCR.TE bit is set to 1 and then SCR.TIE bit is 1, or when the SCR.TIE and SCR.TE bit are simultaneously set to 1.

A TXI interrupt request is not generated when the SCR.TE bit is set to 1 while the SCR.TIE bit is 0 or the SCR.TE bit is set to 1 while the SCR.TIE bit is 1.

Note that setting the SCR.TIE bit to 1 while the SCR.TE bit is 0 leads to the generation of a TXI interrupt.

When the SCR.TEIE bit is 1 and if the next data is not yet written to the FTDR register by the timing when the last bit of the transmit data is transmitted, the SSRFIFO.TEND flag is 1 and a TEI interrupt request is generated.

When the SCR.RIE bit is 1, the number of the data stored in the receive FIFO is equal to or greater than the threshold (FCR.RTRG[3:0] bits), an RXI interrupt request is generated.

When the SCR.RIE bit is 1 and the SSRFIFO.ORER flag is 1 or the data with a framing or parity error is stored in the receive FIFO, an ERI interrupt request is generated. At this time, if the number of the data stored in the receive FIFO is equal to or greater than the threshold (FCR.RTRG[3:0] bits), an RXI interrupt request is generated. Clearing the ORER, FER, and PER flags in the SSRFIFO register cancels an ERI interrupt request.

Table 35.39 Interrupt Source

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation
ERI	Receive error	ORER, FER, PER, DFER, or DPER	Not possible	Not possible
		DR (when FCR.DRES = 1)		
RXI	Receive FIFO full	RDF	Possible	Possible
	Receive data ready	DR (when FCR.DRES = 0)		
	Data match	DCMF		
TXI	Transmit FIFO empty	TDRE	Possible	Possible
TEI	Transmit end	TEND	Not possible	Not possible

35.12.3 Interrupts in Smart Card Interface Mode

Table 35.40 lists interrupt sources in smart card interface mode. A transmit end interrupt (TEI) request cannot be used in this mode.

Table 35.40 SCI Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation
ERI	Receive error or error signal detection	ORER, PER, or ERS	Not possible	Not possible
RXI	Receive data full	—	Possible	Possible
TXI	Transmit data empty	TEND	Possible	Possible

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 SSR.TEND flag is set to 1, a TXI interrupt request is generated. This TXI interrupt request activates the DTC or DMAC allowing transfer of transmit data if the TXI request is specified beforehand as a source of DTC or DMAC activation. The TEND flag is automatically set to 0 when the DTC or DMAC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During the retransmission, the TEND flag is kept to 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, the SSR.ERS flag is not automatically cleared to 0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the SCR.RIE bit to 1 to enable an ERI interrupt request to be generated at error occurrence.

When transmitting/receiving data using the DTC or DMAC, be sure to make settings to enable the DTC or DMAC before making SCI settings. For DTC or DMAC settings, refer to section 20, Data Transfer Controller (DTCb) and section 18, DMA Controller (DMACAb).

In reception, an RXI interrupt request is generated when receive data is set to the RDR register. 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.

35.12.4 Interrupts in Simple I²C Mode

The interrupt sources in simple I²C mode are listed in Table 35.41. The STI interrupt is allocated to the transmit end interrupt (TEI) request. The receive error interrupt (ERI) request cannot be used.

The DTC or DMAC can also be used to handle transfer in simple I²C mode.

When the value of the SIMR2.IICINTM bit is 1, an RXI request will be generated on the falling edge of the SSCLn signal for the eighth bit. If the RXI has been set up as an activating request for the DTC or DMAC beforehand, the RXI request will activate the DTC or DMAC to handle transfer of the received data. Furthermore, a TXI request is generated on the falling edge of the SSCLn signal for the ninth bit (acknowledge bit). If the TXI has been set up as an activating request for the DTC or DMAC beforehand, the TXI request will activate the DTC or DMAC to handle transfer of the transmit data.

When the value of the SIMR2.IICINTM bit is 0, an RXI request (ACK detection) if the input on the SSDAn pin is at the low level or a TXI request (NACK detection) if the input on the SSDAn pin is at the high level will be generated on the rising edge of the SSCLn signal for the ninth bit (acknowledge bit). If the RXI has been set up as an activating request for the DTC or DMAC beforehand, the RXI request will activate the DTC or DMAC to handle transfer of the received data. Also, if the DTC or DMAC is used for data transfer in reception or transmission, be sure to set up and enable the DTC or DMAC before setting up the SCI.

When the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits in the SIMR3 register are used to generate a start condition, restart condition, or stop condition, the STI request is issued when generation is complete.

Table 35.41 SCI Interrupt Sources

Name	Interrupt Source		Interrupt Flag	DTC Activation	DMAC Activation
	IICINTM bit = 0	IICINTM bit = 1			
RXI	ACK detection	Reception	—	Possible	Possible
TXI	NACK detection	Transmission	—	Possible*1	Possible*1
STI	Completion of generation of a start, restart, or stop condition		IICSTIF	Not possible	Not possible

Note 1. Activation of the DTC or DMAC is only possible when the SIMR2.IICINTM bit is 1 (use reception and transmission interrupts).

35.12.5 Interrupt Requests from the Extended Serial Mode Control Section

The extended serial mode control section has a total of six types of interrupt request for generating the SCIX0 interrupt (Break Field low width detected), SCIX1 interrupt (Control Field 0 match, Control Field 1 match, priority interrupt bit detected), SCIX2 interrupt (bus collision detected), and SCIX3 interrupt (valid edge detected). When any of the interrupt factors is generated, the corresponding status flag is set to 1. Details of all of the interrupt requests are listed in Table 35.42.

Table 35.42 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 the CF0DR register
SCIX1 interrupt (Control Field 1 match)	CF1MF	The data received in Control Field 1 matching the value set in the PCF1DR or SCF1DR register
SCIX1 interrupt (priority interrupt bit detected)	PIBDF	The value of the bit specified as the priority interrupt bit matching the value set in the PCF1DR register
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

35.13 Event Linking

By employing interrupt request signals as event signals, SCI5 is able to provide linked operation through the event link controller (ELC) for modules selected in advance.
Event signals can be output regardless of the values of the corresponding interrupt request enable bits.

(1) Error (receive error, error signal detected) event output

- Indicates abnormal termination due to a parity error during reception in asynchronous mode.
- Indicates abnormal termination due to a framing error during reception in asynchronous mode.
- Indicates abnormal termination due to an overrun error during reception.
- Indicates detection of the error signal during transmission in smart card interface mode.

(2) Receive data full event output

- Indicates that received data have been set in the receive data register (RDR or RDRL).
- Indicates that ACK has been detected if the SIMR2.IICINTM bit is 0 in simple I^2C mode.
- Indicates that the 8th-bit SSCL5 falling edge has been detected if the SIMR2.IICINTM bit is 1 in simple I^2C mode.
- When the SIMR2.IICINTM bit is 1 during master transmission in simple I^2C mode, set the event link controller (ELC) so that receive data full events are not used.

(3) Transmit data empty event output

- Indicates that the SCR.TE bit has been changed from 0 to 1.
- Indicates that transmit data have been transferred from the transmit data register (TDR or TDRL) to the transmit shift register (TSR).
- Indicates that transmission has been completed in smart card interface mode.
- Indicates that NACK has been detected if the SIMR2.IICINTM bit is 0 in simple I^2C mode.
- Indicates that the ninth-bit SSCL5 falling edge has been detected if the SIMR2.IICINTM bit is 1 in simple I^2C mode.

(4) Transmit end event output

- Indicates the completion of transmission.
- Indicates that the starting condition, resumption condition, or termination condition has been generated in simple I^2C mode.

35.14 Usage Notes

35.14.1 Setting the Module Stop Function

Module stop control register B (MSTPCRB) and module stop control register C (MSTPCRC) are used to stop and start SCI operations. With the value after a reset, SCI operations are stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

35.14.2 Break Detection and Processing

(1) SCI0 to SCI9 and SCI12, and SCI10 and SCI11 when FIFO is Disabled

When a framing error is detected, a break can be detected by reading the RXDn pin value directly or reading the value of the SPTR.RXDMON flag (only for SCI0 to SCI11). In a break, the input from the RXDn pin becomes all 0s, and so the SSR.FER flag is set to 1 (framing error has occurred), and the SSR.PER flag may also be set to 1 (parity error has occurred). When the SEMR.RXDESEL bit is 0, the SCI continues the receive operation even after a break is received. Therefore, note that even if the FER flag is set to 0 (no framing error occurred), it will be set to 1 again. When the SEMR.RXDESEL bit is 1, the SCI sets the SSR.FER flag to 1 and stops receiving operation until a start bit of the next data frame is detected. If the SSR.FER flag is set to 0 at this time, the SSR.FER flag retains 0 during the break. When the RXDn pin becomes high and the break ends, detecting the beginning of the start bit at the first falling edge of the RXDn pin allows the SCI to start the receiving operation.

(2) SCI10 and SCI11 When FIFO is Enabled

When a framing error is detected and the following one-frame data received is all 0s, the SCI stops receiving operation. Upon the detection of a framing error, reading the value of the SPTR.RXDMON flag detects a break. After the RXDn pin becomes high and the break ends, the SCI resumes receiving operation.

35.14.3 Mark State and Sending Breaks

When the SCR.TE bit is 0 (serial transmission is disabled), the TXDn pin becomes high-impedance. To forcibly set the TXDn pin to mark or space state while the TE bit is 0, set the I/O port associated registers and switch the TXDn pin to general output port.

For holding the communication line in the mark (“1”) state until the TE bit is set to 1 (serial transmission is enabled), set the corresponding bit in the PODR register to 1 for high output from general output port. To start communications, set the TE bit to 1 and then the corresponding bit in the PMR register to 1.

To send a break (the space state for longer than a certain period of time) while data transmission, set the corresponding bit in the PODR register to 0 (low output), and set the corresponding bit in the PMR register to 0 (general I/O port). Then set the TE bit to 0 if necessary. When the TE bit is set to 0, the transmitter is initialized regardless of the current transmit status.

The SPTR register, if it is included, can set the TXDn pin in mark/space state without switching the pin function to general output port. Set the SPTR.SPB2IO bit to 1 (output) and the SPB2DT bit to 1 (mark) or 0 (space), and then set the TE bit to 0.

35.14.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode)

Transmission cannot be started when a receive error flag (SSR.ORER) is set to 1, even if data is written to the TDR register (FTDR register). Be sure to set the receive error flags to 0 before starting transmission. Note also that the receive error flags cannot be set to 0 even if the SCR.RE bit is set to 0 (serial reception is disabled).

35.14.5 Writing Data to the TDR Register

Data can be written to registers TDR, TDRH, and TDRL. However, if new data is written to registers TDR, TDRH, and TDRL when transmit data is remaining in registers TDR, TDRH, and TDRL, the previous data in registers TDR, TDRH, and TDRL is lost because it has not been transferred to the TSR register yet. Be sure to write transmit data to registers TDR, TDRH, and TDRL in the TXI interrupt request handling routine.

35.14.6 Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode and Simple SPI Mode)

When the external clock source is used as a synchronization clock, the following restrictions apply.

(1) Start of transmission

Update the TDR register by the CPU, DMAC, or DTC and wait for at least five PCLK cycles before allowing the transmit clock to be input (refer to Figure 35.82).

(2) Continuous transmission

(a) Write the next transmit data to the TDR or TDRL register before the falling edge of the transmit clock (bit 7) (refer to Figure 35.82).

(b) When updating the TDR register after bit 7 has started to transmit, update the TDR register while the synchronization clock is in the low-level period, and set the high-level width of the transmit clock (bit 7) to four PCLK cycles or longer (refer to Figure 35.82).

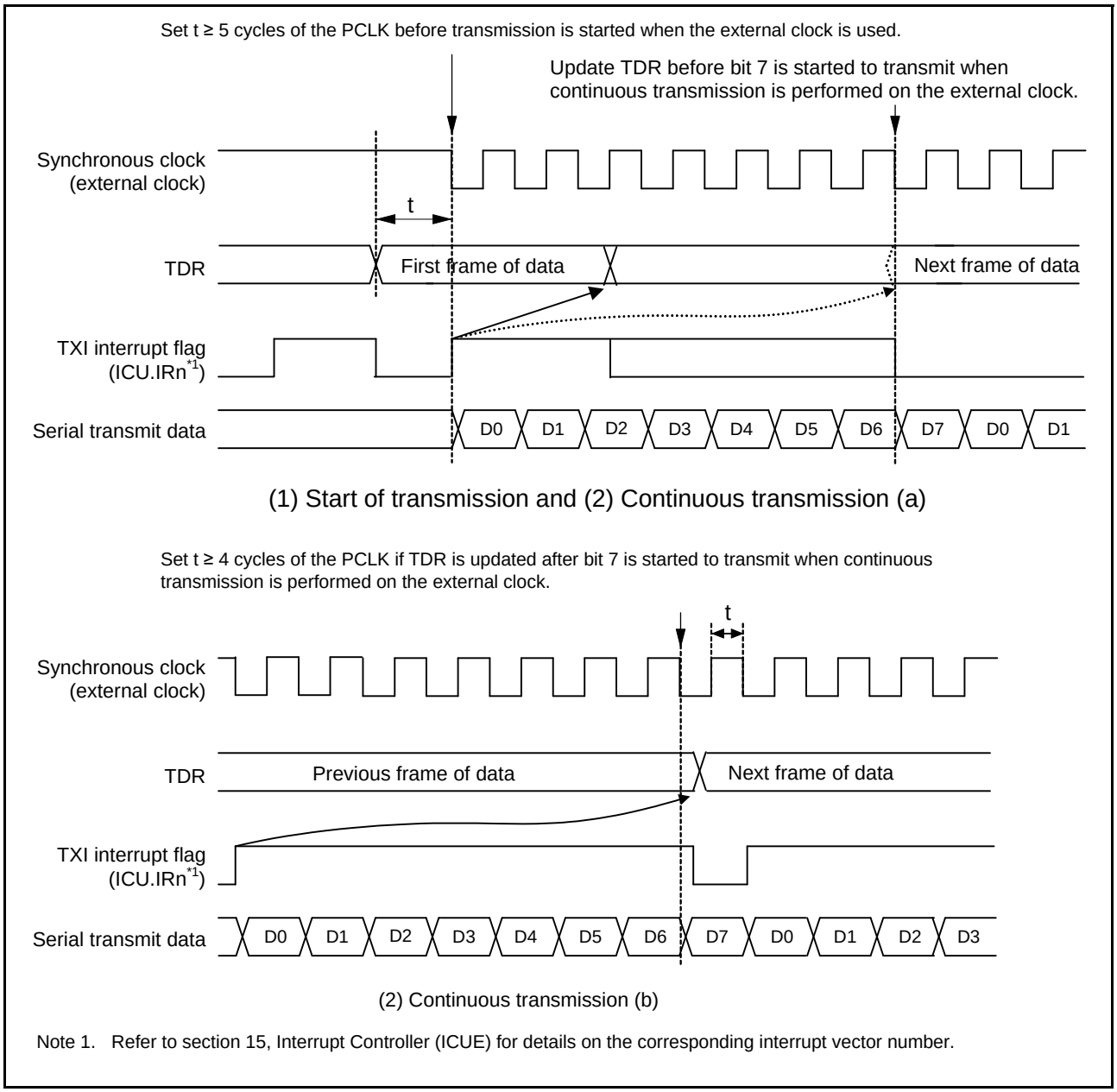


Figure 35.82 Restrictions on Use of External Clock in Clock Synchronous Transmission

35.14.7 Restrictions on Using DMAC or DTC

When using the DMAC or DTC to read the RDR, RDRH, and RDRL registers, be sure to set the receive data full interrupt (RXI) as the activation source of the relevant SCI.

35.14.8 Notes on Starting Transfer

At the point where transfer starts when the interrupt status flag (IRn.IR 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). For details on the interrupt status flag, refer to section 15, Interrupt Controller (ICUE).

- Confirm that transfer has stopped (the setting of the SCR.TE or SCR.RE bits is 0).
- Set the corresponding interrupt enable bit (SCR.TIE or SCR.RIE) to 0.
- Read the corresponding interrupt enable bit (SCR.TIE or SCR.RIE bit) to check that it has become 0.
- Set the interrupt status flag (IRn.IR flag) in the interrupt controller to 0.

35.14.9 SCI Operations during Low Power Consumption State

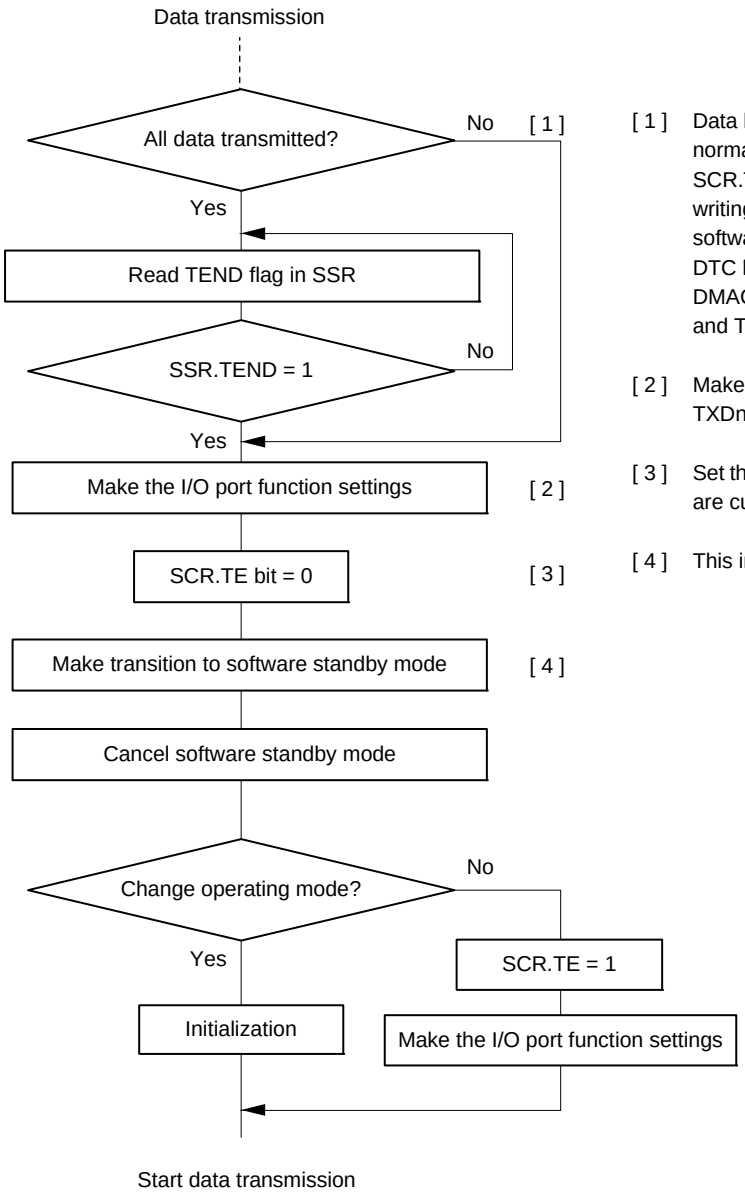
(1) Transmission

When making settings for the module stopped state or in transitions to software standby, stop operations (by setting the TIE, TE, and TEIE bits in the SCR register to 0) after switching the TXDn pin to the general I/O port pin function or fix the output level of the TXDn pin by the SPTR register (only for SCI0 to SCI11). Setting the TE bit to 0 resets the TSR register and the SSR.TEND flag. The SSRFIFO.TEND flag in the SCI10 and SCI11 is not initialized. Depending on the port settings or SPTR register setting (only for SCI0 to SCI11), output pins may output the level before a transition to the low power consumption state is made after release from the module stopped state or software standby mode. When transitions to these states are made during transmission, the data being transmitted become indeterminate.
To transmit data in the same transmit mode after cancellation of the low power consumption state, set the TE bit to 1, read the SSR register, and write data to the TDR register sequentially to start data transmission. To transmit data with a different transmit mode, initialize the SCI first.
Figure 35.83 shows a sample flowchart for transition to software standby mode during transmission. Figure 35.84 and Figure 35.85 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 transmit mode using DTC/DMA transfer, stop the transmit operations (TE = 0). To start transmission after cancellation using the DTC/DMAC, set the TE and TIE bits to 1. The TXI interrupt flag is set to 1 and transmission starts using the DTC/DMAC.

(2) Reception

(a) When the data match function is not used for release from the low-power consumption state

Before specifying the module stop state or making a transition to software standby mode, stop the receive operations (SCR.RE = 0). If transition is made during data reception, the data being received will be invalid.
To receive data in the same receive 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 receive mode, initialize the SCI first.
Figure 35.86 shows a sample flowchart for transition to software standby mode during reception.



[1] Data being transmitted is lost halfway. Data can be normally transmitted from the CPU by setting the SCR.TE bit to 1, reading the SSR register, and writing data to the TDR register after canceling software standby mode. However, if the DMAC or DTC has been activated, the data remaining in the DMAC or DTC will be transmitted when both the TE and TIE bits in the SCR register are set to 1.

[2] Make the I/O port function settings to switch the TXDn pin to operate as a general I/O port.

[3] Set the TIE and TEIE bits in the SCR register if they are currently set to 1.

[4] This includes the setting for the module stop state.

Figure 35.83 Example of Flowchart for Transition to Software Standby Mode during Transmission

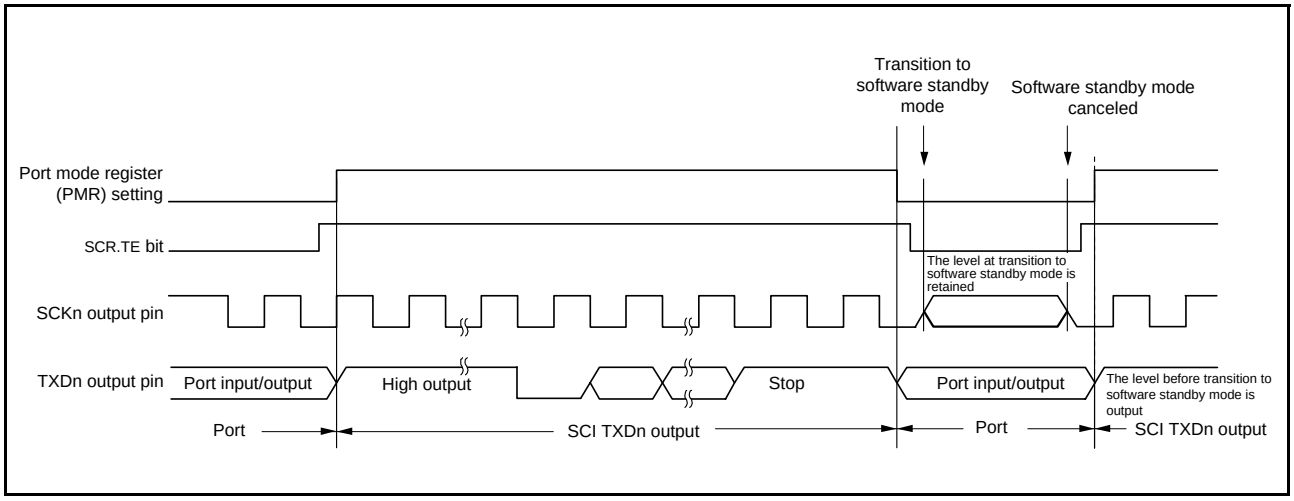


Figure 35.84 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission)

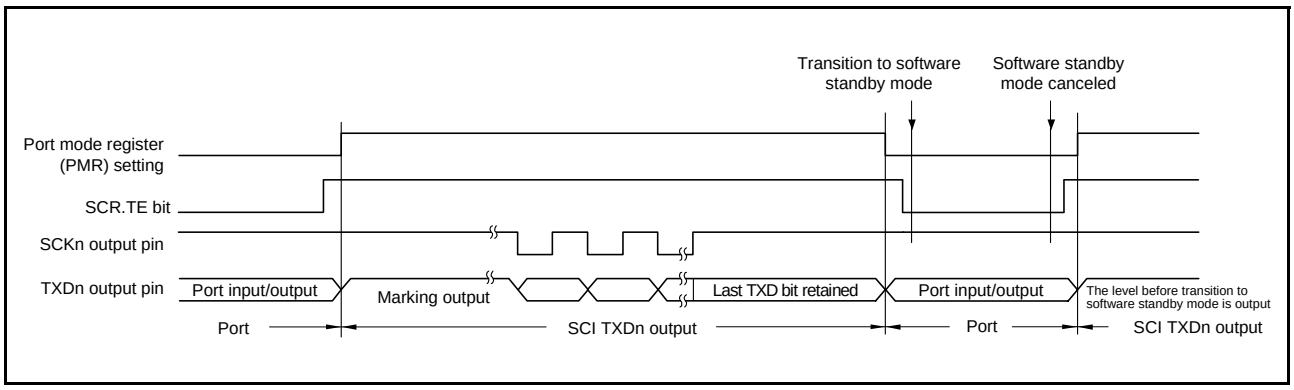


Figure 35.85 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission)

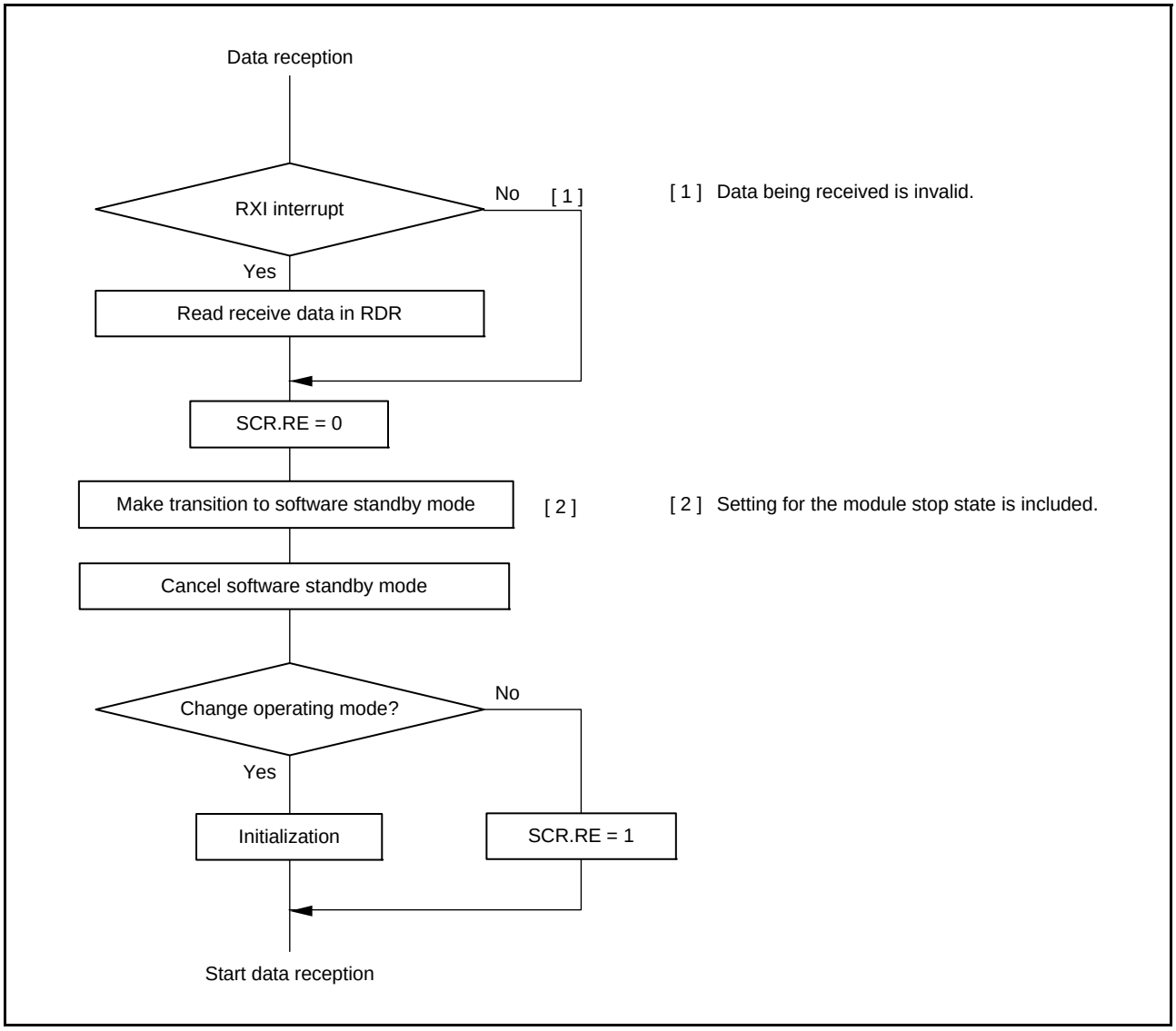


Figure 35.86 Example of Flowchart for Transition to Software Standby Mode during Reception

(b) When the data match function is used for release from the low-power consumption state

Set an operating mode to be set after release from the low-power consumption state before reducing power consumed by the SCI by using the function of reducing power consumption. Afterwards, set the comparison data in the CDR.CMPD[8:0] bits and set the DCCR.DCME bit to 1. While the SCR.RE bit is 1, enter to the low-power consumption state.

If transition to the low-power consumption state may take place while the RXDn pin is driven low, set the SEMR.RXDESEL bit to 0. When the SEMR.RXDESEL bit is 1, the start bit may not be detected at the time of release from the low-power consumption state.

35.14.10 External Clock Input in Clock Synchronous Mode and Simple SPI Mode

In clock synchronous mode and simple SPI mode, the external clock SCKn must be input as follows:
High-pulse period, low-pulse period = 2 PCLK cycles or more, period = 6 PCLK cycles or more

35.14.11 Limitations on Simple SPI Mode

(1) Master Mode

- Use a resistor to pull up or pull down the clock line matching the initial settings for the transfer clock set by the SPMR.CKPH and CKPOL bits when the SPMR.SSE bit is 1.
  This prevents the clock line from being placed in the high-impedance state when the SCR.TE bit is set to 0 or unexpected edges from being generated on the clock line when the SCR.TE bit is changed from 0 to 1. When the SPMR.SSE bit is 0 in single master mode, pulling up or pulling down the clock line is not necessary because the clock line is not placed in the high-impedance state even when the SCR.TE bit is set to 0.
- In the case of the setting for clock delay (SPMR.CKPH bit is 1), the receive data full interrupt (RXI) is generated before the final clock edge on the SCKn pin as indicated in Figure 35.87. If the TE and RE bits in the SCR register become 0 at this time before the final edge of the clock signal on the SCKn pin, the SCKn pin is placed in the high-impedance state, so the width of the last clock pulse of the transfer clock is shortened. Furthermore, an RXI interrupt may lead to the input signal on the SSn# pin of a connected slave going to the high level before the final edge of the clock signal on the SCKn pin, leading to incorrect operation of the slave.
- In a multi-master configuration, take care because the SCKn pin output becomes high-impedance while the input on the SSn# pin is at the low level if a mode fault error occurs as the current character is being transferred, stopping supply of the clock signal to the connected slave. Remake the settings for the connected slave to avoid misaligned bits when transfer is restarted.

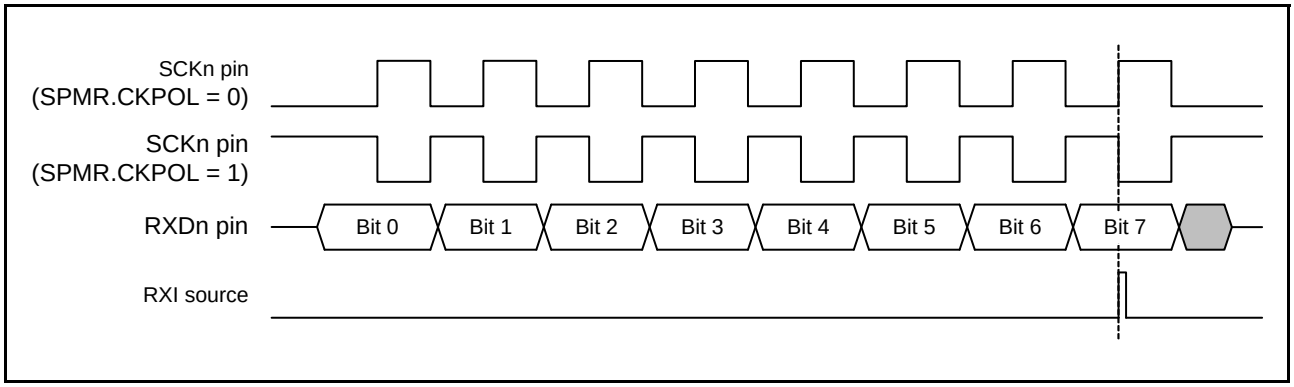


Figure 35.87 Timing of the RXI Interrupt in Simple SPI Mode (with Clock Delay)

(2) Slave Mode

- Secure at least five cycles of the PCLK from writing transmit data in the TDR register to start of the external clock input. Also secure at least five cycles of the PCLK from input of low level on the SSn# pin to start of the external clock input.
- Provide an external clock signal from the master the same as the transmit/receive data length.
- Control the input on the SSn# pin before the start and after the end of data transfer.
- When the level being input on the SSn# pin is to be changed from low to high while the current character is being transferred, set the TE and RE bits in the SCR register to 0 and, after remaking the settings, restart transfer of the first byte.

35.14.12 Limitation 1 on Usage of the Extended Serial Mode Control Section

When the PCR.SHARPS bit is set to 1, output on the TXDX12/RXDX12 pin is only possible when the following conditions apply.

- The timer is in Break Field low width output mode and the value of the TCR.TCST bit is 1 (when the TCST bit is set to 1, the high level continues to be output for up to one cycle of the clock source for counting by the timer counter before output of the low level)
- The value of the SCR.TE bit is 1.

35.14.13 Limitation 2 on Usage of the Extended Serial Mode Control Section

The TXI, RXI, ERI, and TEI interrupt requests are generated even if the extended serial mode is enabled. However, the RXI interrupt should not be enabled during reception of a Start Frame because the extended serial mode control section uses the receive data full signal.
To use the RXI interrupts during a reception of the Information Frame, use it in accordance with one of the following procedures. When a receive error is detected, clear the receive error flag and initialize the extended serial mode control section.

(1) Set the SCR.RIE bit to 0 to disable the output of interrupt requests. Check the error flags in the SSR register on completion of the reception of a Start Frame, because an ERI interrupt is not generated if a receive error occurs. After reception of the Start Frame is completed, set the SCR.RIE bit to 1 by the time the first byte of the Information Frame is received.
(2) Set the SCR.RIE bit 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.

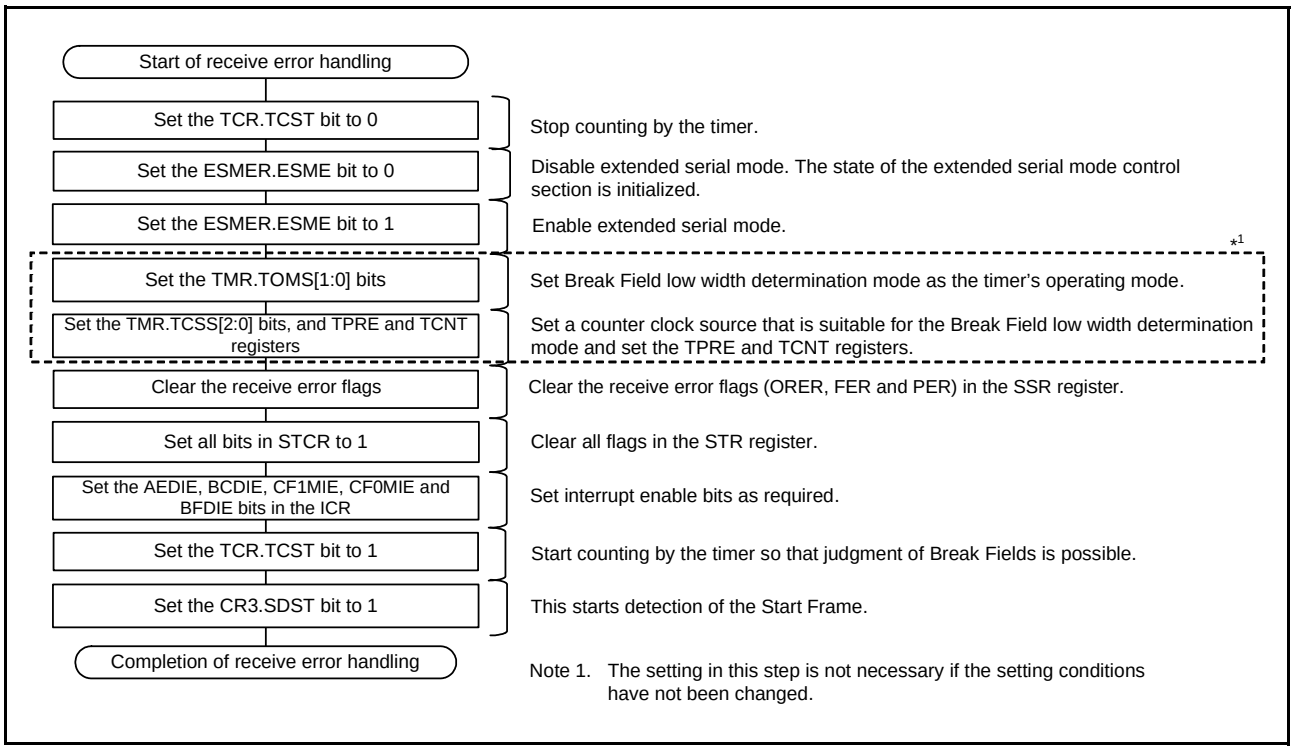


Figure 35.88 Example of Flowchart for Receive Error Handling (during Reception of the Start Frame)

35.14.14 Note on Transmit Enable Bit (TE Bit)

When setting the pin function to “TXDn” while the SCR.TE bit is 0 (serial transmission is disabled) or setting the TE bit to 0 while the pin function is “TXDn”, output of the TXDn pin becomes high-impedance.

Prevent the TXDn line from becoming high-impedance by any of the following ways:

(1) Connect a pull-up or pull-down resistor to the TXDn line.

(2) Set the TE bit to 1*1 before changing the pin function to “TXDn”. Before setting the TE bit to 0, change the pin function to “general-purpose I/O port” and drive the pin high or low.

(3) Set the SPTR.SPB2IO bit to 1 first, and change the pin function to “TXDn”. Leave the value of the SPB2IO bit as 1 after that (for SCI0 to SCI11).

Note 1. An interrupt is generated when the TE bit is set to 1 while the TXI interrupt is enabled (SCI0 to SCI9 and SCI12, and SCI10 and SCI11 when FIFO is Disabled). If this creates a problem, change the pin function to “TXDn” first, and then set the corresponding ICU.IERm.IENj bit to 1.

35.14.15 Note on Stopping Reception When Using the RTS Function in Asynchronous Mode

One clock cycle of PCLK is required for the time from setting the SCR.RE bit to 0 to stopping the RTS signal generator in asynchronous mode.

When reading the RDR (or RDRL) register after setting the SCR.RE bit to 0, confirm that the RE bit has been set to 0 before reading the RDR (or RDRL) register to prevent these two processes from being performed consecutively.

36. Serial Communications Interface (RSCI)

36.1 Overview

RSCI can handle both asynchronous and clock synchronous serial communications. RSCI has 32-stage FIFO buffers in transmission/reception blocks, and it can select the FIFO composition, and it can transmit/receive efficiently, and it can also communicate continuously.

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 RSCI also supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for identification cards).

The RSCI is also supports serial communication using manchester code (manchester mode), simple SPI interfaces, simple I^2C-bus interfaces (single master), and extended serial communication.

In addition, asynchronous mode has a support function for generating AMI waveform of negative logic coding with 50% duty cycle used in home bus system (HBS) communications.

Table 36.1 lists the RSCI specifications.

Table 36.1 RSCI Specifications (1/3)

Item	Description
Serial communication modes	• Asynchronous • Manchester • Clock synchronous • Smart card interface • Simple I^2C-bus • Simple SPI bus • Extended serial
Transfer speed	Bit rate specifiable with the on-chip baud rate generator.
Full-duplex communications	Transmitter: Continuous transmission possible using double-buffer structure. Receiver: Continuous reception possible using double-buffer structure.
Half-duplex communications	Half-duplex communication is possible by using only TXDn pins
Data transfer	Selectable as LSB first or MSB first transfer
I/O signal level inverting function	Input signal and output signal can be inverted independently.
Interrupt sources	Transmit end, transmit data empty, receive data full, receive error, receive data ready, and receive data match Break Field detection/transmission, Bus collision detection, Active edge detection Completion of generation of a start condition, restart condition, or stop condition
RS-485 driver control function	Output DE signal to enable external transceiver transmit mode
Loopback function	Self-diagnosis of communication function is possible by connecting TXD and RXD inside the RSCI
Low power consumption function	Module stop state can be set for each channel.

Table 36.1 RSCI Specifications (2/3)

Item		Description
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
	Transmission/Reception	Selectable either 1 stage register or 32 stage FIFO
	Data match detection	The interrupt request can issue by detecting the match between receive data and comparison data.
	Start-bit detection	Low level or falling edge is selectable.
	Receive data sampling timing adjustment	Sampling point of receive data can be changed to front or rear of center of bit.
	Transmit signal transition timing adjustment	Falling or rising edge of the transmit data can be delayed.
	Break detection	When a framing error occurs, a break can be detected by reading the SSR register.
	Clock source	An internal or external clock can be selected.
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Multi-processor communications function	Serial communication among multiple processors
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters.
	HBS support mode	Transmission/reception using inverted RZI (Return to Zero, Inverted) code is possible.
Manchester mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Receive error detection	Parity, overrun, framing, manchester code errors, preface, start bit, and receive Sync
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission.
	Clock source	Only internal clock can be used. (The setting of external clock is prohibited because it is not the object of operation guarantee.)
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Multi-processor communications function	Serial communication among multiple processors
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters.
	Manchester encoding/ decoding function	Function to perform manchester encoding/decoding of transmission/reception data and communicate using manchester code
	Preface setting/detection function	Function to detect the beginning of a frame from the preface pattern. Preface pattern can be selected from 4 types. The length can also be changed from 0 to 15 bits.
	Start Bit setting/detection function	The Start Bit length can be set to 1 bit or 3 bits. In the case of 3-bit length, it is possible to judge the type of subsequent data with two types of patterns.
	Reception retiming function	Function to perform timing correction for each bit center edge by using manchester code having edge at bit center
Smart card interface mode	Error processing	An error signal can be automatically transmitted when detecting a parity error during reception
		Data can be automatically retransmitted when receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are supported.

Table 36.1 RSCI Specifications (3/3)

Item		Description
Extended serial mode	Start Frame Transmission	Break Field transmission possible, Break Field transmission complete interrupt output possible Bus collision detection possible, bus collision detection interrupt output possible
	Start Frame Reception	Break Field detectable, Break Field detected interrupt output possible Control Field 0/1 data comparison function Control Field 1 can set two types of comparison data of primary and secondary Priority interrupt bit can be set in Control Field 1 Bit rate measurement function
Simple I2C mode	Transfer format	I2C-bus format
	Operating mode	Master (single-master operation only)
	Transfer rate	Up to 400 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.
Clock synchronous mode	Data length	8 bits
	Adjustment of receive sampling timing	Adjustable receive sampling timing after the default timing in master mode only when using internal clock
	Receive error detection	Overrun error
	Clock source	An internal clock (master mode) or external clock (slave mode) can be selected.
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
	Transmission/Reception	Selectable either 1 stage register or 32 stage FIFO
Simple SPI bus	Data length	8 bits
	Detection of errors	Overrun error
	Clock source	An internal clock (master mode) or external clock (slave mode) can be selected.
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Adjustment of receive sampling timing	Adjustable receive sampling timing after the default timing in master mode only when using internal clock
	SS input pin function	When the SSn # pin is high level, the output pin can be set to high impedance
	Adjustment of receive sampling timing	Four kinds of settings for clock phase and clock polarity are selectable.
	Transmission/Reception	Selectable either 1 stage register or 32 stage FIFO
Bit rate modulation function		Correction of outputs from the on-chip baud rate generator can reduce errors.
Event link function		Error (receive error or error signal detection) event output
		Receive data full event output
		Transmit data empty event output
		Transmit end event output
		Receive data match event output
		Receive data unmatch event output
		Active edge detection event output

Figure 36.1 shows RSCI Block Diagram (n = 010, 011) with all functions.

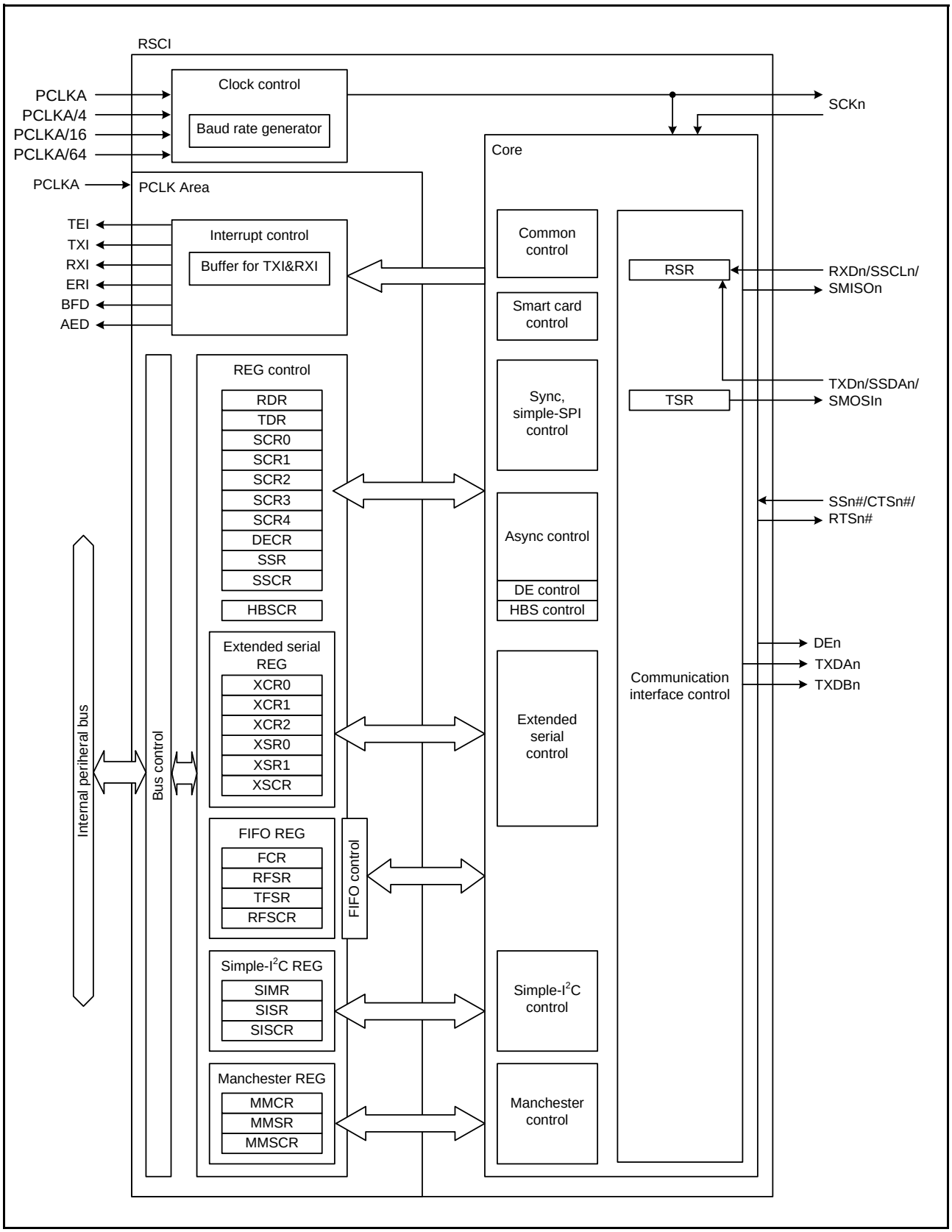


Figure 36.1 RSCI Block Diagram (n = 010, 011)

Table 36.2 to Table 36.5 list RSCI's input/output pins.

Table 36.2 RSCI Input/Output Pin (Asynchronous Mode/Clock Synchronous Mode/Manchester Mode/ Extended Serial Mode)

Channel	Pin Name	I/O	Function
RSCI10	SCK010	I/O	RSCI10 clock input/output
	RXD010	Input	RSCI10 receive data input
	TXD010	Output	RSCI10 transmit data output
	RTS010#	Output	RSCI10 request-to-send signal output
	CTS010#	Input	RSCI10 transmission start control input
	DE010	Output	RSCI10 RS-485 driver control output
RSCI11	SCK011	I/O	RSCI11 clock input/output
	RXD011	Input	RSCI11 receive data input
	TXD011	Output	RSCI11 transmit data output
	RTS011#	Output	RSCI11 request-to-send signal output
	CTS011#	Input	RSCI11 transmission start control input
	DE011	Output	RSCI11 RS-485 driver control output

Table 36.3 RSCI Input/Output Pin (Simple I^2C Mode)

Channel	Pin Name	I/O	Function
RSCI10	SSCL010	I/O	RSCI10 I^2C clock input/output
	SSDA010	I/O	RSCI10 I^2C data input/output
RSCI11	SSCL011	I/O	RSCI11 I^2C clock input/output
	SSDA011	I/O	RSCI11 I^2C data input/output

Table 36.4 RSCI Input/Output Pin (Simple SPI Mode)

Channel	Pin Name	I/O	Function
RSCI10	SCK010	I/O	RSCI10 clock input/output
	SMISO010	I/O	RSCI10 slave transmit data input/output
	SMOSI010	I/O	RSCI10 master transmit data input/output
	SS010#	Input	RSCI10 slave select input
RSCI11	SCK011	I/O	RSCI11 clock input/output
	SMISO011	I/O	RSCI11 slave transmit data input/output
	SMOSI011	I/O	RSCI11 master transmit data input/output
	SS011#	Input	RSCI11 slave select input

Table 36.5 RSCI Input/Output Pin (HBS Support Mode)

Channel	Pin Name	I/O	Function
RSCI10	RXD010	Input	RSCI10 receive data input
	TXD010	Output	RSCI10 transmit data output
RSCI11	RXD011	Input	RSCI11 receive data input
	TXD011	Output	RSCI11 transmit data output
	TXDA011/TXDB011	Output	RSCI11 transmit data output (in alternate output)

36.2 Register Descriptions

This chapter describes the RSCI registers, their functional specifications, and their operating specifications.

36.2.1 Receive Shift Register (RSR)

RSR is a shift register which is used to receive serial data input from the RXDn pin and converts it into parallel data. When one frame of data has been received, it is automatically transferred to the RDR register.
The RSR register cannot be directly accessed by the CPU.

36.2.2 Receive Data Register (RDR)

Address(es): RSCI10.RDR 000E 2000h, RSCI11.RDR 000E 2080h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	AFER	APER	—	—	ORER	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	FER	PER	DR	MPB	RDAT[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	RDAT[8:0]	Receive Data	RDAT[8:0] bits are a 9-bit field for storing received data. Received data is stored in [6:0] when 7-bit data is selected, in [7:0] when 8-bit data is selected, and in [8:0] when 9-bit data is selected. And 0 is stored in the unused bit.	R
b9	MPB	Multi-Processor Bit Monitor Flag	0: Data transmission cycles 1: ID transmission cycles	R
b10	DR	Receive Data Ready Flag	RFSR.DR flag can be read.	R
b11	PER	Parity Error Flag	(Valid only in asynchronous mode) 0: There is no parity error in the data read from the receive FIFO 1: There is parity error in the data read from the receive FIFO	R
b12	FER	Framing Error Flag	(Valid only in asynchronous mode) 0: There is no framing error in the data read from the receive FIFO 1: There is framing error in the data read from the receive FIFO	R
b23 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	ORER	Overrun Error Flag	SSR.ORER flag can be read.	R
b26, b25	—	Reserved	These bits are read as 0. The write value should be 0.	R
b27	APER	Aggregate Parity Error Flag	SSR.APER flag can be read.	R
b28	AFER	Aggregate Framing Error Flag	SSR.AFER flag can be read.	R
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R

In FIFO mode (SCR3.FM bit = 1), this register is 32-stage FIFO buffer configuration.

RDAT[8:0] Bits (Receive Data)

After one frame of data is received, the received data is transferred from the RSR register to this registers, thus allowing the RSR register to receive the next data.
The RSR and RDR registers have a double-buffered construction to enable continuous reception.
In non-FIFO mode, read the RDR register only once when a receive data full interrupt (RXI) request is issued. Without reading received data from RDR register, if the next one frame is received, an overrun error occurs.
In FIFO mode, continuous reception is executed until 32 stages are stored. If data is read when there is no received data in the receive FIFO (RDR register), the value is undefined. When the receive FIFO (RDR register) are full of received data, subsequent serial receive data is lost.
The CPU cannot write to RDR register.
0 is stored in the bit position which isn’t received (RDAT[8] or RDAT[7]) at the time of 7bit or 8bit communication of asynchronous and manchester mode.

MPB Flag (Multi-Processor Bit Monitor Flag)

In asynchronous mode and manchester mode, during multi-processor communication (SCR3.MP bit = 1), the value of the multi-processor bit corresponding to the received data (RDAT[8:0] bits) can be read.

PER Flag (Parity Error Flag)

Indicates whether the data read from the receive FIFO has a parity error.
The FER and PER flags store error information of received data only in FIFO mode. In non-FIFO mode, 0 is stored.

FER Flag (Framing Error Flag)

Indicates whether the data read from receive FIFO has a framing error.
The FER and PER flags store error information of received data only in FIFO mode. In non-FIFO mode, 0 is stored.

36.2.3 Transmit Data Register (TDR)

Address(es): RSCI10.TDR 000E 2004h, RSCI11.TDR 000E 2084h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SYNC	—	—	MPBT	TDAT[8:0]								
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	TDAT[8:0]	Transmit Data	TDAT[8:0] bits are a 9-bit field for setting transmit data. Transmit data is set in [6:0] when 7-bit data is selected, in [7:0] when 8-bit data is selected, and in [8:0] when 9-bit data is selected. When writing the TDR register in 8-bit units, write first to the TDR.LH and then TDR.LL.	R/W
b9	MPBT	Transmit Multi-Processor	Value of the multi-processor bit in the transmission frame. This bit is use in asynchronous and manchester mode. When writing to this bit when not used, write the initial value. 0: Data transmission cycles 1: ID transmission cycles	R/W
b11, b10	—	Reserved	These bits are read as 1. The write value should be 1.	R
b12	SYNC	Sync Pulse Select	It is valid when MMCR.SBLEN bit = 1 and MMCR.SYNCE bit = 1 in manchester mode. When writing to this bit when not used, write the initial value. 0: The Start Bit is transmitted as DATA Sync. 1: The Start Bit is transmitted as COMMAND Sync.	R/W
b31 to b13	—	Reserved	These bits are read as 1. The write value should be 1.	R

In FIFO mode (SCR3.FM bit = 1), this register is 32-stage FIFO buffer configuration.

TDAT[8:0] Bits (Transmit Data)

The TDAT[8:0] bits are a 9-bit field for storing transmit data.

When empty space is detected in the TSR register, the transmit data stored in the TDR register is transferred to TSR register, and transmitting is started.

The TSR and TDR registers have a double-buffered construction to realize continuous reception. When the next data to be transmitted is stored in TDR register after one frame of data has been transmitted, the transmitting operation is continued by transfer to the TSR register.

In FIFO mode, continuous serial transmission is executed until there is no transmit data left in the transmit FIFO (TDR register). When transmit FIFO is full of transmit data (32 frames), no more data can be written. If writing of new data is attempted, the data is ignored.

In non-FIFO mode, the TDR register is always readable and writable by the CPU. When a transmit data empty interrupt (TXI) request is issued and SCR0.TE bit is 1, write transmit data to the TDR register only once.

When writing the TDR register in 8-bit units, write first to the TDR.LH and then the TDR.LL.

MPBT Bit (Transmit Multi-Processor)

Selects the multi-processor bit of transmit frame.

SYNC Bit (Sync Pulse Select)

This bit is valid when the MMCR.SYNCE and MMCR.SBLEN bits are set to 1 in Manchester mode (SCR3.MOD [2: 0] bit = 101b).

The Sync type of start bit area in the transmission frame can be set to Data Sync or Command Sync.

36.2.4 Transmit Shift Register (TSR)

TSR register is a shift register that transmits serial data. TSR register cannot be directly accessed by the CPU.

To perform serial data transmission, the SCI first automatically transfers transmit data from TDR register to TSR register, and then sends the data to the TXDn pin.

36.2.5 Control Register 0 (SCR0)

Address(es): RSCI10.SCR0 000E 2008h, RSCI11.SCR0 000E 2088h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	SSE	—	—	TEIE	TIE	—	—	—	RIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	IDSEL	DCME	MPIE	—	—	—	TE	—	—	—	RE
Value after 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	RE	Receive Enable	0: Serial reception is disabled 1: Serial reception is enabled	R/W *1, *3
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	TE	Transmit Enable	0: Serial transmission is disabled 1: Serial transmission is enabled	R/W *1
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	MPIE	Multi-Processor Interrupt Enable	(Valid in asynchronous mode and manchester mode when SCR3.MP is 1.) This bit should set 0 in smart card interface mode. 0: Non-multi-processor reception 1: Multi-processor reception When the data with the multi-processor bit set to 0 is received, the data is not read, and setting the status flags to 1 is disabled. When the data with the multiprocessor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and non-multi-processor reception is resumed. If you want to continue receiving operation using the multiprocessor function, set this bit to 1 sufficiently earlier than receiving the STOP bit of the next received frame.	R/W *2
b9	DCME	Data Compare Match Enable	(Valid only in asynchronous mode) 0: Data match detection function is disabled 1: Data match detection function is enabled	R/W *2
b10	IDSEL	ID Frame Select	(Valid only in asynchronous mode with multi-processor) 0: It's always compared data in spite of the value of the multi-processor bit. 1: It's compared data when the multi-processor bit is 1 (ID frame) only.	R/W *4
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	RIE	Receive Interrupt Enable	0: RXI and ERI interrupt requests are disabled 1: RXI and ERI interrupt requests are enabled	R/W
b19 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	TIE	Transmit Interrupt Enable	0: TXI interrupt request is disabled 1: TXI interrupt request is enabled	R/W
b21	TEIE	Transmit End Interrupt Enable	This bit should set 0 in smart card interface mode. 0: TEI interrupt request is disabled 1: TEI interrupt request is enabled	R/W
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	SSE	SSn# Pin Function Enable	(Valid in simple SPI mode.) In slave mode (SCR3.CKE[1:0] bits = 1xb), set this bit to 1. 0: SSn# pin function is disabled. 1: SSn# pin function is enabled.	R/W *4
b31 to b25	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. In clock synchronous mode (SCR3.MOD[2:0] bits = 010b), simple SPI mode (SCR3.MOD[2:0] bits = 011b), and simple I^2C

mode (SCR3.MOD[2:0] bits = 100b), 1 can be written only when TE bit = 0 and RE bit = 0. After setting TE bit or RE bit to 1, only 0 can be written in TE bit and RE bit. In other mode, writing is enabled under any condition.

Note 2. This bit is a bit that is cleared by hardware. Note that writing to a bit other than this bit with a bit manipulation instruction may cause this bit to be unintentionally set to 1 by a read-modify-write operation.

Note 3. In clock synchronous mode and simple SPI mode, receive only setting with internal clock (master mode) is prohibited (TE bit = 0 and RE bit = 1 setting prohibited).

Note 4. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

RE Bit (Receive Enable)

Enables or disables serial receive operation.

When this bit is set to 1, serial reception starts when RSCI detects the start bit in synchronous mode, the falling edge of the RXD input in Manchester mode, the synchronous clock input in clock synchronous mode, or the start bit in smart card interface mode.

Note that the SCR0 and SCR3 registers should be set prior to setting the RE bit to 1 in order to designate the reception format.

Except smart card interface mode, even if reception is halted by setting the RE bit to 0, the SSR.RDRF, AFER, APER, ORER, MMSR.MCER, SBER, SYER, and PFER flags in non-FIFO mode, and RFSR.DR flag in FIFO mode are not affected and the previous values is retained. In smart card interface mode, even if reception is halted by setting the RE bit to 0, the SSR.AFER, APER, and ORER flags 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 becomes possible. Transmission is started by writing transmit data to TDR register. Note that SCR3 should be set prior to setting the TE bit to 1 in order to designate the transmission format.

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 (SSR.RDRF, ORER, AFER, RFSR.DR, MMSR.MCER, SYER, PFER, SBER) are not set.

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, refer to section 36.4, Multi-Processor Communication Function. If you want to continue receiving operation using the multiprocessor function, set this bit to 1 sufficiently earlier than receiving the STOP bit of the next received frame.

When the receive data includes the multi-processor bit set to 0, the receive data is not transferred from the RSR register to the RDR register, a receive error is not detected, and setting the flags ORER and AFER, MCER, SYER, PFER, and SBER to 1 is disabled.

When the receive data includes the multi-processor bit set to 1, the MPB flag is set to 1, the MPIE bit is automatically set to 0, the RXI and ERI interrupt requests are enabled (if SCR0.RIE bit is set to 1), and setting the flags ORER, AFER, MCER, SYER, PFER, and SBER to 1 is enabled.

MPIE should be set to 0 if multi-processor communications function is not to be used.

DCME Bit (Data Compare Match Enable)

It can select whether the data match detection function uses or not.

When DCME bit is 1, if RSCI detects the match to the comparison data (SCR4.CMPD[8:0] bits) with receive data, DCME bit is cleared automatically, and after that, RSCI operation mode will be receive mode without data match detection function.

Refer to section 36.3.6, Data Match Detection.

The write value should be 0 other than asynchronous mode.

IDSEL Bit (ID Frame Select)

This bit specifies the condition of the received data to be compared. The bit is valid only when the DCME bit is 1.
When setting this bit to 1, only data in received frames (ID frames) in which the multi-processor bit has the value 1 are compared.
When this bit is set to 0, all received data is compared.

RIE Bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.
RXI and ERI interrupt request is disabled by setting the RIE bit to 0.
An ERI interrupt request can be canceled by reading 1 from the SSR.ORER, AFER, or APER flag and then setting the flag to 0, or setting the RIE bit to 0.
In the case of manchester mode, the MMSR.MCER, SYER, PFER, and SBER flags are also the cause of ERI interrupt request, so the same processing is necessary. For details of these flags, see section 36.2.12, Manchester Mode Control Register (MMCR) and section 36.2.21, Manchester Mode Status Register (MMSR).

TIE Bit (Transmit Interrupt Enable)

Enables or disables TXI interrupt request.
An TXI interrupt request is disabled by setting the TIE bit to 0. At the beginning of transmission, set 1 to SCR0.TE bit and SCR0.TIE bit simultaneously. Then the TXI interrupt request is generated.

TEIE Bit (Transmit End Interrupt Enable)

T Enables or disables a TEI interrupt request.
A TEI interrupt request is disabled by setting the TEIE bit to 0.
In simple I^2C mode, the TEI is allocated to the interrupt on completion of issuing a start, restart, or stop condition (STI). In this case, the TEIE bit can be used to enable or disable the STI.

SSE Bit (SSn# Pin Function Enable)

Set this bit to 1 if the SSn# pin is to be used in control of transmission and reception (in simple SPI mode). Set this bit to 0 in any other mode. Do not set both the SSE and CTSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).
In the slave mode (SCR3.CKE[1:0] bits = 10b or 11b), SSE should be set 1.
In the master mode (SCR3.CKE[1:0] bits = 00b or 01b) and single-master, the SSn# pin on the master side is not required to control reception and transmission, so SSE should be set 0.

36.2.6 Control Register 1 (SCR1)

Address(es): RSCI10.SCR1 000E 200Ch, RSCI11.SCR1 000E 208Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	NFEN	—	NFCS[2:0]			—	—	—	HDSEL	—	—	—	LOOP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RINV	TINV	—	—	PM	PE	—	—	SPB2IO	SPB2DT	—	—	CRSEP	CTSE
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
b0	CTSE	CTS Enable	0: CTS function is disabled (RTS output function is enabled). 1: CTS function is enabled.	R/W *1
b1	CRSEP	CTS/RTS Separation*2	0: Use either CTS or RTS function 1: Use both CTS and RTS functions at the same time	R/W *1
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	SPB2DT	Serial Port Break Data	The output level of TXDn (TXDAn/TXDBn*5) pin is selected when SCR0.TE bit = 0 and SPB2IO bit = 1.*3 When TINV is 0, 0: Low level is output in TXDn (TXDAn/TXDBn*5) pin. 1: High level is output in TXDn (TXDAn/TXDBn*5) pin. When TINV is 1, 0: High level is output in TXDn (TXDAn/TXDBn*5) pin. 1: Low level is output in TXDn (TXDAn/TXDBn*5) pin.	R/W
b5	SPB2IO	Serial Port Break I/O	It's selected whether the value of SPB2DT is output to TXDn (TXDAn/TXDBn*5) pin when SCR0.TE = 0.*3 0: The value of SPB2DT bit isn't output in TXDn (TXDAn/TXDBn*5) pin. 1: The value of SPB2DT bit is output in TXDn (TXDAn/TXDBn*5) pin.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	PE	Parity Enable	(Valid only in asynchronous mode and manchester mode. In smart card interface mode, set 1 to this bit.) 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 *1
b9	PM	Parity Mode	(Valid only when the PE bit is 1) 0: Selects even parity 1: Selects odd parity	R/W *1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12	TINV	Transmitter Output Invert*4	0: Transmit data is not inverted and output to TXDn (TXDAn/TXDBn*5) pin. 1: Transmit data is inverted and output to TXDn (TXDAn/TXDBn*5) pin.	R/W *1
b13	RINV	Receiver Input Invert*4	0: Received data from RXDn is not inverted and input. 1: Received data from RXDn is inverted and input.	R/W *1
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	LOOP	Loopback Mode Setting	It can be used when internal clock operation in asynchronous mode, internal mode operation in manchester mode, internal clock operation in clock synchronous mode. 0: Normal mode 1: Loopback mode	R/W *1

Bit	Symbol	Bit Name	Description	R/W
b19 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	HDSEL	Half-Duplex Communication Select	In the smart card interface mode, the simple I2C mode, or in the simple SPI mode, this bit should be set 0. 0: TXDn pin, RXDn pin independent 1: TXDn/RXDn pin combination use (Half-duplex communication using TXDn pin)	R/W *1
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R
b26 to b24	NFCS[2:0]	Noise Filter Clock Select	(Valid in asynchronous mode and manchester mode, extended serial mode, and simple I2C mode.) Select for the noise filter's clock source. b26 b24 0 0 0: The base clock signal divided by 1. 0 0 1: The on-chip baud rate generator source clock*6 divided by 1. 0 1 0: The on-chip baud rate generator source clock*6 divided by 2. 0 1 1: The on-chip baud rate generator source clock*6 divided by 4. 1 0 0: The on-chip baud rate generator source clock*6 divided by 8. Settings other than above are prohibited. In simple I2C mode, 000b setting is prohibited. “The on-chip baud rate generator source clock” means the clock selected by SCR2.CKS[1:0] bits.	R/W *1
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R
b28	NFEN	Digital Noise Filter Enable	(Valid in asynchronous mode, manchester mode, extended serial mode and simple I2C) In asynchronous mode, manchester mode and extended serial 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 I2C 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.	R/W *1
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.
Note 2. This bit is available in asynchronous mode and manchester mode. Set this bit to 0 in other mode.
Note 3. Please use this bit in asynchronous mode and manchester mode only. Movement by other mode isn’t guaranteed.
Note 4. RINV/TINV should be set to 0 in smart card interface mode and simple I2C mode.
Note 5. When the alternate output is enabled in HBS support mode.
Note 6. The clock is selected by SCR2.CKS[1:0] bits.

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, simple I2C mode, and extended serial mode. Do not set both the CTSE and SSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

CRSEP Bit (CTS/RTS Separation)

This bit selects usage of the CTSn#, RTSn#, and CTSn#/RTSn# pins when the CTSE bit is 1.
When either CTS or RTS function is to be used, set this bit to 0.
When both CTS and RTS functions are to be used, set this bit to 1.
When the CTSE bit is set to 0, set this bit to 0.
Refer to Table 36.6 for the relationship between the CSREP and CTSE bit settings and the pin functions.

Table 36.6 Relationship between the CRSEP and CTSE Bit Settings and Pin Functions

CTSE Bit	CRSEP Bit	CTSn#/RTSn# Multiplexed Pin	CTSn# Dedicated Pin	RTSn# Dedicated Pin
0	0	RTSn# signal output	Disabled	RTSn# signal output
1	0	CTSn# signal input	CTSn# signal input	Disabled
1	1	RTSn# signal output	CTSn# signal input	RTSn# signal output

SPB2DT Bit (Serial Port Break Data), SPB2IO Bit (Serial Port Break I/O)

The TXDn (TXDAn/TXDBn) pins status decided by combination of SCR0.TE bit, SCR1.SPB2IO bit and SCR1.SPB2DT bit is indicated in Table 36.7.

Table 36.7 Controlling the TXDn (TXDAn/TXDBn) Pins

SCR0.TE Bit Setting	SCR1.SPB2IO Bit Setting	SCR1.SPB2DT Bit Setting	TINV Bit Setting	TXDn Pin Status
0 (Transmission disabled)	0 (Input)	0 or 1	0 or 1	Hi-Z
	1 (Output)	0	0	Low is output
			1	High is output
		1	0	High is output
			1	Low is output
1 (Transmission enabled)	0 or 1	0 or 1	0 or 1	Transmit data output pin

PE Bit (Parity Enable)

When PE bit to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception. In the multiprocessor format, the parity bit is not added or checked regardless of this bit setting.

PM Bit (Parity Mode)

Selects the parity mode for transmission and reception (even or odd). In multi-processor mode, this bit is invalid.
For details on the usage of this bit in smart card interface mode, refer to section 36.7.2, Data Format (Except in Block Transfer Mode).

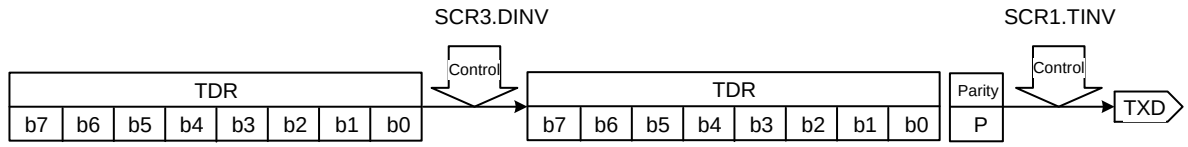
TINV Bit (Transmitter Output Invert), RINV Bit (Receiver Input Invert)

The data of RDR register is controlled by RINV bit and SCR3.DINV bit. And the data from TXDn pin is controlled by TINV bit and SCR3.DINV bit. The control by RINV/TINV bits are done to communication pins (RXDn/TXDn), so they can control not only data bits but also other bits (start bit, stop bit, parity bit). Please refer to Figure 36.2 in detail. When the TXDAn/TXDBn pins are used, the data is also inverted according to the TINV value.
During half-duplex communication and slave operation in simple SPI mode, use the TXDn pin for reception, so set the inversion control of the received data with the TINV bit.

Note: Sentences and a timing chart of the RSCI operation explanation are mentioned by TINV bit = 0 and RINV bit = 0 when TINV's value and RINV's value are not specified.

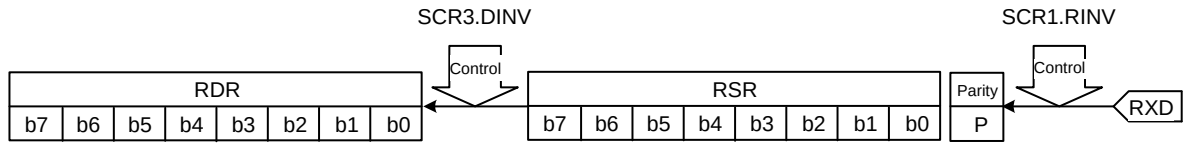
The receive/transmit data control (Data size = 8bits, Even parity, MSB first)

The transmit data is controlled by SCR1.TINV and SCR3.DINV.



SCR3. DINV	SCR1. TINV	TDR	TSR	Prity (even)	TXDn waveform
0	0	BEh	BEh	0	
0	1	BEh	BEh	0	
1	0	BEh	41h	0	
1	1	BEh	41h	0	

The received data is controlled by SCR1.RINV and SCR3.DINV.



SCR3. DINV	SCR1. RINV	RDR	RSR	Prity (even)	RXDn waveform
0	0	BEh	BEh	0	
1	0	41h	BEh	0	
0	1	BEh	BEh	0	
1	1	41h	BEh	0	

Figure 36.2 Example of the Receive/Transmit Data Control

LOOP Bit (Loopback Mode Setting)

When this bit is 1, RSCI blocks the input path from RXD and connects the output path to TXD to the reception data register.
Transmit data can be inverted and received by combining it with TINV bit.
Clock synchronous mode Set to 0 at slave operation and asynchronous mode use of external clock, and extended serial mode.

HDSEL Bit (Half-Duplex Communication Select)

Setting this bit to 1 enables half-duplex communication using the TXDn pin. However, it cannot be used in simple SPI mode, simple I^2C mode and smart card interface mode.
If this bit is set to 1 and SCR0.TE bit = 1, SCR0.RE bit = 0, the TXDn pin becomes communication output. If this bit is set to 1 and SCR0.TE bit = 0, SCR0.RE bit = 1, the TXDn pin becomes the communication input. For details, see section 36.16, Half-Duplex Communication Function.

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, manchester mode and extended serial mode set these bits from 000b to 100b. In simple I^2C mode, set the bits to a value in the range from 001b to 100b.

NFEN Bit (Digital Noise Filter 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, manchester mode, extended serial 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.

36.2.7 Control Register 2 (SCR2)

Address(es): RSCI10.SCR2 000E 2010h, RSCI11.SCR2 000E 2090h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MDDR[7:0]								—	—	CKS[1:0]		—	—	—	BRME
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BRR[7:0]								—	ABCSE	ABCS	BGDM	—	BCP[2:0]		
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	BCP[2:0]	Base Clock Pulse	Selects the number of base clock cycles in smart card interface mode. b2 b0 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 *1
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b4	BGDM	Baud Rate Generator Double-Speed Mode Select	Valid in asynchronous/manchester/clock synchronous/simple SPI mode and SCR3.CKE[1] bit = 0. 0: Baud rate generator outputs the clock with single frequency. 1: Baud rate generator outputs the clock with doubled frequency.	R/W *1
b5	ABCS	Asynchronous Mode Base Clock Select	(Valid only in asynchronous mode, manchester mode and extended serial mode) 0: Selects 16 base clock cycles for 1-bit period. 1: Selects 8 base clock cycles for 1-bit period.	R/W *1
b6	ABCSE	Asynchronous Mode Base Clock Select Extended	(Valid only in asynchronous mode and SCR3.CKE[1] bit = 0) 0: Clock cycles for 1-bit period is decided with combination between SCR2.BGDM bit and SCR2.ABCS bit. 1: Baud rate is 6 base clock cycles for 1-bit period and the clock of a double frequency is output from the baud rate generator.	R/W *1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R
b15 to b8	BRR[7:0]	Bit Rate Setting	An 8-bit field that adjusts the bit rate.	R/W *1
b16	BRME	Bit Rate Modulation Enable	0: Bit rate modulation function is disabled. 1: Bit rate modulation function is enabled.	R/W *1
b19 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b21, b20	CKS[1:0]	Clock Select	b21 b20 0 0: PCLKA (n = 0)*3 0 1: PCLKA/4 (n = 1)*3 1 0: PCLKA/16 (n = 2)*3 1 1: PCLKA/64 (n = 3)*3	R/W *1
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R
b31 to b24	MDDR[7:0]	Modulation Duty Setting	MDDR[7:0] bits corrects the bit rate adjusted by the BRR[7:0] bits.	R/W *1

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.
Note 2. S is the value of S in BRR[7:0] bits explanation.
Note 3. n is the decimal notation of the value of n in BRR[7:0] bits explanation.

BCP[2:0] 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.
For details, refer to section 36.7.4, Receive Data Sampling Timing and Reception Margin.

BGDM Bit (Baud Rate Generator Double-Speed Mode Select)

This bit is valid when the on-chip baud rate generator is selected as the clock source (SCR3.CKE[1] bit = 0) in asynchronous mode, manchester mode, clock synchronous mode, simple SPI mode. When external clock is selected (SCR3.CKE[1] bit = 1), set it to 0. For the clock output from the baud rate generator, either single or doubled frequency can be selected. The base clock is generated by the clock output from the baud rate generator. When the BGDM bit is set to 1, the base clock cycle is halved and the bit rate is doubled.
Set this bit to 0 in modes other than asynchronous mode or manchester mode or clock synchronous mode or simple SPI.

ABCS Bit (Asynchronous Mode Base Clock Select)

Selects the clock cycles for 1-bit period.
Set it to 0 in modes other than asynchronous mode, manchester mode and extended serial mode.

ABCSE Bit (Asynchronous Mode Base Clock Select Extended)

The pulse number for a base clock at 1-bit period is 6 and the clock of a double frequency is output from baud rate generator. Only when the bit rate is set to 6 dividing frequency of the bus clock, please use this bit and set SCR2.CKS[1:0] bits = 00b and BRR[7:0] bits = 0.
Set it to 0 in modes other than asynchronous mode. Even in asynchronous mode, set it to 0 when using external clock.

Table 36.8 Base Clock Cycle Number per 1-Bit

ABCSE Bit	ABCS Bit	BGDM Bit	The Base Clock Cycles/ 1-Bit	The Output Frequency of the Baud Rate Generator
0	0	0	16	×1
0	0	1	16	×2
0	1	0	8	×1
1	1	1	8	×2
1	—	—	6	×2

—: Don’t care

BRR[7:0] Bits (Bit Rate Setting)

BRR[7:0] bits are an 8-bit field that adjusts the bit rate.
RSCI has independent baud rate generator control, different bit rates can be set for each. Table 36.9 shows the relationship between the setting (N) in the BRR[7:0] bits and the bit rate (B) for asynchronous mode, multiprocessor transfer, manchester mode, clock synchronous mode, smart card interface mode, simple SPI mode, and simple I^2C mode.

Table 36.9 Relationship between N Setting in BRR[7:0] Bits and Bit Rate B

Mode	SCR2 Settings			BRR[7:0] Bits Setting	Error (%)
	BGDM bit	ABCS bit	ABCSE bit		
Asynchronous, multi-processor communication, manchester, extended serial*3	0	0	0	$N = \frac{PCLKA \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLKA \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	1	0	0	$N = \frac{PCLKA \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLKA \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	0	1	0		
	1	1	0	$N = \frac{PCLKA \times 10^6}{16 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLKA \times 10^6}{B \times 16 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	0 or 1	0 or 1	1*2	$N = \frac{PCLKA \times 10^6}{12 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLKA \times 10^6}{B \times 12 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
Clock synchronous, simple SPI	0	0 (Initial value)	0 (Initial value)	$N = \frac{PCLKA \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	
	1	0 (Initial value)	0 (Initial value)	$N = \frac{PCLKA \times 10^6}{4 \times 2^{2n-1} \times B} - 1$	
Smart card interface				$N = \frac{PCLKA \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$Error = \left\{ \frac{PCLKA \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1 \right\} \times 100$
Simple I^2C*1				$N = \frac{PCLKA \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	

B: Bit rate (bps)
N: BRR[7:0] bits setting ($0 \le N \le 255$)
PCLKA: Operating frequency (MHz)
n and S: Determined by the settings of the SCR2 registers as listed in the table below.Please be careful about "$2^{(2n+1)}$" is used in the expression for smart card interface, "$2^{(2n-1)}$" is used in other mode.

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.
Note 2. In manchester mode, only ABCSE bit = 0 can be selected.
Note 3. In extended serial mode, BGDM bit = 0 and ABCSE bit = 0 can be selected.

Table 36.10 Calculating Widths at High and Low Level for SCL

Mode	SCL	Formula (Result in Seconds)
I^2C	High period (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 7 \times \frac{1}{PCLKA \times 10^6}$
	Low period (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 8 \times \frac{1}{PCLKA \times 10^6}$

Table 36.11 Clock Source Settings

SCR2 Setting		
CKS[1:0] Bits	Clock Source	n
0 0	PCLKA	0
0 1	PCLKA/4	1
1 0	PCLKA/16	2
1 1	PCLKA/64	3

Table 36.12 Base Clock Settings in Smart Card Interface Mode

SCR2 Setting		
BCP[2:0] Bits	**Base Clock Cycles for 1-bit Period**	**S**
0 0 0	93 clock cycles	93
0 0 1	128 clock cycles	128
0 1 0	186 clock cycles	186
0 1 1	512 clock cycles	512
1 0 0	32 clock cycles	32
1 0 1	64 clock cycles	64
1 1 0	372 clock cycles	372
1 1 1	256 clock cycles	256

Table 36.13 and Table 36.14 list examples of N settings in BRR[7:0] in asynchronous mode and manchester mode. Table 36.15 lists the maximum bit rate settable for each operating frequency. Examples of BRR[7:0] bits (N) settings in clock synchronous mode and simple SPI mode are listed in Table 36.17. Examples of BRR[7:0] bits (N) settings in smart card interface mode are listed in Table 36.19. Examples of BRR[7:0] bits (N) settings in simple I^2C mode are listed in Table 36.21. In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, refer to section 36.7.4, Receive Data Sampling Timing and Reception Margin. Table 36.16 and Table 36.18 list the maximum bit rates with external clock input.

When either the asynchronous mode base clock select bit (ABCS) or the baud rate generator double-speed mode select bit (BGDM) is set to 1 in asynchronous mode and manchester mode, the bit rate becomes twice that listed in Table 36.13 and Table 36.14. When both of those registers are set to 1, the bit rate becomes four times the listed value.

Table 36.13 Examples of BRR[7:0] Bits Settings for Various Bit Rates (Asynchronous Mode and Manchester Mode) (1)

Bit Rate (bps)	Operating Frequency PCLKA (MHz)														
	8			9.8304			10			12			12.288		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	25	0.16	0	31	0.00	0	32	–1.36	0	38	0.16	0	39	0.00
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	–2.34	0	19	0.00
31250	0	7	0.00	0	9	–1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	—	—	—	0	7	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

Bit Rate (bps)	Operating Frequency PCLKA (MHz)														
	14			16			17.2032			18			19.6608		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	–0.17	3	70	0.03	3	75	0.48	3	79	–0.12	3	86	0.31
150	2	181	0.16	2	207	0.16	2	223	0.00	2	233	0.16	2	255	0.00
300	2	90	0.16	2	103	0.16	2	111	0.00	2	116	0.16	2	127	0.00
600	1	181	0.16	1	207	0.16	1	223	0.00	1	233	0.16	1	255	0.00
1200	1	90	0.16	1	103	0.16	1	111	0.00	1	116	0.16	1	127	0.00
2400	0	181	0.16	0	207	0.16	0	223	0.00	0	233	0.16	0	255	0.00
4800	0	90	0.16	0	103	0.16	0	111	0.00	0	116	0.16	0	127	0.00
9600	0	45	–0.93	0	51	0.16	0	55	0.00	0	58	–0.69	0	63	0.00
19200	0	22	–0.93	0	25	0.16	0	27	0.00	0	28	1.02	0	31	0.00
31250	0	13	0.00	0	15	0.00	0	16	1.20	0	17	0.00	0	19	–1.70
38400	—	—	—	0	12	0.16	0	13	0.00	0	14	–2.34	0	15	0.00

Note: This is an example when the SCR2.ABCS bit = 0, SCR2.BGDM bit = 0 and SCR2.ABCSE bit = 0.
When either ABCS bit or BGDM bit is set to 1, the bit rate doubles.
When both ABCS bit = 1 and BGDM bit = 1, the bit rate increases four times.

Table 36.14 Examples of BRR[7:0] Bits Settings for Various Bit Rates (Asynchronous Mode and Manchester Mode) (2)

Bit Rate (bps)	Operating Frequency PCLKA (MHz)														
	20			25			30			33			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	88	–0.25	3	110	–0.02	3	132	0.13	3	145	0.33	3	177	–0.25
150	3	64	0.16	3	80	0.47	3	97	–0.35	3	106	0.39	3	129	0.16
300	2	129	0.16	2	162	–0.15	2	194	0.16	2	214	–0.07	3	64	0.16
600	2	64	0.16	2	80	0.47	2	97	–0.35	2	106	0.39	2	129	0.16
1200	1	129	0.16	1	162	–0.15	1	194	0.16	1	214	–0.07	2	64	0.16
2400	1	64	0.16	1	80	0.47	1	97	–0.35	1	106	0.39	1	129	0.16
4800	0	129	0.16	0	162	–0.15	0	194	0.16	0	214	–0.07	1	64	0.16
9600	0	64	0.16	0	80	0.47	0	97	–0.35	0	106	0.39	0	129	0.16
19200	0	32	–1.36	0	40	–0.76	0	48	–0.35	0	53	–0.54	0	64	0.16
31250	0	19	0.00	0	24	0.00	0	29	0.00	0	32	0.00	0	39	0.00
38400	0	15	1.73	0	19	1.73	0	23	1.73	0	26	–0.54	0	32	–1.36

Bit Rate (bps)	Operating Frequency PCLKA (MHz)											
	50			60			100			120		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	221	–0.02	—	—	—	—	—	—	—	—	—
150	3	162	–0.15	3	194	0.16	—	—	—	—	—	—
300	3	80	0.47	3	97	–0.35	3	162	–0.15	3	194	0.16
600	2	162	–0.15	2	194	0.16	3	80	0.47	3	97	–0.35
1200	2	80	0.47	2	97	–0.35	2	162	–0.15	2	194	0.16
2400	1	162	–0.15	1	194	0.16	2	80	0.47	2	97	–0.35
4800	1	80	0.47	1	97	–0.35	1	162	–0.15	1	194	0.16
9600	0	162	–0.15	0	194	0.16	1	80	0.47	1	97	–0.35
19200	0	80	0.47	0	97	–0.35	0	162	–0.15	0	194	0.16
31250	0	49	0.00	0	59	0.00	1	24	0.00	0	119	0.00
38400	0	40	–0.76	0	48	–0.35	0	80	0.47	0	97	–0.35

Note: This is an example when the SCR2.ABCS bit = 0, SCR2.BGDM bit = 0 and SCR2.ABCSE bit = 0.
When either ABCS bit or BGDM bit is set to 1, the bit rate doubles.
When both ABCS bit = 1 and BGDM bit = 1, the bit rate increases four times.

Table 36.15 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode and Manchester Mode) (1)

PCLKA (MHz)	SCR2 Settings					Maximum Bit Rate (bps)	PCLKA (MHz)	SCR2 Settings					Maximum Bit Rate (bps)
	BGDM Bit	ABCS Bit	ABCSE Bit	n	N			BGDM Bit	ABCS Bit	ABCSE Bit	n	N	
8	0	0	0	0	0	250000	19.6608	0	0	0	0	0	614400
		1	0	0	0	500000			1	0	0	0	1228800
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1000000			1	0	0	0	2457600
	0 or 1	0 or 1	1	0	0	1333333		0 or 1	0 or 1	1	0	0	3276800
9.8304	0	0	0	0	0	307200	20	0	0	0	0	0	625000
		1	0	0	0	614400			1	0	0	0	1250000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1228800			1	0	0	0	2500000
	0 or 1	0 or 1	1	0	0	1638400		0 or 1	0 or 1	1	0	0	3333333
10	0	0	0	0	0	312500	25	0	0	0	0	0	781250
		1	0	0	0	625000			1	0	0	0	1562500
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1250000			1	0	0	0	3125000
	0 or 1	0 or 1	1	0	0	1666667		0 or 1	0 or 1	1	0	0	4166667
12	0	0	0	0	0	375000	30	0	0	0	0	0	937500
		1	0	0	0	750000			1	0	0	0	1875000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1500000			1	0	0	0	3750000
	0 or 1	0 or 1	1	0	0	2000000		0 or 1	0 or 1	1	0	0	5000000
12.288	0	0	0	0	0	384000	33	0	0	0	0	0	1031250
		1	0	0	0	768000			1	0	0	0	2062500
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1536000			1	0	0	0	4125000
	0 or 1	0 or 1	1	0	0	2048000		0 or 1	0 or 1	1	0	0	5500000
14	0	0	0	0	0	437500	40	0	0	0	0	0	1250000
		1	0	0	0	875000			1	0	0	0	2500000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	1750000			1	0	0	0	5000000
	0 or 1	0 or 1	1	0	0	2333333		0 or 1	0 or 1	1	0	0	6666667
16	0	0	0	0	0	500000	50	0	0	0	0	0	1562500
		1	0	0	0	1000000			1	0	0	0	3125000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	2000000			1	0	0	0	6250000
	0 or 1	0 or 1	1	0	0	2666667		0 or 1	0 or 1	1	0	0	8333333
17.2032	0	0	0	0	0	537600	60	0	0	0	0	0	1875000
		1	0	0	0	1075200			1	0	0	0	3750000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	2150400			1	0	0	0	7500000
	0 or 1	0 or 1	1	0	0	2867200		0 or 1	0 or 1	1	0	0	10000000
18	0	0	0	0	0	562500	120	0	0	0	0	0	3750000
		1	0	0	0	1125000			1	0	0	0	7500000
	1	0	0	0	0			1	0	0	0	0	
		1	0	0	0	2250000			1	0	0	0	15000000
	0 or 1	0 or 1	1	0	0	3000000		0 or 1	0 or 1	1	0	0	20000000

Table 36.16 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

		Maximum Bit Rate (bps)	
PCLKA (MHz)	External Input Clock (MHz)	SCR2.ABCS Bit = 0	SCR2.ABCS Bit = 1
8	2.0000	125000	250000
9.8304	2.4576	153600	307200
10	2.5000	156250	312500
12	3.0000	187500	375000
12.288	3.0720	192000	384000
14	3.5000	218750	437500
16	4.0000	250000	500000
17.2032	4.3008	268800	537600
18	4.5000	281250	562500
19.6608	4.9152	307200	614400
20	5.0000	312500	625000
25	6.2500	390625	781250
30	7.5000	468750	937500
33	8.2500	515625	1031250
40	10.0000	625000	1250000
50	12.5000	781250	1562500
60	15.0000	937500	1875000
120	30.0000	1875000	3750000

Table 36.17 Examples of BRR[7:0] Bits Settings for Various Bit Rates (Clock Synchronous Mode, Simple SPI Mode)

Bit Rate (bps)	Operating Frequency PCLKA (MHz)														
	8			10			30			60			120		
	BGDM	n	N	BGDM	n	N	BGDM	n	N	BGDM	n	N	BGDM	n	N
250	0	3	124	0	3	177	—	—	—	—	—	—	—	—	—
500	0	2	249	0	3	77	0	3	233	—	—	—	—	—	—
1 k	0	2	124	0	3	38	0	3	116	0	3	233	—	—	—
2.5 k	0	2	49	0	1	249	0	3	46	0	3	93	0	3	187
5 k	0	2	24	0	1	124	0	2	93	0	3	46	0	3	93
10 k	0	1	49	0	0	249	0	2	46	0	2	93	0	3	46
25 k	0	2	4	0	1	24	0	1	74	0	1	149	0	2	74
50 k	0	1	9	0	0	49	0	0	149	0	1	74	0	1	149
100 k	0	1	4	0	0	24	0	0	74	0	0	149	0	1	74
250 k	0	1	1	0	0	9	0	0	29	0	1	14	0	1	29
500 k	0	1	0	0	0	4	0	0	14	0	0	29	0	1	14
1 M	0	0	1	1	0	4	1	0	14	0	0	14	0	0	29
2.5 M	—	—	—	0	0	0	0	0	2	0	0	5	0	1	2
5 M	—	—	—	1	0	0	1	0	2	0	0	2	0	0	5
7.5 M	—	—	—	—	—	—	0	0	0	0	0	1	1	1	0
60M	—	—	—	—	—	—	—	—	—	—	—	—	1	0	0

—: Can be set, but an error over 10% will occur.

Table 36.18 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode, Simple SPI Mode)

PCLKA (MHz)	External Input Clock (MHz)	Maximum Bit Rate (Mbps)
8	4	4
10	5	5
12	6	6
14	7	7
16	8	8
18	9	9
20	10	10
25	12.5	12.5
30	15	15
33	16.5	16.5
40	20	20
50	25	25
60	30	30
120	60	60

Table 36.19 BRR[7:0] Bits Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372)

Bit Rate (bps)	PCLKA (MHz)	n	N	Error (%)
9600	7.1424	0	0	0.00
	10.00	0	1	–30.00
	10.7136	0	1	–25.00
	13.00	0	1	–8.99
	14.2848	0	1	0.00
	16.00	0	1	12.01
	18.00	0	2	–15.99
	20.00	0	2	–6.66
	25.00	0	3	–12.49
	30.00	0	3	5.01
	33.00	0	4	–7.59
	40.00	0	5	–6.66
	50.00	0	6	0.01
	60.00	0	7	5.01
	120.00	0	16	–1.17

Table 36.20 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 32)

PCLKA (MHz)	Maximum Bit Rate (bps)	n	N
10.00	156250	0	0
10.7136	167400	0	0
13.00	203125	0	0
16.00	250000	0	0
18.00	281250	0	0
20.00	312500	0	0
25.00	390625	0	0
30.00	468750	0	0
33.00	515625	0	0
40.00	625000	0	0
50.00	781250	0	0
60.00	937500	0	0
120.00	1875000	0	0

Table 36.21 BRR[7:0] Bits Settings for Various Bit Rates (Simple I²C Mode)

Bit Rate (bps)	Operating Frequency PCLKA (MHz)														
	8			10			16			20			25		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10 k	0	24	0.0	0	31	–2.3	1	12	–3.8	1	15	–2.3	1	19	–2.3
25 k	0	9	0.0	0	12	–3.8	1	4	0.0	1	6	–10.7	1	7	–2.3
50 k	0	4	0.0	0	6	–10.7	1	2	–16.7	1	3	–21.9	1	3	–2.3
100 k	0	2	–16.7	0	3	–21.9	0	4	0.0	0	6	–10.7	1	1	–2.3
250 k	0	0	0.0	0	1	–37.5	0	1	0.0	0	2	–16.7	0	3	–21.9
350 k										0	1	–10.7	0	2	–25.6
400 k										0	1	–21.9	0	1	–2.3

Bit Rate (bps)	Operating Frequency PCLKA (MHz)														
	30			33			40			50			60		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10 k	1	23	–2.3	1	25	–0.8	0	124	0.00	2	9	–2.3	1	46	–0.27
25 k	1	9	–6.3	1	10	–6.3	0	40	0.00	2	3	–2.3	0	74	0.00
50 k	1	4	–6.3	1	5	–14.1	0	24	0.00	2	1	–2.3	0	37	–1.32
100 k	1	2	–21.9	1	2	–14.1	0	12	–3.85	1	3	–2.3	0	18	–1.32
250 k	0	3	–6.3	0	4	–17.5	0	4	0.00	0	6	–10.7	0	7	–6.25
350 k	0	2	–10.7	0	2	–1.8	0	3	–10.71	0	4	–10.7	0	4	7.14
400 k	0	1	17.2	0	2	–14.1	0	2	4.17	0	3	–2.34	0	4	–6.25

Bit Rate (bps)	Operating Frequency PCLKA (MHz)		
	120		
	n	N	Error (%)
10 k	1	93	–0.27
25 k	0	149	0.00
50 k	0	74	0.00
100 k	0	37	–1.31
250 k	0	14	0.00
350 k	0	10	–2.60
400 k	0	8	4.17

Table 36.22 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple I²C Mode)

Bit Rate (bps)	Operating Frequency PCLKA (MHz)											
	8			10			16			20		
	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)
10 k	0	24	43.75/50.00	0	31	44.80/51.20	1	12	45.50/52.00	1	15	44.80/51.20
25 k	0	9	17.50/20.00	0	12	18.20/20.80	1	4	17.50/20.00	1	6	19.60/22.40
50 k	0	4	8.75/10.00	0	6	9.80/11.20	1	2	10.50/12.00	1	3	11.20/12.80
100 k	0	2	5.25/6.00	0	3	5.60/6.40	0	4	4.37/5.00	0	6	4.90/5.60
250 k	0	0	1.75/2.00	0	1	2.80/3.20	0	1	1.75/2.00	0	2	2.10/2.40
350 k										0	1	1.40/1.60
400 k										0	1	1.40/1.60

Bit Rate (bps)	Operating Frequency PCLKA (MHz)											
	25			30			33			40		
	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)
10 k	1	19	44.80/51.20	1	23	44.80/51.20	1	25	44.12/50.42	1	32	46.20/52.80
25 k	1	7	17.92/20.48	1	9	18.66/21.33	1	10	18.66/21.33	1	12	18.20/20.80
50 k	1	3	8.96/10.24	1	4	9.33/10.66	1	5	10.18/11.63	1	6	9.80/11.20
100 k	1	1	4.48/5.12	1	2	5.60/6.40	1	2	5.09/5.81	0	13	4.90/5.60
250 k	0	3	2.24/2.56	0	3	1.86/2.13	0	4	2.12/2.42	0	4	1.75/2.00
350 k	0	2	1.68/1.92	0	2	1.40/1.60	0	2	1.27/1.45	0	3	1.40/1.60
400 k	0	1	1.12/1.28	0	1	0.93/1.07	0	2	1.27/1.45	0	2	1.05/1.20

Bit Rate (bps)	Operating Frequency PCLKA (MHz)								
	50			60			120		
	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)
10 k	2	9	44.80/51.20	1	47	44.80/51.20	1	93	43.87/50.13
25 k	2	3	17.92/20.48	0	74	17.50/20.00	0	149	17.50/20.00
50 k	2	1	8.96/10.24	0	37	8.87/10.13	0	74	8.75/10.00
100 k	1	3	4.48/5.12	0	18	4.43/5.07	0	37	4.43/5.07
250 k	0	6	1.96/2.24	0	7	1.87/2.13	0	15	1.87/2.13
350 k	0	4	1.40/1.60	0	5	1.40/1.60	0	10	1.28/1.47
400 k	0	3	1.12/1.28	0	4	1.17/1.33	0	9	1.17/1.33

BRME Bit (Bit Rate Modulation Enable)

Enables and disables the bit rate modulation function. The bit rate generated by on-chip baud rate generator is evenly corrected when this function is enabled.

This bit can only be set to 1 in asynchronous mode and simple I²C mode. Set this bit to 0 in clock synchronous mode, simple SPI mode, smart card interface mode, manchester mode, and extended serial mode.

CKS[1:0] Bits (Clock Select)

These bits select the clock source for the on-chip baud rate generator.

For the relationship between the settings of these bits and the baud rate, refer to BRR[7:0] bits explanation.

MDDR[7:0] Bits (Modulation Duty Setting)

When the BRME bit is set to 1, the bit rate generated by the on-chip baud rate generator is evenly corrected according to the settings of MDDR (M/256). The relationship between the MDDR[7:0] bits setting (M) and the bit rate (B) is given in Table 36.23.

The initial value of MDDR[7:0] bits is FFh. Bit 7 in this register is fixed to 1.

Table 36.23 Relationship between MDDR[7:0] Bits Setting (M) and Bit Rate (B) When Bit Rate Modulation Function is Used

Mode*1	SCR2 Settings			BRR[7:0] Bits Setting	Error (%)
	BGDM Bit	ABCS Bit	ABCSE Bit		
Asynchronous, multi-processor communication	0	0	0	$N = \frac{PCLKA \times 10^6}{64 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLKA \times 10^6}{B \times 64 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
	1	0	0	$N = \frac{PCLKA \times 10^6}{32 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLKA \times 10^6}{B \times 32 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
	0	1	0		
	1	1	0	$N = \frac{PCLKA \times 10^6}{16 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLKA \times 10^6}{B \times 16 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
	0 or 1	0 or 1	1	$N = \frac{PCLKA \times 10^6}{12 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLKA \times 10^6}{B \times 12 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
Simple I²C*2				$N = \frac{PCLKA \times 10^6}{64 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	

B: Bit rate (bps)
M: MDDR[7:0] bits setting (128 ≤ M ≤ 255)
N: BRR[7:0] bits setting (0 ≤ N ≤ 255)
PCLKA: Operating frequency (MHz)
n: Determined by the settings of the SCR2.CKS[1:0] bits as listed in Table 36.11, Clock Source Settings.

Note 1. Do not use this function in clock synchronous mode, simple SPI mode, smart card Interface mode, manchester mode and extended serial mode.

Note 2. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I²C mode satisfy the I²C standard.

Table 36.24 and Table 36.25 list examples of N settings in BRR[7:0] bits and M settings in MDDR[7:0] bits in asynchronous mode.

Table 36.24 Examples of BRR[7:0] Bits and MDDR[7:0] Bits Settings for Various Bit Rates (Asynchronous Mode) (1)

Bit Rate (bps)	Operating Frequency PCLKA (MHz)														
	8					9.8304					10				
	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)
38400	0	5	236	0	0.03	0	7	(256) *1	0	0.00	0	10	173	1	–0.01
57600	0	3	236	0	0.03	0	4	240	0	0.00	0	4	236	0	0.03
115200	0	1	236	0	0.03	0	1	192	0	0.00	0	4	236	1	0.03
230400	0	0	236	0	0.03	0	0	192	0	0.00	0	1	189	1	0.14
460800	0	0	236	1	0.03	0	0	192	1	0.00	0	0	189	1	0.14

Bit Rate (bps)	Operating Frequency PCLKA (MHz)														
	12					12.288					14				
	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)
38400	0	8	236	0	0.03	0	9	(256) *1	0	0.00	0	16	191	1	0.00
57600	0	5	236	0	0.03	0	4	192	0	0.00	0	13	236	1	0.03
115200	0	2	236	0	0.03	0	4	192	1	0.00	0	6	236	1	0.03
230400	0	2	236	1	0.03	0	2	230	1	–0.17	0	2	202	1	–0.11
460800	0	0	157	1	–0.18	0	0	154	1	0.26	0	0	135	1	0.14

Bit Rate (bps)	Operating Frequency PCLKA (MHz)														
	16					17.2032					18				
	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)
38400	0	11	236	0	0.03	0	13	(256) *1	0	0.00	0	18	166	1	–0.01
57600	0	7	236	0	0.03	0	6	192	0	0.00	0	18	249	1	–0.01
115200	0	3	236	0	0.03	0	6	192	1	0.00	0	8	236	1	0.03
230400	0	1	236	0	0.03	0	3	219	1	–0.20	0	1	210	0	0.14
460800	0	1	236	1	0.03	0	1	219	1	–0.20	0	0	210	0	0.14

Note: This is an example when the SCR2.ABCS bit = 0 and SCR2.ABCSE bit = 0.
Note 1. It means that the bit rate modulation function is disable. (SCR2.BRME bit = 0, M = 256)

Table 36.25 Examples of BRR[7:0] Bits and MDDR[7:0] Bits Settings for Various Bit Rates (Asynchronous Mode) (2)

Bit Rate (bps)	Operating Frequency PCLKA (MHz)														
	19.6608					20					25				
	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)
38400	0	15	(256) *1	0	0.00	0	10	173	0	–0.01	0	11	151	0	0.00
57600	0	9	240	0	0.00	0	9	236	0	0.03	0	7	151	0	0.00
115200	0	4	240	0	0.00	0	4	236	0	0.03	0	3	151	0	0.00
230400	0	1	192	0	0.00	0	4	236	1	0.03	0	1	151	0	0.00
460800	0	0	192	0	0.00	0	0	189	0	0.14	0	0	151	0	0.00

Bit Rate (bps)	Operating Frequency PCLKA (MHz)														
	30					33					40				
	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)
38400	0	36	194	1	0.01	0	14	143	0	0.01	0	21	173	0	–0.01
57600	0	10	173	0	–0.01	0	9	143	0	0.01	0	38	230	1	–0.01
115200	0	10	173	1	–0.01	0	4	143	0	0.01	0	9	236	0	0.03
230400	0	6	220	1	–0.09	0	4	143	1	0.01	0	4	236	0	0.03
460800	0	3	252	1	0.14	0	1	229	0	0.10	0	4	236	1	0.03

Bit Rate (bps)	Operating Frequency PCLKA (MHz)														
	50					60					120				
	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)	n	N	M	BGDM Bit	Error (%)
38400	0	23	151	0	0.00	0	36	194	0	0.01	0	73	194	0	0.01
57600	0	15	151	0	0.00	0	21	173	0	–0.01	0	58	232	0	0.00
115200	0	7	151	0	0.00	0	10	173	0	–0.01	0	21	173	0	–0.01
230400	0	3	151	0	0.00	0	10	173	1	–0.01	0	10	173	0	–0.01
460800	0	1	151	0	0.00	0	6	220	1	–0.09	0	10	173	1	–0.01

Note: This is an example when the SCR2.ABCS bit = 0 and SCR2.ABCSE bit = 0.
Note 1. It means that the bit rate modulation function is disable. (SCR2.BRME bit = 0, M = 256)

36.2.8 Control Register 3 (SCR3)

Address(es): RSCI10.SCR3 000E 2014h, RSCI11.SCR3 000E 2094h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	BLK	GM	—	—	CKE[1:0]		—	—	DEEN	FM	MP	MOD[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
	RXDESEL	STOP	DINV	DDIR	—	—	CHR[1:0]		—	—	—	—	—	—	CPOL	CPHA
Value after reset:	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	CPHA	Clock Phase Select	(Valid in clock synchronous mode and simple SPI mode. Set this bit only when SCR0.TE bit = 0 and RE bit = 0.) 0: Data is sampled at an odd edge and changes at an even edge. (Clock is delayed.) 1: Data changes at an odd edge and is sampled at an even edge. (Clock is not delayed.)	R/W *1
b1	CPOL	Clock Polarity Select	(Valid in clock synchronous mode and simple SPI mode. Set this bit only when SCR0.TE bit = 0 and RE bit = 0.) 0: SCKn in idle state is 0. 1: SCKn in idle state is 1.	R/W *1
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b9, b8	CHR[1:0]	Character Length Select	(Valid in asynchronous mode and manchester mode) *2 Select the data length for transmission and reception. b9 b8 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length*3	R/W *1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12	DDIR	Transfer Data Direction Select	0: MSB first 1: LSB first Set this bit to 0 in simple I^2C mode and set this bit to 1 in extended serial mode.	R/W *1
b13	DINV	Transfer Data Invert	0: TDR register contents are transmitted to TSR register as they are. RSR register contents are stored to RDR register as they are. 1: TDR register contents are inverted before being transmitted to TSR register. RSR register contents are inverted and stored to RDR register. Set this bit to 0 in simple I^2C mode. The level of communication pins (RXDn/TXDn) are controlled by combination of this bit and SCR1.TINV/RINV. Please refer to Figure 36.2 for details.	R/W *1
b14	STOP	Stop Bit Length Select	(Valid in asynchronous mode, manchester mode, extended serial mode) 0: 1 stop bit/break delimiter length is 1bit 1: 2 stop bits/break delimiter length is 2bits	R/W *1
b15	RXDESEL	Asynchronous Start Bit Edge Detection Select	(Valid only in asynchronous mode) Set this bit to 1 in extended serial mode. 0: The low level on the RXDn pin is detected as the start bit. 1: A falling edge on the RXDn pin is detected as the start bit.	R/W *1

Bit	Symbol	Bit Name	Description	R/W
b18 to b16	MOD[2:0]	Communication Mode Select	Select the RSCI communication mode. b18 b16 0 0 0: Asynchronous mode 0 0 1: Smart card interface mode 0 1 0: Clock synchronous mode 0 1 1: Simple SPI mode 1 0 0: Simple I^2C mode 1 0 1: Manchester mode 1 1 0: Extended serial mode 1 1 1: Setting prohibited	R/W *1
b19	MP	Multi-Processor Mode	(Valid in asynchronous mode, manchester mode) 0: Multi-processor communications function is disabled 1: Multi-processor communications function is enabled	R/W *1
b20	FM	FIFO Mode Select	(Valid in asynchronous mode (including multi-processor mode), clock synchronous mode, simple SPI mode) 0: TDR register, RDR register is non-FIFO buffer configuration 1: TDR register, RDR register is FIFO buffer configuration	R/W *1
b21	DEEN	Driver Control Function Enable	(Valid only in asynchronous mode) 0: RS-485 driver control function disable. 1: RS-485 driver control function enable.	R/W *1
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R
b25, b24	CKE[1:0]	Clock Enable	In the case of asynchronous mode b25 b24 0 0: On-chip baud rate generator The SCKn pin is available for use as an I/O port in accord with the I/O port settings. 0 1: On-chip baud rate generator The clock with the same frequency as the bit rate is output from the SCKn pin. 1 x: External clock • When using the external clock 16 times the bit rate should be input from the SCKn pin when SCR2.ABCS bit is 0. Input a clock signal with a frequency 8 times the bit rate when the SCR2.ABCS bit is 1. In the case of manchester mode and extended serial mode b25 b24 0 0: On-chip baud rate generator The SCKn pin functions as I/O port. Settings other than above are prohibited. In the case of clock synchronous mode and simple SPI mode b25 b24 0 x: Internal clock (master operation) The SCKn pin functions as the clock output pin. 1 x: External clock (slave operation) The SCKn pin functions as the clock input pin. In the case of smart card interface mode When SCR3.GM bit = 0 b25 b24 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: Prohibited When SCR3.GM bit = 1 b25 b24 0 0: Output fixed low 0 1: Clock output 1 0: Output fixed high 1 1: Clock output	R/W *1
b26	—	Reserved	Set this bit to 0.	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R
b28	GM	GSM Mode	(Valid only in smart card interface mode) 0: Non-GSM mode operation 1: GSM mode operation	R/W *1

Bit	Symbol	Bit Name	Description	R/W
b29	BLK	Block Transfer Mode	(Valid only in smart card interface mode) 0: Non-block transfer mode operation 1: Block transfer mode operation	R/W *1
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.
Note 2. In other than asynchronous mode and manchester mode, this bit setting is invalid and a fixed data length of 8 bits is used. Set these bits to 10b in extended serial mode.
Note 3. LSB first is fixed and the MSB (bit 7) in TDR register is not transmitted in transmission.

CPHA Bit (Clock Phase Select)

This bit selects the phase of the clock signal output through the SCKn pin. Refer to Figure 36.108 for details.
Set the bit to 1 in other than simple SPI mode and clock synchronous mode.

CPOL Bit (Clock Polarity Select)

This bit selects the polarity of the clock signal output through the SCKn pin. Refer to Figure 36.108 for details.
Set the bit to 1 in other than simple SPI mode and clock synchronous mode.

CHR[1:0] Bits (Character Length Select)

Selects the data length for transmission and reception.
Except of asynchronous mode and manchester mode, a fixed data length of 8 bits is used.

DDIR Bit (Transfer Data Direction Select)

Select whether to transmit/receive data in MSB first or LSB first.

DINV Bit (Transfer Data Invert)

DINV bit can invert the transmit data bit from TDR register to TSR register, and also can invert the received data from RSR register to RDR register. This bit does not affect the logic level of the parity bit. To invert the parity bit, invert the SCR1.PM bit.

STOP Bit (Stop Bit Length Select)

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.
In addition, it is used as break delimiter length setting when sending Start Frame in extended serial mode.

RXDESEL Bit (Asynchronous Start Bit Edge Detection Select)

Selects the detection method of the start bit for reception in asynchronous mode. When a break occurs, data receiving operation depends on the settings of this bit. Set this bit to 1 when reception should be stopped while a break occurs or when reception should be started without retaining the RXDn pin input at high level for the period of one data frame or longer after completion of the break.
Set this bit to 1 in extended serial mode. Set this bit to 0 in modes except of asynchronous mode and extended serial mode.

MOD[2:0] Bits (Communication Mode Select)

Selects the RSCI communication mode.

Table 36.26 Relationship between Communication Mode Selection Bits (MOD[2:0]), Other Operation Mode Setting Bits

Communication mode	Asynchronous								SMIF	Clock Synchronous		Simple SPI				Simple I²C	Manchester		Extended Serial
SCR3.MOD[2:0]	000b								001b	010b		011b				100b	101b		110b
SCR3.MP	0				1				—	—		—				—	0	1	—
SCR3.FM	0		1		0		1		—	0	1	0		1		—	—		—
SCR3.DEEN	0	1	0	1	0	1	0	1	—	—		—				—	—		—
SCR3.SSE	—								—	—		0	1	0	1	—	—		—

—: Prohibited setting

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.

FM Bit (FIFO Mode Select)

When the FM bit is set to 1, the TDR register/RDR register switches to FIFO configuration, and transmit FIFO (TDR register)/receive FIFO (RDR register) can be used for serial transmission/reception.

DEEN Bit (Driver Control Function Enable)

Select RS-485 Driver control function disable or enable.

CKE[1:0] Bits (Clock Enable)

These bits select the clock source and SCKn pin function.

In smart card interface mode, these bits control the clock output from the SCKn pin.

In GSM mode, clock output can be dynamically switched. For details, refer to section 36.7.8, Clock Output Control.

GM Bit (GSM Mode)

Setting this bit to 1 allows GSM mode operation.

In GSM mode, the SSR.TEND flag set timing is put forward to 11.0 etu (elementary time unit = 1-bit transfer time) from the start and the clock output control function is appended. For details, refer to section 36.7.6, Serial Data Transmission (Except in Block Transfer Mode) and section 36.7.8, Clock Output Control.

BLK Bit (Block Transfer Mode)

Setting this bit to 1 allows block transfer mode operation.

For details, refer to section 36.7.3, Block Transfer Mode.

36.2.9 Control Register 4 (SCR4)

Address(es): RSCI10.SCR4 000E 2018h, RSCI11.SCR4 000E 2098h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TTMG[3:0]				RTMG[3:0]				—	—	—	—	—	—	TTADJ	RTADJ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CMPD[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	CMPD[8:0]	Compare Match Data	(Valid only in asynchronous mode) Set the compared data when using data match detection function	R/W *1
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	RTADJ	Receive Data Sampling Timing Adjustment	(Valid in asynchronous mode using internal clock, extended serial mode using internal clock, clock synchronous mode operating as master, simple SPI mode operating as master) 0: Adjust sampling timing disable. 1: Adjust sampling timing enable.	R/W *1
b17	TTADJ	Transmit Signal Transition Timing Adjustment	(Valid only in asynchronous mode using internal clock) 0: Adjust transmit timing disable. 1: Adjust transmit timing enable.	R/W *1
b23 to b18	—	Reserved	These bits are read as 0. The write value should be 0.	R
b27 to b24	RTMG[3:0]	Receive Data Sampling Timing Select	In the case of asynchronous mode and extended serial mode b27 b24 1 1 1 1: Data are sampled 7 clocks earlier than default point. 1 1 1 0: Data are sampled 6 clocks earlier than default point. 1 1 0 1: Data are sampled 5 clocks earlier than default point. 1 1 0 0: Data are sampled 4 clocks earlier than default point. 1 0 1 1: Data are sampled 3 clocks earlier than default point. 1 0 1 0: Data are sampled 2 clocks earlier than default point. 1 0 0 1: Data are sampled 1 clocks earlier than default point. x 0 0 0: Data are sampled at default point. 0 0 0 1: Data are sampled 1 clock later than default point. 0 0 1 0: Data are sampled 2 clock later than default point. 0 0 1 1: Data are sampled 3 clock later than default point. 0 1 0 0: Data are sampled 4 clock later than default point. 0 1 0 1: Data are sampled 5 clock later than default point. 0 1 1 0: Data are sampled 6 clock later than default point. 0 1 1 1: Data are sampled 7 clock later than default point. In the case of clock synchronous mode and simple SPI mode b27 b24 0 0 0 0: 1 PCLKA delay 0 0 0 1: 2 PCLKA delay 0 0 1 0: 3 PCLKA delay 0 0 1 1: 4 PCLKA delay Settings other than above are prohibited.	R/W *1

Bit	Symbol	Bit Name	Description	R/W
b31 to b28	TTMG[3:0]	Transmit Signal Transition Timing Select	b31 b28 1 1 1 1: Delays the 1 to 0 transitions for 7 clocks. 1 1 1 0: Delays the 1 to 0 transitions for 6 clocks. 1 1 0 1: Delays the 1 to 0 transitions for 5 clocks. 1 1 0 0: Delays the 1 to 0 transitions for 4 clocks. 1 0 1 1: Delays the 1 to 0 transitions for 3 clocks. 1 0 1 0: Delays the 1 to 0 transitions for 2 clocks. 1 0 0 1: Delays the 1 to 0 transitions for 1 clocks. x 0 0 0: Does not change the waveform. 0 0 0 1: Delays the 0 to 1 transitions for 1 clock. 0 0 1 0: Delays the 0 to 1 transitions for 2 clock. 0 0 1 1: Delays the 0 to 1 transitions for 3 clock. 0 1 0 0: Delays the 0 to 1 transitions for 4 clock. 0 1 0 1: Delays the 0 to 1 transitions for 5 clock. 0 1 1 0: Delays the 0 to 1 transitions for 6 clock. 0 1 1 1: Delays the 0 to 1 transitions for 7 clock.	R/W *1

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

CMPD[8:0] Bits (Compare Match Data)

Set the comparison data for receive data, when data match detection function is enabled (SCR0.DCME bit = 1). SCR4.CMPD[8:0] bits should be written while SCR0.DCME bit is 0.

For the comparison data, it can select length from 3 types, they are CMPD[6:0] with 7bit length enable, CMPD[7:0] with 8bit, and CMPD[8:0] with 9bit length.

RTADJ Bit (Receive Data Sampling Timing Adjustment)

When this bit is 1, the receive sampling timing adjustment function is enabled. Control is different in asynchronous mode, extended serial mode, clock synchronous mode, and simple SPI mode.

In asynchronous mode using internal clock, refer to section 36.3.10, Receive Data Sampling Timing Adjustment for details. The operation when the extended serial mode internal clock is selected is the same as when the asynchronous clock internal clock is selected.

In clock synchronous mode as master, simple SPI mode operating as master, refer to section 36.10.7, Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with Internal Clock Used for details. Only the digital delay of the master mode receive sampling clock (MRCLK) can be controlled by this bit. MRCLK analog delay cannot be controlled.

TTADJ Bit (Transmit Signal Transition Timing Adjustment)

When this bit is 1, the transmit signal transition timing adjustment function is enabled. The transmit signal transition timing adjustment function can adjust the edge timing of the waveform output from the TXDn pin. Refer to section 36.3.11, Transmit Data Transition Timing Adjustment for details.

RTMG[3:0] Bits (Receive Data Sampling Timing Select)

When the RTADJ bit is 1, the receive sampling timing can be adjusted according to this bit setting value. The adjustment value in the synchronous mode and the extended serial mode is the base clock × RTMG[2:0] setting value.

TTMG[3:0] Bits (Transmit Signal Transition Timing Select)

The edge timing of the TXDn pin specified by the TTMG[3:0] bits is adjusted by the base clock × TTMG[2:0] setting value. Make sure that the TTMG[2:0] bit setting is less than the number of base clock cycles for 1-bit period.

36.2.10 I²C Mode Register (SIMR)

Address(es): RSCI10.SIMR 000E 2020h, RSCI11.SIMR 000E 20A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	IICSCLS[1:0]		IICSDAS[1:0]		—	IICSTPREQ	IICRSTAREQ	IICSTAREQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	IICACKT	—	—	—	IICCSC	IICINTM	—	—	—	IICDL[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	IICDL[4:0]	SDA Output Delay Select	(Cycles below are of the clock signal from the on-chip baud rate generator.) b4 b0 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
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	IICINTM	I²C Interrupt Mode Select	0: Use ACK/NACK interrupts. 1: Use reception and transmission interrupts	R/W *1
b9	IICCSC	Clock Synchronization	0: No synchronization with the clock signal 1: Synchronization with the clock signal	R/W *1
b12 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R
b13	IICACKT	ACK Transmission Data	0: ACK transmission 1: NACK transmission and reception of ACK/NACK	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	IICSTAREQ	Start Condition Generation	0: A start condition is not generated. 1: A start condition is generated.*2, *4, *5, *6	R/W
b17	IICRSTAREQ	Restart Condition Generation	0: A restart condition is not generated. 1: A restart condition is generated.*3, *4, *5, *6	R/W
b18	IICSTPREQ	Stop Condition Generation	0: A stop condition is not generated. 1: A stop condition is generated.*3, *4, *5, *6	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R
b21, b20	IICSDAS[1:0]	SDA Output Select	b21 b20 0 0: Serial data output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SSDAn pin. 1 1: Place the SSDAn pin in the high-impedance state.	R/W
b23, b22	IICSCLS[1:0]	SCL Output Select	b23 b22 0 0: Serial clock output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SSCLn pin. 1 1: Place the SSCLn pin in the high-impedance state.	R/W
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.
Note 2. In the bus free state, perform the start condition generation.
Note 3. In the bus busy state, perform restart or stop condition generation when the SSCLn pin after acknowledgment described in Figure 36.78 and Figure 36.79 is low level.
Note 4. Do not set more than one from among the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits to 1 at a given time.

Note 5. Execute the generation of a condition after the value of the IICSTIF flag is 0.
Note 6. Do not write 0 to this bit while it is 1. Generation of a condition is suspended by writing 0 to this bit while it is 1.

IICDL[4:0] Bits (SDA 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 generators the base. The signal obtained by frequency-dividing PCLKA by the divisor set in SCR2.CKS[1:0] bits is supplied as the clock signal from the on-chip baud rate generator.
Set these bits to 00000b unless operation is in simple I^2C mode. In simple I^2C mode, set the bits to a value in the range from 00001b to 11111b.

IICINTM Bit (I^2C Interrupt Mode Select)

This bit selects the sources of interrupt requests in simple I^2C mode.

IICCSC Bit (Clock Synchronization)

Set the IICCSC bit to 1 if the internally generated SCL signal is to be synchronized when the SSCLn pin has been placed at the low level in the case of a wait inserted by the other device, etc.
The internal SCL signal is not synchronized if the IICCSC bit is 0. The internal SCL signal is generated in accordance with the rate selected in the BRR[7:0] bits 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.

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.
If you want to generate the start condition after generating the stop condition, start the generation of the start condition with a half cycle period of the bit rate from the stop condition generation interrupt request output.
[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

IICSDAS[1:0] Bits (SDA Output Select)

These bits control output from the SSDAn pin.

IICSCLS[1:0] Bits (SCL Output Select)

These bits control output from the SSCLn pin.

36.2.11 FIFO Control Register (FCR)

Address(es): RSCI10.FCR 000E 2024h, RSCI11.FCR 000E 20A4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	RSTRG[4:0]					RFRST	—	—	RTRG[4:0]				
Value after reset:	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TFRST	—	—	TTRG[4:0]					—	—	—	—	—	—	—	DRES
Value after 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	DRES	Receive Data Ready Interrupt Select	(Valid in asynchronous mode) This bit select the interrupt request for a reception data ready detection. 0: Reception data full interrupt (RXI) 1: Receive error interrupt (ERI)	R/W *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12 to b8	TTRG[4:0]	Transmit FIFO Threshold Setting	(Valid in asynchronous mode (including multi-processor mode), clock synchronous mode, simple SPI mode) b12 b8 0 0 0 0 0: Trigger number 0 : : 1 1 1 1 1: Trigger number 31	R/W *1
b14, b13	—	Reserved	These bits are read as 0. The write value should be 0.	R
b15	TFRST	Transmit FIFO Reset	This bit enables only when SCR3.FM bit is 1. The read value is always 0. 0: It is invalid. It does not affect the operation. 1: The number of data stored in transmit FIFO (TDR register) are made 0 The read value is always 0.	W*1
b20 to b16	RTRG[4:0]	Receive FIFO Threshold Setting	(Valid in asynchronous mode (including multi-processor mode), clock synchronous mode, simple SPI mode) b20 b16 0 0 0 0 0: Trigger number 0 : : 1 1 1 1 1: Trigger number 31	R/W *1
b22, b21	—	Reserved	These bits are read as 0. The write value should be 0.	R
b23	RFRST	Receive FIFO Reset	This bit enables only when SCR3.FM bit is 1. The read value is always 0. 0: It is invalid. It does not affect the operation. 1: The number of data stored in receive FIFO (RDR register) are made 0 The read value is always 0.	W*1
b28 to b24	RSTRG[4:0]	RTS# Output Threshold Setting	(Valid in asynchronous mode (including multi-processor mode), clock synchronous mode) This bit enables only when SCR3.FM bit = 1, SCR1.CTSE bit = 0, and SCR0.SSE bit = 0. b28 b24 0 0 0 0 0: Trigger number 0 : : 1 1 1 1 1: Trigger number 31	R/W *1
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

DRES Bit (Receive Data Ready Interrupt Select)

Select whether the detection of receive data ready (RFSR.DR flag = 1) is the cause of RXI interrupt request or the cause of ERI interrupt request.

TTRG[4:0] Bits (Transmit FIFO Threshold Setting)

The TDFE flag is set to 1 when the quantity of transmit data in the transmit FIFO (TDR register) is equal to or less than the specified transmit triggering number. If SCR0.TIE bit = 1, TXI interrupt request is occurred.

TFRST Bit (Transmit FIFO Reset)

When the TFRST bit is set to 1, the number of the transmission data stored in transmit FIFO (TDR register) is made 0.

RTRG[4:0] Bits (Receive FIFO Threshold Setting)

The SSR.RDRF flag is set to 1 when the quantity of receive data in the receive FIFO (RDR register) is equal to or greater than the specified receive triggering number. If SCR0.RIE bit = 1, RXI interrupt request is occurred. When FCR.RTRG[4:0] bits are set to 0, RDRF flag is set if the quantity of data in receive FIFO is greater than or equal to 1.

RFRST Bit (Receive FIFO Reset)

When the RFRST bit is set to 1, the number of the reception data stored in receive FIFO (RDR register) is made 0.

RSTRG[4:0] Bits (RTS# Output Threshold Setting)

When the quantity of receive data stored in the receive FIFO (RDR register) is equal to or greater than this number, the RTSn# signal is in the High state. When FCR.RSTRG[4:0] bits are set to 0, RTSn# is in the high state if the quantity of data in receive FIFO is greater than or equal to 1.

36.2.12 Manchester Mode Control Register (MMCR)

Address(es): RSCI10.MMCR 000E 202Ch, RSCI11.MMCR 000E 20ACh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	SBERIE	SYERIE	PFERIE	—	—	RPPAT[1:0]		RPLEN[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
	—	—	TPPAT[1:0]		TPLEN[3:0]				—	SBLEN	SYNCE	SBPTN	—	SADJE	ENCS	DECS
Value after 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	DECS	Decoding Convention Select	Sets the polarity of the received manchester code 0: Low to high transition is decoded to a logic 0 and high to low transition is decoded to a logic 1. 1: high to low transition is decoded to a logic 0 and low to high transition is decoded to a logic 1.	R/W *1
b1	ENCS	Encoding Convention Select	Sets the polarity of the transmit manchester code 0: Logic 0 is encoded to a low to high transition and logic 1 is encoded to a high to low transition. 1: Logic 0 is encoded to a high to low transition and logic 1 is encoded to a low to high transition.	R/W *1
b2	SADJE	Receive Timing Self Adjustment Enable	Sets the receive retiming function 0: Disables the receive retiming function 1: Enables the receive retiming function	R/W *1
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b4	SBPTN	Start Bit Pattern Select	Sets the Sync type of the start bit(s) in the manchester code When the start bit area consists of one bit. (SBLEN bit = 0) • When transmitting 0: The start bit is added as a low to high transition. 1: The start bit is added as a high to low transition. • When receiving 0: Only when the start bit is a low to high transition, the data is received. The other cases are judged as an error. 1: Only when the start bit is a high to low transition, the data is received. The other cases are judged as an error. When the start bit area consists of three bits. (SBLEN bit = 1) • When transmitting 0: The start bits are added as a low to high transition. (DATA Sync) 1: The start bits are coded as a high to low transition. (COMMAND Sync) • When receiving When the start bit area consists of three bits, data is received regardless of the value of this bit.	R/W *1
b5	SYNCE	Sync Enable	0: The start bit pattern is set with the SBPTN bit. 1: The start bit pattern is set with the SYNC bit.	R/W *1
b6	SBLEN	Start Bit Length Select	0: The start bit area consists of one bit. 1: The start bit area consists of three bits. (COMMAND Sync or DATA Sync)	R/W *1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R
b11 to b8	TPLEN[3:0]	Transmit Preface Length Setting	Set the preface length of the transmit data in manchester mode 0: Disables the transmit preface generation 1 to 15: Transmit preface length (bit length)	R/W *1

Bit	Symbol	Bit Name	Description	R/W
b13, b12	TPPAT[1:0]	Transmit Preface Pattern Select	Set the preface pattern of the transmit data b13 b12 0 0: ALL ZERO 0 1: ZERO ONE 1 0: ONE ZERO 1 1: ALL ONE	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R
b19 to b16	RPLEN[3:0]	Receive Preface Length Setting	Set the preface length in received frames when manchester mode is enabled 0: Disables the receive preface generation 1 to 15: Receive preface length (bit length)	R/W *1
b21, b20	RPPAT[1:0]	Receive Preface Pattern Select	Set the preface pattern of received frames b21 b20 0 0: ALL ZERO 0 1: ZERO ONE 1 0: ONE ZERO 1 1: ALL ONE	R/W *1
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	PFERIE	Preface Error Interrupt Enable	Specifies whether to handle a preface error as an interrupt source 0: Does not handle a preface error as an interrupt source 1: Handles a preface error as an interrupt source	R/W *1
b25	SYERIE	Sync Error Interrupt Enable	Specifies whether to handle a receive Sync error as an interrupt source 0: Does not handle a receive Sync error as an interrupt source 1: Handles a receive Sync error as an interrupt source	R/W *1
b26	SBERIE	Start Bit Error Interrupt Enable	Specifies whether to handle a start bit error as an interrupt source 0: Does not handle a start bit error as an interrupt source 1: Handles a start bit error as an interrupt source	R/W *1
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

DECS Bit (Decoding Convention Select)

This bit sets the polarity of the received manchester code. For details on the data reception, see section 36.5.7, Serial Data Reception in Manchester Mode.

ENCS Bit (Encoding Convention Select)

This bit sets the polarity of the transmit manchester code. For details on the data transmission, see section 36.5.6, Serial Data Transmission in Manchester Mode.

SADJE Bit (Receive Timing Self Adjustment Enable)

This bit sets the receive retiming function in manchester mode.

For information on the receive retiming function, see section 36.5.9, Receive Retiming.

SBPTN Bit (Start Bit Pattern Select)

This bit is valid when the SYNCE bit of this register is set to 0.

The Sync type can be set by combining this bit and the SBLEN bit.

For the start bit area determined by the combination of this bit and the SBLEN bit, see Figure 36.36 and Figure 36.37.

SYNCE Bit (Sync Enable)

This bit is valid when the SBLEN bit of this register is set to 1. This bit determines the destination to be referred to for setting the Sync type of the start bit area added to manchester frames.
When this bit is set to 0, the SBPTN bit of this register is referred to.
When this bit is set to 1, the SYNC bit in the TDR register is referred to.

SBLEN Bit (Start Bit Length Select)

This bit sets the start bit area in manchester frames.
When this bit is set to 1, the start bit area added to each frame consists of three bits, and the SYNCE and SBPTN bits in this register are valid.
When this bit is set to 0, the start bit area added to each frame consists of one bit.

TPLEN[3:0] Bits (Transmit Preface Length Setting)

These bits set the preface bit length of the transmit data in manchester mode.
The settable range is 0h to Fh (0 to 15). 0h disables the transmit preface, which is not added.

TPPAT[1:0] Bits (Transmit Preface Pattern Select)

These bits set one of the four preface patterns in manchester mode. For the transmit data when the TPPAT[1:0] bits are set, see Figure 36.35.
When these bits are set to 00b, the preface area is set to all zeros.
When these bits are set to 01b, the preface area is set to the zero-one-zero-one pattern.
When these bits are set to 10b, the preface area is set to the one-zero-one-zero pattern.
When these bits are set to 11b, the preface area is set to all ones.

RPLEN[3:0] Bits (Receive Preface Length Setting)

These bits set the preface bit length of the received frames in manchester mode.
The settable range is 0h to Fh (0 to 15). 0h disables the receive preface, which is not added. When 1h to Fh is set, the set value is handled as the receive preface bit length.

RPPAT[1:0] Bits (Receive Preface Pattern Select)

These bits set one of the four preface patterns in manchester mode. For the transmit and receive data when the TPPAT[1:0] bits are set, see Figure 36.35.
When these bits are set to 00b, the preface area is handled as all zeros.
When these bits are set to 01b, the preface area is handled as the zero-one-zero-one pattern.
When these bits are set to 10b, the preface area is handled as the one-zero-one-zero pattern.
When these bits are set to 11b, the preface area is handled as all ones.

PFERIE Bit (Preface Error Interrupt Enable)

This bit specifies whether to handle a preface error as an interrupt source.
When it is set to 0, a preface error is not handled as an interrupt source. When it is set to 1, a preface error is handled as an interrupt source.

SYERIE Bit (Sync Error Interrupt Enable)

This bit specifies whether to handle a receive Sync error as an interrupt source.
When it is set to 0, a receive Sync error is not handled as an interrupt source. When it is set to 1, a receive Sync error is handled as an interrupt source.

SBERIE Bit (Start Bit Error Interrupt Enable)

This bit specifies whether to handle a start bit error as an interrupt source.

When it is set to 0, a start bit error is not handled as an interrupt source. When it is set to 1, a start bit error is handled as an interrupt source.

36.2.13 DE Signal Control Register (DECR)

Address(es): RSCI10.DECR 000E 2030h, RSCI11.DECR 000E 20B0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	DEHLD[4:0]				
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
	—	—	—	DESU[4:0]					—	—	—	—	—	—	—	DELVL
Value after 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	DELVL	DE Signal Active Level Select	(Valid only in asynchronous mode) 0: The DE signal is active high. 1: The DE signal is active low.	R/W *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b12 to b8	DESU[4:0]	DE Signal Setup Time Setting	(Valid only in asynchronous mode) Set the DE signal setup time in the number of the base clock cycles. The bit setting is enabled when the SCR3.DEEN bit is 1. Do not set 00000b.	R/W *1
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20 to b16	DEHLD[4:0]	DE Signal Hold Time Setting	(Valid only in asynchronous mode) Set the DE signal hold time in the number of the base clock cycles. The bit setting is enabled when the SCR3.DEEN bit is 1. Do not set 00000b.	R/W *1
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

DELVL Bit (DE Signal Active Level Select)

Select the active level of the DE (driver enable) signal.

DESU[4:0] Bits (DE Signal Setup Time Setting)

Set the DE signal setup time (time from the assertion of the DE signal to the start of transmission of the start bit). It is expressed in the number of the base clock cycles (1/8 or 1/16 bit period). The actual transmission of the start bit starts after the setup time and transmission wait time have elapsed.

DEHLD[4:0] Bits (DE Signal Hold Time Setting)

Set the DE signal hold time (time from the completion of transmission of the stop bit of the last transmission message to negation of the DE signal). It is expressed in the number of the base clock cycles (1/8 or 1/16 bit period).
If the transmission data is written during the hold time, transmit starting operation is different depends on the writing timing (following two cases: the transmission of the start bit starts after the transmission wait time has elapsed without negating the DE signal, or it starts after the DE signal is negated and asserted again and then the setup time and transmission wait time have elapsed.).

36.2.14 Extended Serial Mode Control Register 0 (XCR0)

Address(es): RSCI10.XCR0 000E 2034h, RSCI11.XCR0 000E 20B4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	BCCS[1:0]		—	AEDIE	COFIE	BFDIE	—	—	BCDIE	BFOIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PIBS[2:0]			PIBE	CF1DS[1:0]		CF0RE	BFE	—	—	—	—	—	—	TCSS[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	TCSS[1:0]	Timer Count Clock Source Select	(Valid in extended serial mode) Select the clock source of the timer in the extended serial module. b1 b0 0 0: PCLKA 0 1: PCLKA/4 1 0: PCLKA/16 1 1: PCLKA/64	R/W *1, *2
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	BFE	Break Field Detection Enable	Set the presence or absence of Break Field of Start Frame. 0: No Break Field 1: With Break Field	R/W *1, *4
b9	CF0RE	Control Field 0 Reception Enable	Set the presence or absence of Control Field 0 of Start Frame 0: No Control Field 0 1: With Control Field 0	R/W *1, *4
b11, b10	CF1DS[1:0]	Control Field 1 Compare Data Select	Select the compare data for Control Field 1 b11 b10 0 0: Select XCR1.PCF1D[7:0] bits as the compare data 0 1: Select XCR1.SCF1D[7:0] bits as the compare data 1 0: Select both XCR1.PCF1D[7:0] bits and XCR1.SCF1D[7:0] bits as the compare data 1 1: Prohibition	R/W *1, *4
b12	PIBE	Priority Interrupt Bit Enable	0: Priority interrupt bit disable 1: Priority interrupt bit enable	R/W *1, *4
b15 to b13	PIBS[2:0]	Priority Interrupt Bit Select	Specify one of bits 0 to 7 of Control Field 1 as the priority interrupt bit. b15 b13 0 0 0: Bit 0 of Control Field 1 0 0 1: Bit 1 of Control Field 1 0 1 0: Bit 2 of Control Field 1 0 1 1: Bit 3 of Control Field 1 1 0 0: Bit 4 of Control Field 1 1 0 1: Bit 5 of Control Field 1 1 1 0: Bit 6 of Control Field 1 1 1 1: Bit 7 of Control Field 1	R/W *1, *4
b16	BFOIE	Break Field Low Width Output Complete Interrupt Enable	Select whether to include Break Field transmission completion as a TXI interrupt factor. 0: Break Field transmission completion is not included in TXI interrupt factor 1: Break Field transmission completion is included in TXI interrupt factor	R/W *1
b17	BCDIE	Bus Collision Detected Interrupt Enable	Select whether to output an ERI interrupt when a bus collision is detected. 0: Bus conflict detection is not included in ERI interrupt factor 1: Bus conflict detection is included in ERI interrupt factor	R/W *1
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R

Bit	Symbol	Bit Name	Description	R/W
b20	BFDIE	Break Field Low Width Detected Interrupt Enable	Select whether to output a BFD interrupt when a Break Field is detected. 0: Break Field detection interrupt disable 1: Break Field detection interrupt enable	R/W *1
b21	COFIE	Count Overflow Interrupt Enable	Select whether to include counter overflow as an ERI interrupt factor. 0: Counter overflow is not included in ERI interrupt factor 1: Counter overflow is included in ERI interrupt factor	R/W *1
b22	AEDIE	Effective Edge Detected Interrupt Enable	Select whether to output an AED interrupt when a valid edge is detected. 0: Active edge detection interrupt disable 1: Active edge detection interrupt enable	R/W *1
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R
b25, b24	BCCS[1:0]	Bus Collision Detection Clock Select	Select the sampling clock for the bus conflict detection circuit. When SCR2.ABCS bit = 1, setting BCCS[1:0] bits = 1x is prohibited. b25 b24 0 0: Base clock*3 0 1: Base clock/2 1 0: Base clock/4 1 1: Prohibition	R/W *1
b31 to b26	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.
Note 2. Rewrite the TCSS[1:0] bits only when the timer is stopped (XCR1.TCST bit = 0, XCR1.SDST bit = 0, and XCR1.BRME bit = 0).
Note 3. Base clock: 1/16 period of 1 bit period when SCR2.ABCS bit = 0, 1/8 period of 1 bit period when SCR2.ABCS bit = 1.
Note 4. This bit is a setting bit required for Start Frame reception operation. Rewrite this bit when Start Frame reception or transmission is not in progress (XCR1.SDST bit = 0 and XCR1.TCST bit = 0).

TCSS[1:0] Bits (Timer Count Clock Source Select)

Select clock source of timer in extended serial module.

BFE Bit (Break Field Detection Enable)

Set the presence or absence of Break Field of Start Frame.

CF0RE Bit (Control Field 0 Reception Enable)

Set the presence or absence of Control Field 0 of Start Frame.

CF1DS[1:0] Bits (Control Field 1 Compare Data Select)

Select the compare data for Control Field 1.

PIBE Bit (Priority Interrupt Bit Enable)

Select whether to enable priority interrupt bit comparison of Control Field 1. When this bit is 1, regardless of the XCR1.CF1CE[7:0] bits setting value, the bit specified in PIBS[2:0] is compared with the primary comparison data for Control Field 1 (XCR1.PCF1D[7:0] bits).

PIBS[2:0] Bits (Priority Interrupt Bit Select)

Specify bit N (N = 0 to 7) of Control Field 1 as the priority interrupt bit.

BFOIE Bit (Break Field Low Width Output Complete Interrupt Enable)

Select whether to include Break Field transmission completion as a TXI interrupt factor. Set SCR0.TIE bit = 1 and SCR3.MOD[2:0] bits = 110b, to output TXI upon completion of Break Field transmission.

BCDIE Bit (Bus Collision Detected Interrupt Enable)

Select whether to output an ERI interrupt when a bus collision is detected. In extended serial mode (SCR3.MOD[2:0] bits = 110b), ERI output control is performed with this bit. When SCR3.MOD[2:0] bits = 110b and BCDIE bit = 1, an ERI interrupt is issued when a bus collision is detected even if SCR0.RIE bit = 0.

COFIE Bit (Count Overflow Interrupt Enable)

Select whether to include counter overflow as an ERI interrupt factor. Set SCR0.RIE bit = 1 and SCR3.MOD[2:0] bits = 110b are required to output ERI upon counter overflow.

AEDIE Bit (Effective Edge Detected Interrupt Enable)

Select whether to output an AED interrupt when a valid edge is detected. To output AED with valid edge detection, XCR1.BRME bit = 1 and SCR3.MOD[2:0] bits = 110b must be set.

BCCS[1:0] Bits (Bus Collision Detection Clock Select)

Select the sampling clock for the bus conflict detection circuit.

36.2.15 Extended Serial Mode Control Register 1 (XCR1)

Address(es): RSCI10.XCR1 000E 2038h, RSCI11.XCR1 000E 20B8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CF1CE[7:0]								SCF1D[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PCF1D[7:0]								—	—	BRME	SDST	—	—	—	TCST
Value after 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	TCST	Break Field Low Width Output Timer Count Start	0: Break Field transmission timer count stopped 1: Break Field transmission timer count start Do not set this bit and SDST bit to 1 at the same time.	R/W *1
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	SDST	Start Frame Detection Start	0: Start Frame/Break Field detection disabled 1: Start Frame/Break Field detection enabled Do not set this bit and TCST bit to 1 at the same time.	R/W *1
b5	BRME	Bit Rate Measurement Enable	0: Bit rate measurement disabled 1: Bit rate measurement enabled Set this bit to 1 simultaneously with the SDST bit. When this bit is set to 0, it can be set to 0 at any timing.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R
b15 to b8	PCF1D[7:0]	Primary Control Field 1 Compare Data	The priority compare data for Control Field 1	R/W *1
b23 to b16	SCF1D[7:0]	Secondary Control Field 1 Compare Data	The secondary compare data for Control Field 1	R/W *1
b31 to b24	CF1CE[7:0]	Control Field 1 Compare Enable	Select whether to compare bit N of Control Field 1. (N = 0 to 7) 0: Control Field 1 bit N compare disabled 1: Control Field 1 bit N compare enabled	R/W *1

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

TCST Bit (Break Field Low Width Output Timer Count Start)

[Clearing condition]

- When 0 is written to TCST bit. Break Field transmission timer count is stopped and TXDn pin output becomes idle level.
- When Break Field transmission for the period set in XCR2.BFLW[15:0] bits is completed.

[Setting condition]

- When 1 is written to TCST bit. Start Break Field transmission from TXDn pin. Holds 1 during Break Field transmission.

SDST Bit (Start Frame Detection Start)

When 1 is written to this bit, Start Frame detection starts. When XCR0.BFE bit = 1 is set, Break Field can be detected during Start Frame is detected and after Start frame is detected. When XCR0.BFE bit = 0 is set, Break Field is not detected.

When 0 is written to this bit, Start Frame detection and Break Field detection are stopped. However, if XSR0.RXDSF flag = 0 at the time of stop, it is not possible to stop data reception with this bit. Write 0 to SCR0.RE bit to stop the

reception operation or perform reception completion processing (clearing the SSR.RDRF flag or reading the RDR register) after reception is completed.

BRME Bit (Bit Rate Measurement Enable)

Set this bit to 1 simultaneously with the SDST bit. When this bit is set to 1, the valid edge interval of Control Field 0 and Control Field 1 data is measured.

PCF1D[7:0] Bits (Primary Control Field 1 Compare Data)

Set the priority compare data for Control Field 1.

SCF1D[7:0] Bits (Secondary Control Field 1 Compare Data)

Set the secondary compare data for Control Field 1.

CF1CE[7:0] Bits (Control Field 1 Compare Enable)

Select whether to compare bit N of Control Field 1. (N = 0 to 7)
When all of these bits are set to 0 (CF1CE[7:0] bits = 00h), it is always judged that Control Field 1 matches when reception is completed, and XSR0.CF1MF flag is set. This bit is a comparison enable with PCF1D[7:0] bits or SCF1D[7:0] bits, and it is not a priority interrupt bit comparison enable.

36.2.16 Extended Serial Mode Control Register 2 (XCR2)

Address(es): RSCI10.XCR2 000E 203Ch, RSCI11.XCR2 000E 20BCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	BFLW[15:0]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CF0CE[7:0]								CF0D[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	CF0D[7:0]	Control Field Compare Data	The compare data for Control Field 0	R/W *1
b15 to b8	CF0CE[7:0]	Control Field Compare Enable	Select whether to compare bit N of Control Field 0. (N = 0 to 7) 0: Control Field 0 bit N compare disabled 1: Control Field 0 bit N compare enabled	R/W *1
b31 to b16	BFLW[15:0]	Break Field Low Width Setting	This bit sets the Break Field length. The break field length is (BFLW[15:0] setting value + 1) × clock of the timer. The upper limit for setting this register is FFFEh. (Setting prohibited for FFFFh)	R/W *1

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

CF0D[7:0] Bits (Control Field Compare Data)

The compare data for Control Field 0.

CF0CE[7:0] Bits (Control Field Compare Enable)

Select whether to compare bit N of Control Field 0. (N = 0 to 7)

When all of these bits are set to 0 (CF0CE[7:0] bits = 00h), it is always judged that Control Field 0 matches when reception is completed, and the XSR0.CF0MF flag is set.

BFLW[15:0] Bits (Break Field Low Width Setting)

The BFLW[15:0] bits are 16-bit Break Field length setting bits and the initial value is FFFEh.

Set the break field length to 1 frame or more. The LIN standard stipulates that the Break Field length is 13 bits or more.

When sending the Break Field, writing 1 to the TCST bit leads RSCI to output the Break Field on the TXDn pin. At the same time, the timer starts counting with the timer count clock source selected by the XCR0.TCSS[1:0] bits. When the count value matches the value set in this register, up-counting is stopped and Break Field transmission from the TXDn pin is also stopped.

When detecting the Break Field, writing 1 to the SDST bit leads Start Frame detection to be enabled. RSCI starts counting from the negative edge of the RXDn signal. The timer count clock source is selected by the XCR0.TCSS[1:0] bits. When the count value matches the value set in this register, it is determined that a break field has been detected. Up-counting continues until the next valid edge or counter overflow.

36.2.17 Status Register (SSR)

Address(es): RSCI10.SSR 000E 2048h, RSCI11.SSR 000E 20C8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RDRF	TEND	TDRE	AFER	APER	MFF	—	ORER	—	—	—	—	—	DFER	DPER	DCMF
Value after reset:	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	This bit is read as 0. The write value should be 0.	R
b4	ERS	Error Signal Status Flag	(Valid only in Smart card interface mode) 0: Error signal Low not responded 1: Error signal Low responded	R
b14 to b5	—	Reserved	These bits are read as 0.	R
b15	RXDMON	RXD Line Monitoring Flag	The state of the RXDn pin is shown. When RINV is 0, 0: RXDn pin is the Low level. 1: RXDn pin is the High level. When RINV is 1, 0: RXDn pin is the High level. 1: RXDn pin is the Low level.	R
b16	DCMF	Data Match Flag	(Valid only in asynchronous mode) 0: No matched 1: Matched	R
b17	DPER	Matched Data Parity Error Flag	(Valid only in asynchronous mode) 0: No parity error occurred at data match detection 1: A parity error has occurred at data match detection	R
b18	DFER	Matched Data Framing Error Flag	(Valid only in asynchronous mode) 0: No framing error occurred at data match detection 1: A framing error has occurred at data match detection	R
b23 to b19	—	Reserved	These bits are read as 0.	R
b24	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error has occurred	R
b25	—	Reserved	This bit is read as 0.	R
b26	MFF	Mode Fault Flag	(Valid only in simple SPI mode.) 0: No mode fault error 1: Mode fault error	R
b27	APER	Aggregate Parity Error Flag	[Non-FIFO selected (SCR3.FM bit = 0)] 0: No parity error occurred 1: A parity error has occurred [FIFO selected (SCR3.FM bit = 1)] 0: No parity error in all received data in receive FIFO 1: One or more parity errors occurred in received data in receive FIFO	R
b28	AFER	Aggregate Framing Error Flag	[Non-FIFO selected (SCR3.FM bit = 0)] 0: No framing error occurred 1: A framing error has occurred [FIFO selected (SCR3.FM bit = 1)] 0: No framing error in all received data in receive FIFO 1: One or more framing errors occurred in received data in receive FIFO	R

Bit	Symbol	Bit Name	Description	R/W
b29	TDRE	Transmit Data Empty Flag	[Non-FIFO selected (SCR3.FM bit = 0)] 0: Transmit data is in TDR register 1: No transmit data is in TDR register [FIFO selected (SCR3.FM bit = 1)] 0: The quantity of transmit data written in transmit FIFO exceeds the specified transmit triggering number. 1: The quantity of transmit data written in transmit FIFO is equal to or less than the specified transmit triggering number.	R
b30	TEND	Transmit End Flag	0: A character is being transmitted or standing by for transmission. 1: Character transfer has been completed, or sending Break Field.	R
b31	RDRF	Receive Data Full Flag	[Non-FIFO selected (SCR3.FM bit = 0)] 0: No received data is in RDR register 1: Received data is in RDR register [FIFO selected (SCR3.FM bit = 1)] 0: The quantity of receive data written in receive FIFO falls below the specified receive triggering number. 1: The quantity of receive data written in receive FIFO is equal to or greater than the specified receive triggering number.	R

ERS Flag (Error Signal Status Flag)

[Setting condition]

- When an error signal Low is sampled.

[Clearing condition]

- When write 1 to SSCR.ERSC bit.

DCMF Flag (Data Match Flag)

Indicates that RSCI detects the match to the comparison data (SCR4.CMPD[8:0] bits) with receive data.
Clearing the SCR0.RE bit to 0 does not affect the DCMF flag, which retains its previous state.
[Setting condition]

- Matched to the comparison data (SCR4.CMPD[8:0] bits) with receive data, while SCR0.DCME bit = 1.

[Clearing condition]

- When write 1 to SSCR.DCMFC bit.

DPER Flag (Matched Data Parity Error Flag)

It indicates that a parity error occurred at data match detection.
Clearing the SCR0.RE bit to 0 does not affect the DPER flag, which retains its previous state.
[Setting condition]

- When a parity error was detected by the frame in which an data match was detected.

[Clearing condition]

- When write 1 to SSCR.DPERC bit.

Refer to section 15.5.2, Level Detection to exit from the interrupt handling routine after setting the DPER flag to 0.

DFER Flag (Matched Data Framing Error Flag)

It indicates that a framing error occurred at data match detection.
Clearing the SCR0.RE bit to 0 does not affect the DFER flag, which retains its previous state.
[Setting condition]

- When a stop bit of the frame in which an data match was detected is 0.
  When it is a 2-stop mode, the 1st bit of stop bit judges only whether it's 1 and doesn't check the 2nd bit of stop bit.

[Clearing condition]

- When write 1 to SSCR.DFERC bit.

Refer to section 15.5.2, Level Detection to exit from the interrupt handling routine after setting the DFER flag to 0.

ORER Flag (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.
Clearing the SCR0.RE bit to 0 does not affect the ORER flag, which retains its previous state. In simple I^2C mode, this bit is not use.
[Setting condition with non-FIFO mode (SCR3.FM bit = 0)]

- When the next data is received before reading out RDR register with no error reception data stored in RDR register.

In RDR register, 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, reception data isn't forwarded to RDR register. Note that, in clock synchronous mode and simple SPI mode, serial reception will be stop.
[Setting condition with FIFO mode (SCR3.FM bit = 1)]

- When the next serial reception is completed while the receive FIFO is full of 32 receive data.

[Clearing condition]

- When write 1 to SSCR.ORERC bit.

Refer to section 15.5.2, Level Detection to exit from the interrupt handling routine after setting the ORER flag to 0.

MFF Flag (Mode Fault Flag)

This bit indicates mode fault errors. In a multi-master configuration, determine the mode fault error occurrence by reading the MFF flag.
[Setting condition]

- Input on the SSn# pin being at the low level during master operation (SCR3.CKE[1:0] bits = 00b or 01b) in simple SPI mode.

[Clearing condition]

- When write 1 to SSCR.MFFC bit.

APER Flag (Aggregate Parity Error Flag)

Indicates that a parity error has occurred during reception and the reception ends abnormally.
Clearing the SCR0.RE bit to 0 does not affect the APER flag, which retains its previous state.
In clock synchronous mode, simple SPI mode and simple I^2C mode, this bit not used.
[Setting condition]

- In non-FIFO mode, when a parity error is detected during reception.
- In FIFO mode, when one or more parity error is detected for data in receive FIFO.

In non-FIFO mode, the received data when the parity error occurs is transferred to RDR register, but no RXI interrupt request occurs. Note that when the APER flag is being set to 1, the subsequent receive data is not transferred to RDR register.
[Clearing condition]

- When write 1 to SSCR.APERC bit.

Refer to section 15.5.2, Level Detection to exit from the interrupt handling routine after setting the APER flag to 0.

AFER Flag (Aggregate Framing Error Flag)

Indicates that a framing error has occurred during reception and the reception ends abnormally. Clearing the SCR0.RE bit to 0 does not affect the AFER flag, which retains its previous state.
In clock synchronous mode, simple SPI mode and simple I^2C mode, this bit not used.
[Setting condition]

- In non-FIFO mode, when 0 is sampled as the stop bit during reception.
- In FIFO mode, when one or more framing error is detected for data in receive FIFO.
- In manchester mode, when both sampling results (1/4 and 3/4 sampling points) are not 1 for 1 stop bit.

In 2 stop bit mode, only the first stop bit is checked whether it is 1 but the second stop bit is not checked.
In non-FIFO mode, the received data when the framing error occurs is transferred to RDR register, but no RXI interrupt request occurs. In addition, when the AFER flag is being set to 1, the subsequent receive data is not transferred to RDR register.
In extended serial mode, even if a condition that changes to 1 occurs when XCR1.SDST bit = 1, the AFER flag set timing is delayed up to the Break Field judgment timing at the longest, since it may be a Break Field. If an edge is detected on the RXD signal before the Break Field judgment timing, AFER is detected. If no edge is detected in the RXD signal before the Break Field judgment timing, Break Field is detected.
[Clearing condition]

- When write 1 to SSCR.AFERC bit.

Refer to section 15.5.2, Level Detection to exit from the interrupt handling routine after setting the AFER flag to 0.

TDRE Flag (Transmit Data Empty Flag)

(1) Non-FIFO selected (SCR3.FM bit = 0)
Indicates the presence of transmit data in the TDR register.
The condition of SCR0.TE bit = 0 has priority over the condition of 0.
If other conditions that become 1 and conditions that become 0 are satisfied at the same time, the TDRE flag is set to 0.
[Setting condition]

- When SCR0.TE bit is 0.
- When data is transmitted from the TDR register to the TSR register.

[Clearing condition]

- When write 1 to SSCR.TDREC bit.
- When the transmission data is written to the TDR register during SCR0.TE bit = 1.

(2) FIFO selected (SCR3.FM bit = 1)
Indicates that data has been transferred from the transmit FIFO (TDR register) into the transmit shift register (TSR), the quantity of data in transmit FIFO has fallen below the specified transmit triggering number.
When the condition which becomes 1 and the condition which becomes 0 were formed at the same time, TDRE flag will be 0. After that, when the number of data stored in transmit FIFO is judged, and if that is same or greater than TTRG value, TDRE flag is set to 1 after 1 PCLKA cycle.
[Setting condition]

- When the quantity of transmit data written in transmit FIFO is equal to or less than the specified transmit triggering number*1.

[Clearing condition]

- When write 1 to SSCR.TDREC bit.
- When the transmission data is written to transmit FIFO by the DTC or DMAC (the last block transfer when block transfer).

Note 1. Because the transmit FIFO is a 32-stage FIFO register, the maximum number of data that can be written to the TDR register when the TDRE flag is 1 is “32 – TFSR.T[5:0]”. All other data written to the TDR register above that value is ignored.

TEND Flag (Transmit End Flag)

(1) Non-FIFO selected (SCR3.FM bit = 0), and not smart card interface mode (SCR3.MOD[2:0] bits ≠ 001b)

Indicates completion of transmission.

[Setting condition]

- When SCR0.TE bit is 0.
- When the SCR0.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When the TDR register is not updated at the time of transmission of the tail-end bit of a character being transmitted.
- When the TDR register is not updated at the end of DE signal hold time with DE control function enable (SCR3.DEEN bit = 1).
- When Break Field is sending.

[Clearing condition]

- When the transmission data was written to the TDR register during SCR0.TE bit = 1.
- When write 1 to SSCR.TDREC bit during SCR0.TE bit = 1.

Refer to section 15.5.2, Level Detection to exit from the interrupt handling routine after setting the TEND flag to 0.

(2) Non-FIFO selected (SCR3.FM bit = 0), and smart card interface mode (SCR3.MOD[2:0] bits = 001b)

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 condition]

- When SCR0.TE bit is 0.
- When the SCR0.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When a specified period has elapsed after the latest transmission of 1 byte, the ERS flag is 0, and the TDR register is not updated.
  The set timing is determined by register settings as listed below.
  When GM bit = 0 and BLK bit = 0, 12.5 etu after the start of transmission
  When GM bit = 0 and BLK bit = 1, 11.5 etu after the start of transmission
  When GM bit = 1 and BLK bit = 0, 11.0 etu after the start of transmission
  When GM bit = 1 and BLK bit = 1, 11.0 etu after the start of transmission

[Clearing condition]

- When the transmission data was written to the TDR register during SCR0.TE bit = 1.
- When write 1 to SSCR.TDREC bit during SCR0.TE bit = 1.

(3) FIFO selected (SCR3.FM bit = 1)

Indicates that the transmit FIFO does not contain valid data when transmitting the last bit of a serial character, so the transmission is halted.

[Setting condition]

- TEND is set to 1 when transmit FIFO does not contain transmit data when the last bit of a one-byte serial character is transmitted.
- When the TDR register is not updated at the end of DE signal hold time with DE control function enable (SCR3.DEEN bit = 1).

[Clearing condition]

- When the transmission data was written to the TDR.TDAT[7:0] bits during SCR0.TE bit = 1.

RDRF Flag (Receive Data Full Flag)

(1) Non-FIFO selected (SCR3.FM bit = 0)

Indicates the presence of receive data in the RDR register.

[Setting condition]

- When the reception ends normally, and receive data is forwarded from the RSR register to the RDR register.

[Clearing condition]

- When write 1 to SSCR.RDRFC bit.
- When the read data is read from the RDR register.

(2) FIFO selected (SCR3.FM bit = 1)

Indicates that receive data has been transferred to the receive FIFO (RDR register), and the quantity of data in receive FIFO is equal to or greater than the specified receive triggering number. When FCR.RTRG[4:0] bits are set to 0, RDRF flag is set if the quantity of data in receive FIFO is greater than or equal to 1.

[Setting condition]

RDRF is set to 1 when the quantity of receive data in receive FIFO is equal to or greater than the specified receive triggering number*2.

[Clearing condition]

- When write 1 to SSCR.RDRFC bit.
- When the reception data is read form receive FIFO by the DTC or DMAC (the last block transfer when block transfer).

When the condition which becomes 1 and the condition which becomes 0 were formed at the same time, RDRF flag will be 0. After that, when the number of stored data in receive FIFO is judged, and if that is equal to or greater than RTRG value, RDRF flag is set to 1 after 1 PCLK cycle.

Note 2. Since the receive FIFO is a 32-stage FIFO register, the maximum number of data that can be read when RDF is 1 is indicated by the RFSR.R[5:0] bits. After reading all the data in the RDR register, continuing a read access results in an undefined value.

Note: In non-FIFO mode, do not clear the RDRF and TDRE flags by the SSCR register except when the communication is to be aborted.

36.2.18 I²C Status Register (SISR)

Address(es): RSCI10.SISR 000E 204Ch, RSCI11.SISR 000E 20CCh

	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
	—	—	—	—	—	—	—	—	—	—	—	—	IICSTIF	—	—	IICACKR
Value after reset:	0	0	0	0	0	0	0	0	0	0	x	x	0	x	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IICACKR	ACK Reception Data Flag	0: ACK received 1: NACK received	R
b1	—	Reserved	This bit is read as 0.	R
b2	—	Reserved	The read value is undefined.	R
b3	IICSTIF	Condition Generation Completed Flag	0: There are no requests for generating conditions or a condition is being generated. 1: A start, restart, or stop condition is completely generated.	R
b5, b4	—	Reserved	The read value is undefined.	R
b31 to b6	—	Reserved	These bits are read as 0.	R

IICACKR Flag (ACK Reception Data Flag)

Received ACK and NACK bits can be read from this flag.
The IICACKR flag is updated at the rising of SSCLn signal for the ACK/NACK receiving bit.

IICSTIF Flag (Condition Generation Completed Flag)

After generating a condition, this flag indicates that the generation is completed. When using the IICSTAREQ, IICRSTAREQ, or IICSTPREQ bit to cause generation of a condition, do so after setting the IICSTIF flag to 0.
When the IICSTIF flag is 1 while an interrupt request is enabled by setting the SCR0.TEIE bit, an STI request is output.
[Setting condition]

- Completion of the generation of a start, restart, or stop condition (however, in cases where this conflicts with any of the conditions for the flag becoming 0 listed below, the clearing condition takes precedence)

[Clearing condition]

- Writing 1 to SISCR.IICSTIFC bit
- When operation is not in simple I²C
- Writing 0 to SCR0.TE bit

36.2.19 Receive FIFO Status Register (RFSR)

Address(es): RSCI10.RFSR 000E 2050h, RSCI11.RFSR 000E 20D0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	FEC[5:0]						—	—	PEC[5:0]					
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
	—	—	R[5:0]						—	—	—	—	—	—	—	DR
Value after 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	DR	Receive Data Ready Flag	0: Receiving is in progress, or no received data has remained in receive FIFO after normally completed receiving. (receive FIFO is empty) 1: The following receive data does not come for a fixed period after storing data below the threshold in the receive FIFO	R
b1	—	Reserved	This bit is undefined when read.	R
b7 to b2	—	Reserved	These bits are read as 0.	R
b13 to b8	R[5:0]	Receive FIFO Data Count	(Valid in asynchronous mode (including multi-processor), clock synchronous mode, simple SPI mode, when SCR3.FM bit is 1.) Indicate the quantity of receive data stored in receive FIFO	R
b15, b14	—	Reserved	These bits are read as 0.	R
b21 to b16	PEC[5:0]	Parity Error Count	(Valid only in asynchronous mode) Indicates the quantity of data with a parity error among the receive data stored in the receive FIFO data register.	R
b23, b22	—	Reserved	These bits are read as 0.	R
b29 to b24	FEC[5:0]	Framing Error Count	(Valid only in asynchronous mode) Indicates the quantity of data with a framing error among the receive data stored in the receive FIFO data register.	R
b31, b30	—	Reserved	This bit is read as 0.	R

DR Flag (Receive Data Ready Flag)

Indicates that the quantity of data stored in the receive FIFO (RDR register) falls below the specified receive triggering number, and that no next data has been received yet after the elapse of 15 etu from the last stop bit in asynchronous mode. This flag becomes 1 only when the FIFO buffer is enabled in synchronous mode (including multiprocessor mode), and does not become 1 in other modes.

[Setting conditions]

DR is set to 1 when the following conditions are met.

- After receive FIFO (RDR register) receives less data than the specified receive triggering number, no next data has been received yet after the elapse of 15 etu*1 from the last stop bit
- SSR.AFER and APER flags are 0.

[Clearing conditions]

- When all receive data in the receive FIFO (RDR register) is read and 1 is written to RFSCR.DRC bit.
- When SCR3.FM bit is 0.

Refer to section 15.5.2, Level Detection to exit from the interrupt handling routine after setting the DR flag to 0 when the DR flag is set to the source of the receive error interrupt by setting the FCR.DRES bit to 1.

Note 1. This is equivalent to one and a half (1.5) frames in the 8-bit format with one stop bit (etu: elementary time unit).

R[5:0] Bits (Receive FIFO Data Count)

Indicate the quantity of receive data stored in receive FIFO.
00h means no receive data. 20h means receive FIFO is full.

PEC[5:0] Bits (Parity Error Count)

The value indicates the quantity of data stored in the receive FIFO registers with a parity error.

FEC[5:0] Bits (Framing Error Count)

The value indicates the quantity of data stored in the receive FIFO registers with a framing error.

36.2.20 Transmit FIFO Status Register (TFSR)

Address(es): RSCI10.TFSR 000E 2054h, RSCI11.TFSR 000E 20D4h

	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
	—	—	—	—	—	—	—	—	—	—	T[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	T[5:0]	Transmit FIFO Data Count	(Valid in asynchronous mode (including multi-processor), clock synchronous mode, simple SPI mode, when SCR3.FM bit is 1.) Indicate the quantity of non-transmit data stored in transmit FIFO	R
b31 to b6	—	Reserved	These bits are read as 0.	R

T[5:0] Bits (Transmit FIFO Data Count)

Indicate the quantity of non-transmitted data stored in transmit FIFO.

00h means no untransmitted data. 20h means transmit FIFO is full.

36.2.21 Manchester Mode Status Register (MMSR)

Address(es): RSCI10.MMSR 000E 2058h, RSCI11.MMSR 000E 20D8h

	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
	—	—	—	—	—	—	—	—	—	RSYNC	—	MCER	—	SBER	SYER	PFER
Value after 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	PFER	Preface Error Flag	This bit is set when a preface error (pattern mismatch) is detected 0: No preface error detected 1: Preface error detected	R
b1	SYER	Sync Error Flag	This bit is set when no edge is detected in the adjustable range during receive retiming 0: No receive Sync error detected 1: Receive Sync error detected	R
b2	SBER	Start Bit Error Flag	This bit is set when a pattern mismatch in the start bit area is detected 0: No start bit error detected 1: Start bit error detected	R
b3	—	Reserved	This bit is read as 0.	R
b4	MCER	Manchester Code Error Flag	Valid for manchester mode only 0: No Manchester code error occurred 1: Manchester code error has occurred	R
b5	—	Reserved	This bit is read as 0.	R
b6	RSYNC	Received Sync Data	It is valid when MMCR.SBLEN bit = 1 in manchester mode, 0 is read otherwise. 0: The received the Start Bit is DATA Sync 1: The received the Start Bit is COMMAND Sync	R
b31 to b7	—	Reserved	These bits are read as 0.	R

PFER Flag (Preface Error Flag)

This bit indicates that a preface error was detected when receiving frames in manchester mode.

Even when the SCR0.RE bit is set to 0 (serial reception is disabled), the PFER flag is not affected and retains its previous value.

[Setting condition]

When detecting a preface error when receiving frames in manchester mode

The following operations are performed when a preface error occurs.

<When MMCR.PFERIE = 1>

The received data is not transferred to the RDR register and no RXI interrupt request occurs. Instead, an ERI interrupt request occurs. Note that when the PFER flag is being set to 1, the subsequently received data is not transferred to the RDR register.

<When MMCR.PFERIE = 0>

The received data is transferred to the RDR register and an RXI interrupt request occurs. An ERI interrupt request is not generated. The subsequent receive operations are not affected even with the PFER flag being set to 1.

[Clearing condition]
Write 1 to MMSCR.PFERC bit.
Refer to section 15.5.2, Level Detection to exit from the interrupt handling routine after setting the PFER flag to 0.

SYER Flag (Sync Error Flag)

This bit indicates that a receive Sync error was detected when receiving frames in manchester mode with MMCR.SADJE bit = 1 (manchester edge retiming enabled).
Even when the SCR0.RE bit is set to 0 (serial reception is disabled), the SYER flag is not affected and retains its previous value.
[Setting condition]
When detecting a receive Sync error when receiving frames in manchester mode
The following operations are performed when a receive Sync error occurs.
<When MMCR.SYERIE = 1>
Although the received data is transferred to the RDR register, no RXI interrupt request occurs. Instead, an ERI interrupt request occurs. Note that when the SYER flag is being set to 1, the subsequently received data is not transferred to the RDR register.
<When MMCR.SYERIE = 0>
The received data is transferred to the RDR register and an RXI interrupt request occurs. An ERI interrupt request is not generated. The subsequent receive operations are not affected even with the SYER flag being set to 1.
[Clearing condition]
Write 1 to MMSCR.SYERC bit.
Refer to section 15.5.2, Level Detection to exit from the interrupt handling routine after setting the SYER flag to 0.

SBER Flag (Start Bit Error Flag)

This bit indicates that a start bit error was detected when receiving frames in manchester mode.
Even when the SCR0.RE bit is set to 0 (serial reception is disabled), the SBER flag is not affected and retains its previous value.
[Setting condition]
When detecting a start bit error when receiving frames in manchester mode
The following operations are performed when a start bit error occurs.
<When MMCR.SBERIE bit = 1>
The received data is not transferred to the RDR register and no RXI interrupt request occurs. Instead, an ERI interrupt request occurs. Note that when the SBER flag is being set to 1, the subsequently received data is not transferred to the RDR register.
<When MMCR.SBERIE bit = 0>
The received data is transferred to the RDR register and an RXI interrupt request occurs. An ERI interrupt request is not generated. The subsequent receive operations are not affected even with the SBER flag being set to 1.
[Clearing condition]
Write 1 to MMSCR.SBERC bit.
Refer to section 15.5.2, Level Detection to exit from the interrupt handling routine after setting the SBER flag to 0.

MCER Flag (Manchester Code Error Flag)

When data is received in manchester mode, Manchester code error is detected, and it is displayed. Even when the SCR0.RE bit is set to 0 (serial reception is disabled), the MCER flag is not affected and retains its previous value.
[Setting conditions]
When receiving in manchester mode and detecting manchester code error in data area of received frame.
Received data when an error occurs is transferred to the RDR register, but the RXI interrupt request is not generated.
When the manchester code error flag is set to 1, subsequent receive data is not transferred to the RDR register.
For details on manchester code error, see section 36.5.11, Errors in Manchester Mode.
[Clearing condition]
Write 1 to MMSCR.MCERC bit.
Refer to section 15.5.2, Level Detection to exit from the interrupt handling routine after setting the MCER flag to 0.

RSYNC Flag (Received Sync Data)

When manchester mode (SCR3.MOD[2:0] = 101b) and MMCR.SBLEN = 1, this bit indicates the type of Sync of the received the start bit. For other settings, it is fixed to 0.

36.2.22 Extended Serial Mode Status Register 0 (XSR0)

Address(es): RSCI10.XSR0 000E 205Ch, RSCI11.XSR0 000E 20DCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CF1RD[7:0]								CF0RD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	AEDF	COF	PIBDF	CF1MF	CF0MF	BFDF	BCDF	BFOF	—	—	—	—	—	—	RXDSF	SFSF
Value after 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	SFSF	Start Frame Status Flag	0: Start Frame detection disabled or Start Frame detection complete 1: Before Start Frame detection or during detection	R*1
b1	RXDSF	RXD Input Status Flag	0: RXD input to RSCI core is enabled 1: RXD input to RSCI core is disabled (RXD is not input to the RSCI core)	R*1
b7 to b2	—	Reserved	These bits are read as 0.	R
b8	BFOF	Break Field Low Width Output Complete Flag	0: When Break Field transmission is not completed 1: When Break Field transmission is completed	R
b9	BCDF	Bus Collision Detected Flag	0: When bus conflict is not detected 1: When bus conflict is detected	R
b10	BFDF	Break Field Low Width Detection Flag	0: When Break Field is not detected 1: When Break Field is detected	R
b11	CF0MF	Control Field 0 Match Flag	0: When Control Field 0 data and the compare data do not match 1: When Control Field 0 data and the compare data match	R
b12	CF1MF	Control Field 1 Match Flag	0: When Control Field 1 data and the compare data do not match 1: When Control Field 1 data and the compare data match	R
b13	PIBDF	Priority Interrupt Bit Detection Flag	0: When priority interrupt bit is not detected 1: When Priority interrupt bit is detected	R
b14	COF	Count Overflow Flag	0: When the counter for Break Field detection does not overflow 1: When the counter for Break Field detection overflows	R
b15	AEDF	Effective Edge Detection Flag	0: When Active edge is not detected 1: When Active edge is detected	R
b23 to b16	CF0RD[7:0]	Control Field 0 Received Data	Control Field 0 received data.	R
b31 to b24	CF1RD[7:0]	Control Field 1 Received Data	Control Field 1 received data.	R

Note 1. Wait at least 1 PCLKA cycle after the receive data full interrupt (RXI) before reading this register.

SFSF Flag (Start Frame Status Flag)

Indicates whether detect Start Frame is being detected.

[Setting condition]

- When 1 is written to XCR1.SDST bit.
- When a Break Field is detected in the Control Field 0/Control Field 1/Information Field phase and the transition to the Control Field 0 or Control Field 1 reception state occurs.

[Clearing condition]

- When XCR1.SDST bit is 0.
- When Start Frame detection is completed.

RXDSF Flag (RXD Input Status Flag)

Indicates the RXD input status to the RSCI core. When this bit is 1, RXD input is received only by the extended serial module and the Break Field is detected and is not input to the RSCI core.

36.2.23 Extended Serial Mode Status Register 1 (XSR1)

Address(es): RSCI10.XSR1 000E 2060h, RSCI11.XSR1 000E 20E0h

	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
	CCV[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	CCV[15:0]	Captured Count Value	Stores the 16-bit counter capture value.	R
b31 to b16	—	Reserved	These bits are read as 0.	R

CCV[15:0] Bits (Captured Count Value)

Stores the capture value of the 16-bit counter of the extended serial module.

When sending Start Frame

This register holds the previous value.

When receiving Start Frame with bit rate measurement disabled

If a Break Field is detected in the Break Field detection state (refer to Figure 36.73), the Break Field length is captured and held (counter value is captured at the rising edge of RXD).

If a Break Field is detected in a state other than the Break Field detection state, the previous value is retained.

If the counter overflows, it will not be captured.

When receiving Start Frame with bit rate measurement enabled

The count value is captured and held at the valid edge (both RXD edges). However, in the Break Field detection state, the count value is not captured even if a valid edge occurs. Counter capture value retention is canceled by this register read. Even if a valid edge occurs before reading, the counter value is not captured.

36.2.24 Status Clear Register (SSCR)

Address(es): RSCI10.SSCR 000E 2068h, RSCI11.SSCR 000E 20E8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RDRFC	—	TDREC	AFERC	APERC	MFFC	—	ORERC	—	—	—	—	—	DFERC	DPERC	DCMFC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	ERSC	ERS Clear	Setting this bit to 1 clears the SSR.ERS flag. The read value is always 0.	W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	DCMFC	DCMF Clear	Setting this bit to 1 clears the SSR.DCMF flag. The read value is always 0.	W
b17	DPERC	DPER Clear	Setting this bit to 1 clears the SSR.DPER flag. The read value is always 0.	W
b18	DFERC	DFER Clear	Setting this bit to 1 clears the SSR.DFER flag. The read value is always 0.	W
b23 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	ORERC	ORER Clear	Setting this bit to 1 clears the SSR.ORER flag. The read value is always 0.	W
b25	—	Reserved	This bit is read as 0. The write value should be 0.	R
b26	MFFC	MFF Clear	Setting this bit to 1 clears the SSR.MFF flag. The read value is always 0.	W
b27	APERC	APER Clear	Setting this bit to 1 clears the SSR.APER flag. The read value is always 0.	W
b28	AFERC	AFER Clear	Setting this bit to 1 clears the SSR.AFER flag. The read value is always 0.	W
b29	TDREC	TDRE Clear	Setting this bit to 1 clears the SSR.TDRE flag. The read value is always 0.	W
b30	—	Reserved	This bit is read as 0. The write value should be 0.	R
b31	RDRFC	RDRF Clear	Setting this bit to 1 clears the SSR.RDRF flag. The read value is always 0.	W

36.2.25 I²C Status Clear Register (SISCR)

Address(es): RSCI10.SISCR 000E 206Ch, RSCI11.SISCR 000E 20ECh

	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
	—	—	—	—	—	—	—	—	—	—	—	—	IICSTIFC	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b2	—	Reserved	This bit is read as 0.	R
b3	IICSTIFC	IICSTIF Clear	Setting this bit to 1 clears the SISR.IICSTIF flag. The read value is always 0.	W
b31 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R

36.2.26 Receive FIFO Status Clear Register (RFSCR)

Address(es): RSCI10.RFSCR 000E 2070h, RSCI11.RFSCR 000E 20F0h

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DRC
Value after 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	DRC	DR Clear	Setting this bit to 1 clears the RFSR.DR flag. The read value is always 0.	W
b1	—	Reserved	This bit is read as 0.	R
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

36.2.27 Manchester Mode Status Clear Register (MMSCR)

Address(es): RSCI10.MMSCR 000E 2074h, RSCI11.MMSCR 000E 20F4h

	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
	—	—	—	—	—	—	—	—	—	—	—	MCERC	—	SBERC	SYERC	PFERC
Value after 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	PFERC	PFER Clear	Setting this bit to 1 clears the MMSR.PFER flag. The read value is always 0.	W
b1	SYERC	SYER Clear	Setting this bit to 1 clears the MMSR.SYER flag. The read value is always 0.	W
b2	SBERC	SBER Clear	Setting this bit to 1 clears the MMSR.SBER flag. The read value is always 0.	W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b4	MCERC	MCER Clear	Setting this bit to 1 clears the MMSR.MCER flag. The read value is always 0.	W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R

36.2.28 Extended Serial Mode Status Clear Register (XSCR)

Address(es): RSCI10.XSCR 000E 2078h, RSCI11.XSCR 000E 20F8h

	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
	AEDCL	COFC	PIBDCL	CF1MCL	CF0MCL	BFDCL	BCDCL	BFOC	—	—	—	—	—	—	—	—
Value after 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
b8	BFOC	BFOF Clear	Setting this bit to 1 clears the XSR0.BFOF flag. The read value is always 0.	W
b9	BCDCL	BCDF Clear	Setting this bit to 1 clears the XSR0.BCDF flag. The read value is always 0.	W
b10	BFDCL	BFDF Clear	Setting this bit to 1 clears the XSR0.BFDF flag. The read value is always 0.	W
b11	CF0MCL	CF0MF Clear	Setting this bit to 1 clears the XSR0.CF0MF flag. The read value is always 0.	W
b12	CF1MCL	CF1MF Clear	Setting this bit to 1 clears the XSR0.CF1MF flag. The read value is always 0.	W
b13	PIBDCL	PIBDF Clear	Setting this bit to 1 clears the XSR0.PIBDF flag. The read value is always 0.	W
b14	COFC	COF Clear	Setting this bit to 1 clears the XSR0.COF flag. The read value is always 0.	W
b15	AEDCL	AEDF Clear	Setting this bit to 1 clears the XSR0.AEDF flag. The read value is always 0.	W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

36.2.29 HBS Support Mode Control Register (HBSCR)

Address(es): RSCI10.HBSCR 000E 201Eh, RSCI11.HBSCR 000E 209Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	LPS	AOE	—	HBSE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	HBSE	HBS Support Mode Enable	0: Pulse width for data 0 is specified as 100% of a bit period (NRZ coding). 1: Pulse width for data 0 is specified as 50% of a bit period (RZI coding and inversion).	R/W *1
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R
b2	AOE	Alternate Output Enable	0: Data is output from the TXDn pin. 1: Data 0 is alternately output from the TXDAn and TXDBn pins.	R/W *1
b3	LPS	Leading Output Pin Select	0: When HBSE bit = 1 and AOE bit = 1, transmission starts from TXDAn pin 1: When HBSE bit = 1 and AOE bit = 1, transmission starts from TXDBn pin	R/W *1
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Writable only when SCR0.TE bit = 0 and SCR0.RE bit = 0.

HBSE Bit (HBS Support Mode Enable)

When this bit is 1, the transmitter encodes the transmit data to the negative logic RZI code and the receiver decodes the receive data to the NRZ code. Also, transmit data can be output from the TXDAn/TXDBn pin. This function should be used only in asynchronous mode.

AOE Bit (Alternate Output Enable)

This bit is used to select whether the data is output from the TXDn pin or data 0 is alternately output from the TXDAn and TXDBn pins in HBS support mode.

LPS Bit (Leading Output Pin Select)

This bit is used when the AOE bit is 1 in HBS support mode.

When it is set to 0, the start bit is transmitted from TXDAn pin and the data 0 is output alternately from the TXDBn and TXDAn pins.

When it is set to 1, the start bit is transmitted from TXDBn pin and the data 0 is output alternately from the TXDAn and TXDBn pins.

For details, refer to the operation description in section 36.6, HBS Support Mode.

36.3 Operation in Asynchronous Mode

Figure 36.3 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 held in the mark state (high level) when not communicating.
The RSCI monitors the communications line. When the RSCI detects a start bit, it starts serial communication. The detection condition of the start bit changes according to the SCR3.RXDESEL bit setting. RSCI regards space (low level) as a start bit when SCR3.RXDESEL bit is 0. RSCI regards a fall edge as a start bit when RXDESEL bit is 1.
Inside the RSCI, the transmitter and receiver are independent units, enabling full-duplex communications. Both the transmitter and the receiver also have a double-buffered structure (it has also FIFO mode), so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

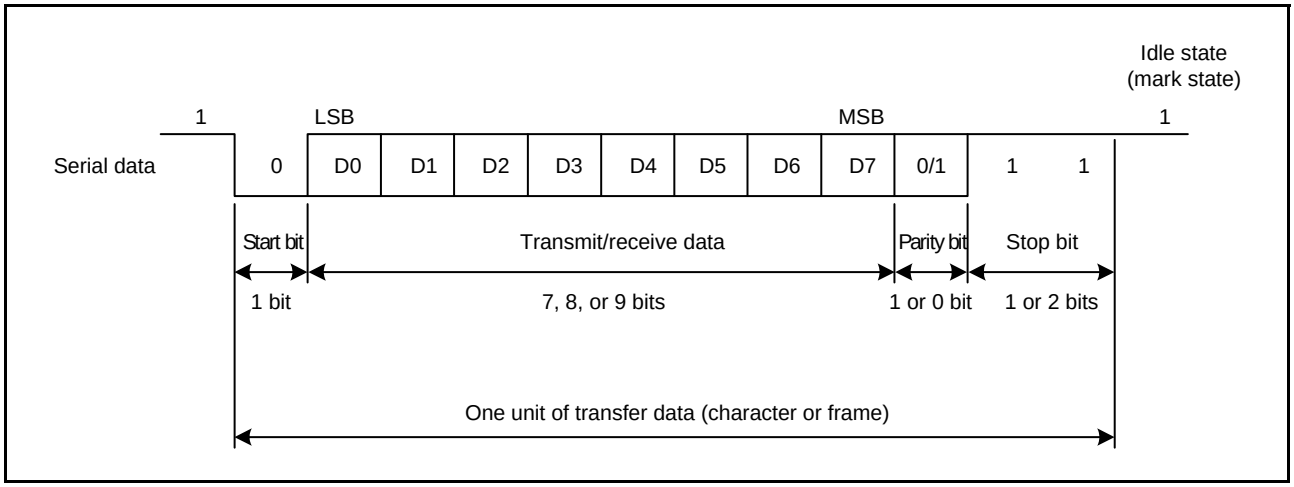


Figure 36.3 Data Format in Asynchronous Serial Communications (Example with 8-Bit Data, Parity, 2 Stop Bits)

36.3.1 Serial Data Transfer Format

Table 36.27 lists the serial data transfer formats that can be used in asynchronous mode.
Any of 18 transfer formats can be selected according to the SCR1 and SCR3 setting. For details of multi-processor function, refer to section 36.4, Multi-Processor Communication Function.

Table 36.27 Serial Transfer Formats (Asynchronous Mode)

SCR3		SCR1	SCR3		Serial Transfer Format and Frame Length
CHR[1]	CHR[0]	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12 13
0	0	0	0	0	S 9-bit data STOP
0	0	0	0	1	S 9-bit data STOP STOP
0	0	1	0	0	S 9-bit data P STOP
0	0	1	0	1	S 9-bit data P STOP STOP
1	0	0	0	0	S 8-bit data STOP
1	0	0	0	1	S 8-bit data STOP STOP
1	0	1	0	0	S 8-bit data P STOP
1	0	1	0	1	S 8-bit data P STOP STOP
1	1	0	0	0	S 7-bit data STOP
1	1	0	0	1	S 7-bit data STOP STOP
1	1	1	0	0	S 7-bit data P STOP
1	1	1	0	1	S 7-bit data P STOP STOP
0	0	—	1	0	S 9-bit data MPB STOP
0	0	—	1	1	S 9-bit data MPB STOP STOP
1	0	—	1	0	S 8-bit data MPB STOP
1	0	—	1	1	S 8-bit data MPB STOP STOP
1	1	—	1	0	S 7-bit data MPB STOP
1	1	—	1	1	S 7-bit data MPB STOP STOP

S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multi-processor bit

36.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the RSCI operates on a base clock with a frequency of 16 times*1 the bit rate.
In reception, the RSCI samples the falling edge of the start bit using the base clock, and performs internal synchronization*2. When sampling timing does not adjust (SCR4.RTADJ bit = 0 or SCR4.RTADJ bit = 1 and SCR4.RTMG[2:0] bits = 000b), 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 36.4. Thus the reception margin in asynchronous mode is determined by formula (1) below.

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ (\%) \quad \cdots \text{Formula (1)}$$

M: Reception margin
N: Ratio of bit rate to clock
(N = 16 when SCR2.ABCSE = 0 and SCR2.ABCS = 0, N = 8 when SCR2.ABCSE = 0 and SCR2.ABCS = 1, N = 6 when SCR2.ABCSE = 1)
D: Duty cycle of clock (D = 0.5 to 1.0)
L: Frame length (L = 9 to 13)
F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined by the formula below.

$M = \{0.5 - 1/(2 \times 16)\} \times 100\ (\%) = 46.875\ (\%)$

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 ABCSE and ABCS bits in the SCR2 register is 0. When the ABCSE bit is 0 and 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. When the ABCSE bit is 1, a frequency of 6 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 3rd pulse of the base clock.

Note 2. The determination condition of the start bit is as follows.
In the case of the function of adjust sampling timing is OFF (RTADJ bit = 0):
The determination condition of a start bit is that Low beyond half bit length continues. It is same as the sampling timing.
In Figure 36.4, the low period should be kept over 8-cycles to detect a start bit. If Low period does not keep over 8-cycles, the RSCI judges this as a noise. So, the RSCI does not start reception and wait start bit.
In the case of the function of adjust sampling timing is ON (RTADJ bit = 1):
The determination condition of a start bit is that Low keeps up until the sampling timing.
Adjusting the sampling timing forward (RTMG[3] bit = 1) increases the possibility of erroneously determining a noise as the start bit.

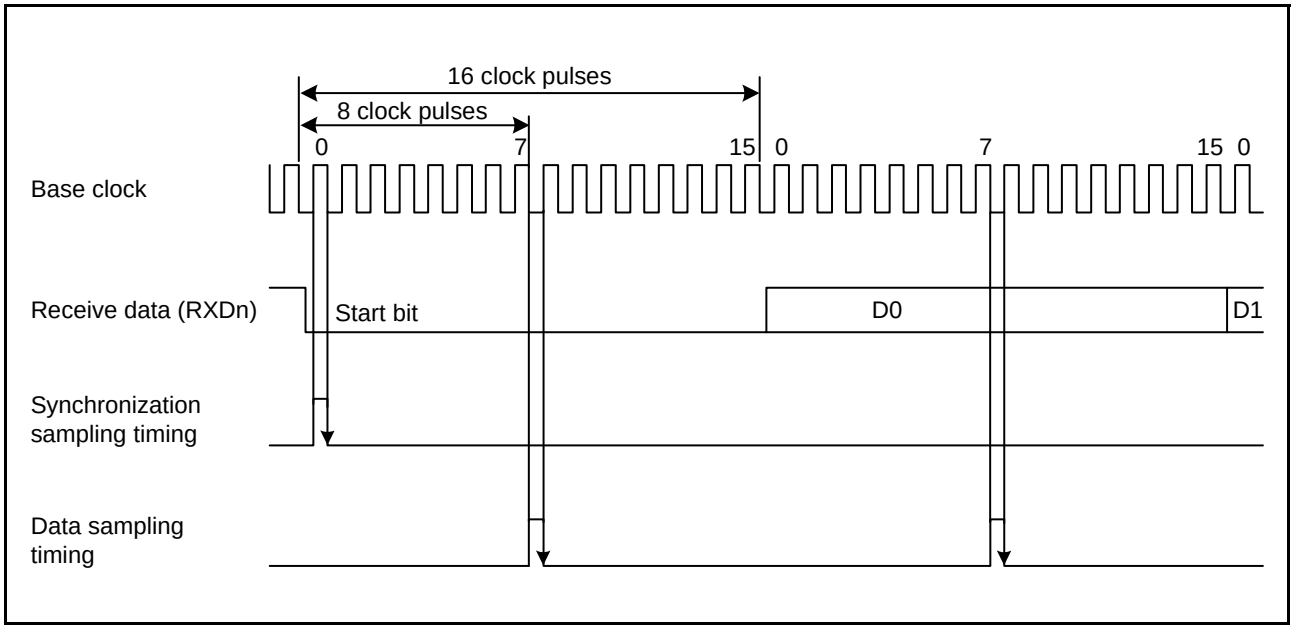


Figure 36.4 Receive Data Sampling Timing in Asynchronous Mode

36.3.3 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 RSCI's transfer clock, according to the setting of the SCR3.CKE[1:0] bits.
When an external clock is input to the SCKn pin, the clock frequency should be 16 times the bit rate (when SCR2.ABCS bit = 0) and 8 times the bit rate (when SCR2.ABCS bit = 1).
When the RSCI 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 36.5.
If you selected an internal clock, the SCKn pin is outputted after SCR0.TE bit is set to 1 or SCR0.RE bit is set to 1.

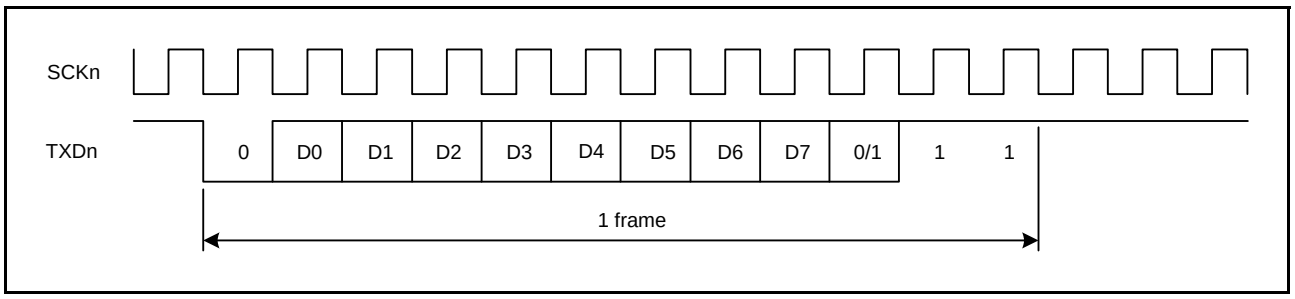


Figure 36.5 Phase Relationship between Output Clock and Transmit Data (Asynchronous Mode: SCR1.PE Bit = 1, SCR3.CHR[1:0] Bits = 10b, MP Bit = 0, STOP Bit = 1)

36.3.4 Double-Speed Mode and Divide-by-6 Mode

When SCR2.ABCS bit is set to 1, the RSCI operates on the bit rate twice that in the case where ABCS bit is set to 0. And when SCR2.BGDM bit is set to 1, the cycle of the base clock is halved and the bit rate is doubled from that in the case where BGDM bit is set to 0. When SCR3.CKE[1] bit is set to 0 and the on-chip baud rate generator is selected, setting the ABCS and BGDM bits to 1 allows the RSCI to operate on a bit rate four times that in the case ABCS bit = 0 and BGDM bit = 0.
When SCR2.ABCSE bit is set to 1, the number of base clock pulses are 6 during a period of 1 bit, and the base clock frequency is half. And RSCI works 16/3 times of bit rate compared with a case of SCR2.ABCS bit = 0, SCR2.BGDM bit = 0 and SCR2.ABCSE bit = 0.
As shown by Formula (1) in section 36.3.2, Receive Data Sampling Timing and Reception Margin in Asynchronous Mode, the reception margin decreases when SCR2.ABCS bit is set to 1 or SCR2.ABCSE bit is set to 1. Therefore, if the desired bit rate can be obtained with SCR2.ABCS bit set to 0 and SCR2.ABCSE bit set to 0, it is recommended to use the RSCI with SCR2.ABCS bit set to 0 and SCR2.ABCSE bit set to 0.

36.3.5 CTS and RTS Functions

The CTS function is the transmission control function by the CTSn# pin. Setting the SCR1.CTSE bit to 1 enables the CTS function. For the functions of CTS and RTS, you can select the alternate setting that uses either function with one pin or the dedicated setting that uses each function independently with two pins. This setting is done with the SCR1.CRSEP bit.
When the CTS function is enabled, placing the low level on the CTSn# pin causes transmission to start.
Even if the CTSn# pin goes high after transmission starts, the frame being transmitted is not affected and transmission will continue.
The RTS function is the transmission request function by the RTSn# pin. In the RTS function, the RTSn# pin output low level when reception becomes possible. Conditions for output of the low and high level are shown below.

(a) When the SCR3.FM Bit is 0 (Non-FIFO Mode)

[Conditions for low-level output]
When the following conditions are all satisfied

- The SCR0.RE bit is 1
- There are no received data yet to be read and reception is not in progress
- The ORER, AFER, and APER flags in the SSR are all 0

[Condition for high-level output]
When the conditions for low-level output are not satisfied

(b) When the SCR3.FM Bit is 1 (FIFO Mode)

[Conditions for low-level output]
When the following conditions are all satisfied

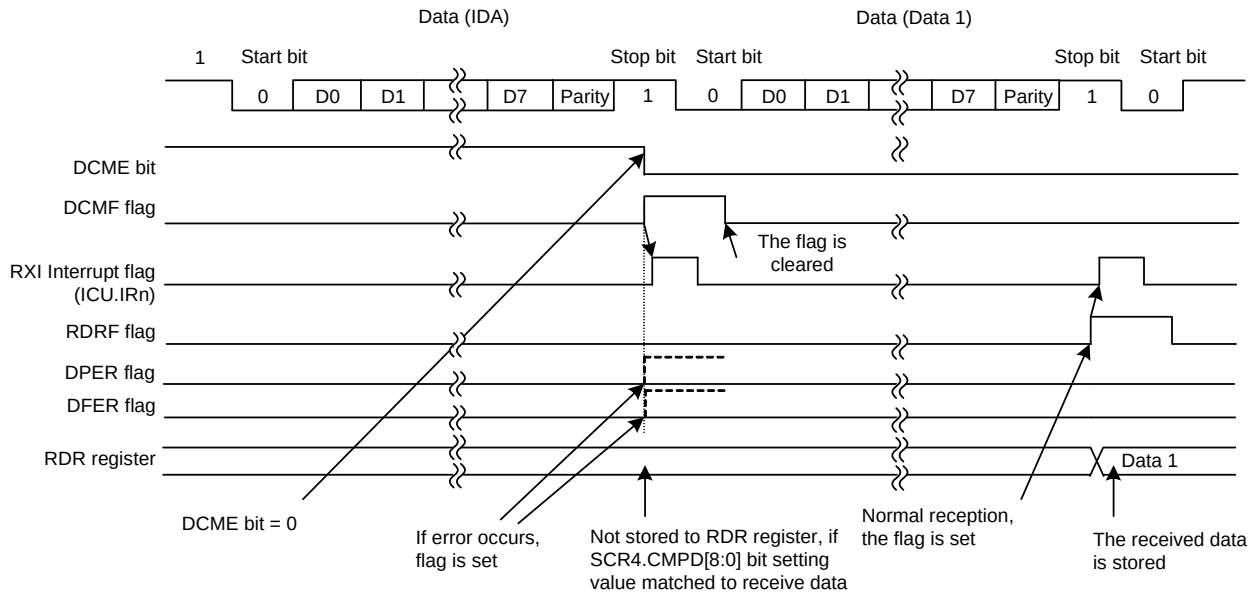
- The SCR0.RE bit is 1
- The number of data stored in the receive FIFO is less than the threshold (FCR.RSTRG[4:0] bits)
- The SSR.ORER (RDR.ORER) flag is 0

[Condition for high-level output]
When the conditions for low-level output are not satisfied

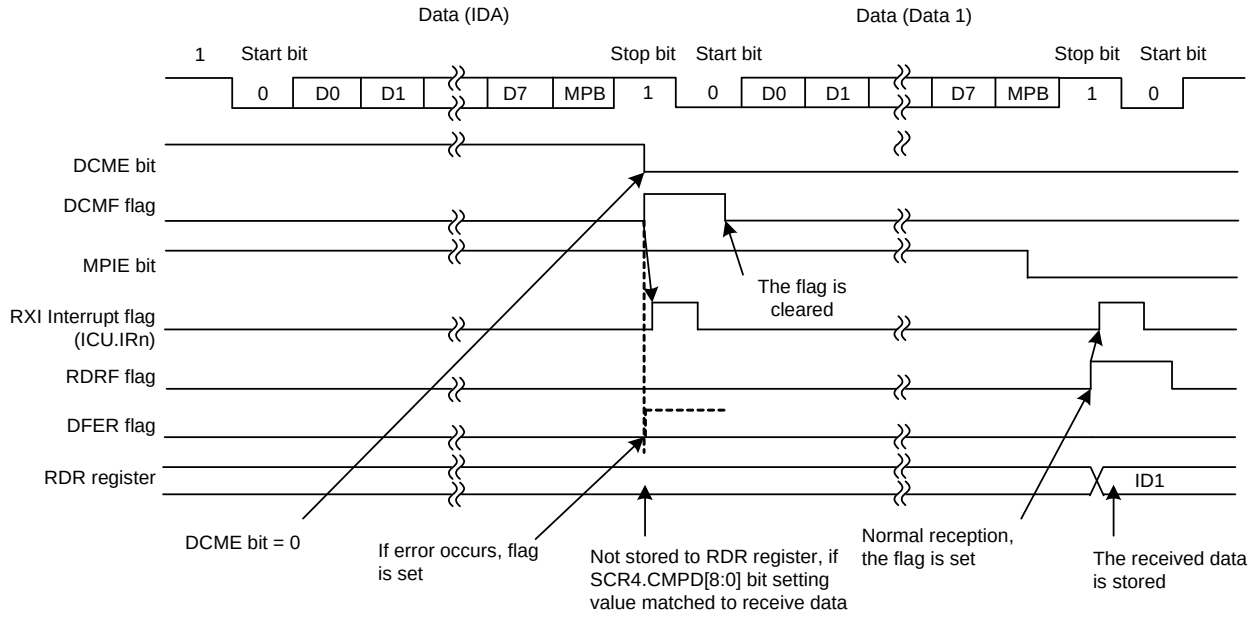
36.3.6 Data Match Detection

The data match detection function can use only the asynchronous mode.
If SCR0.DCME bit is set to 1*2, when one frame of data has been received, RSCI compares that receive data with the data which is set to SCR4.CMPD[8:0] bits. If RSCI detects the match to the comparison data (SCR4.CMPD[8:0] bits*1) with receive data, RSCI can issue RXI interrupt request.
If SCR3.MP bit is set to 0, this comparative target in communication data is valid only data field in receive format. In multi-processor mode (SCR3.MP bit = 1), if SCR0.IDSEL bit is set to 1, the reception data at which MPB is 1 detects data match or unmatch, and the reception data at which MPB is 0 detects always unmatch. If SCR0.IDSEL bit is set to 0, RSCI detects data match or unmatch despite the value of the MPB of the reception data at every reception complete.
Until RSCI detects the match to the comparison data (SCR4.CMPD[8:0] bits) with receive data, the communication data is skipped (discarded), and RSCI can not detect parity error, framing error. When RSCI detects the match, the SCR0.DCME bit is automatically cleared, and SSR.DCMF flag is set to 1. If SCR0.IDSEL bit is set to 1 at this time, SCR0.MPIE bit is automatically cleared. And if SCR0.IDSEL bit is set to 0 at this time, SCR0.MPIE bit is kept. At the same time, if SCR0.RIE bit is set to 1, RSCI issues RXI interrupt request. If RSCI detects framing error in comparative receive data which is detected the match, SSR.DFER flag is set to 1, and if RSCI detects parity error in that frame, SSR.DPER flag is set to 1. That comparative receive data and MPB are not stored to RDR register, and SSR.RDRF flag is retained to 0.
After RSCI detects the match, and the SCR0.DCME bit is automatically cleared, it receives next data continuously in current register setting.
When SSR.DFER flag or SSR.DPER flag is set, the data match isn't detected. Before making the data match detection function effective, please be sure to set SSR.DFER and SSR.DPER flags as 0.
Figure 36.6 and Figure 36.7 show the data match detection example.

Note 1. This comparative target can select one length of 3 types, they are CMPD[6:0] bits with 7bit length enable, CMPD[7:0] bits with 8bit, and CMPD[8:0] bits with 9bit length.
Note 2. Set the SCR0.DCME bit to 1 before receiving the start bit of the received frame that performs data matching.

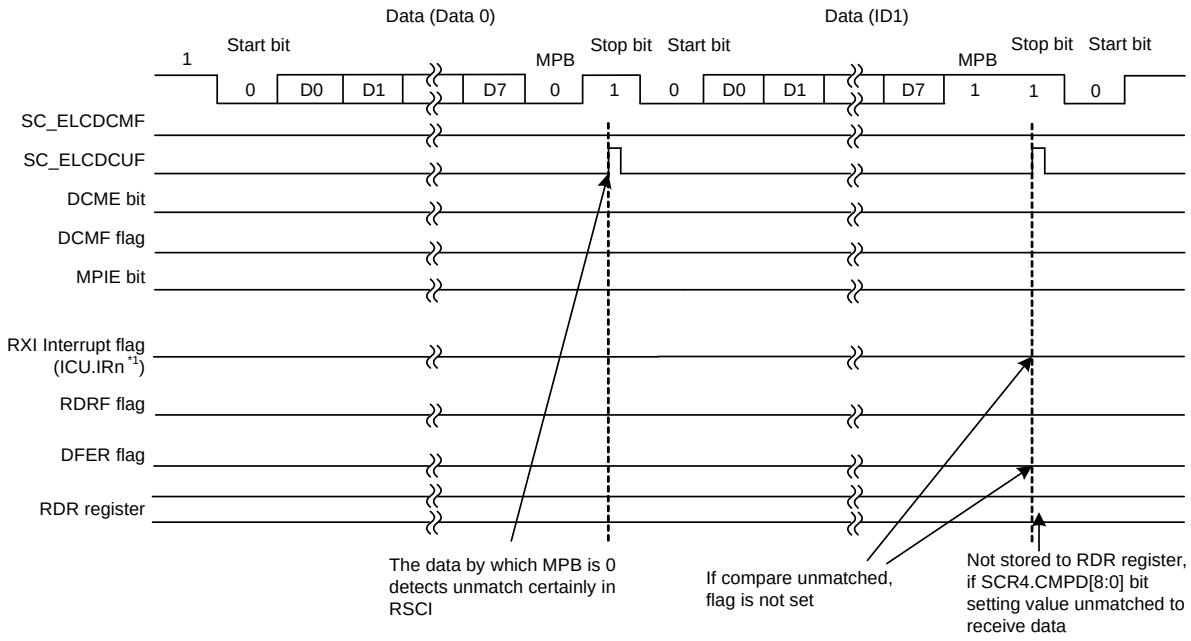


(a) Example of compare matched between receive data and SCR4.CMPD[8:0] bits (8bit length/parity/normal mode)

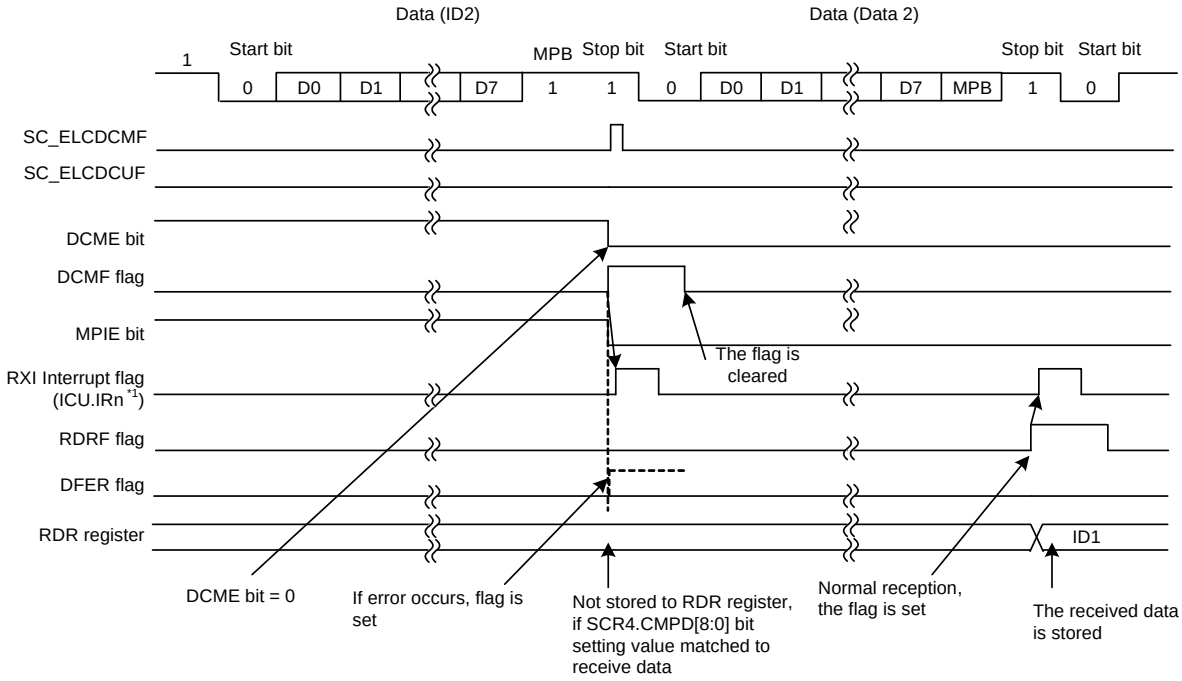


(b) Example of compare matched between receive data and SCR4.CMPD[8:0] bits (8bit length/multi-processor mode)

Figure 36.6 Example of Data Match Detection (1) 8bit-Data



(a) Example of compare unmatched between receive data and SCR4.CMPD[8:0] bits (8bit length/IDSEL = 1)



(b) Example of compare matched between receive data and SCR4.CMPD[8:0] bits (8bit length/IDSEL = 1)

Figure 36.7 Example of Data Match Detection (2) Multi-Processor Mode/8bit-Data

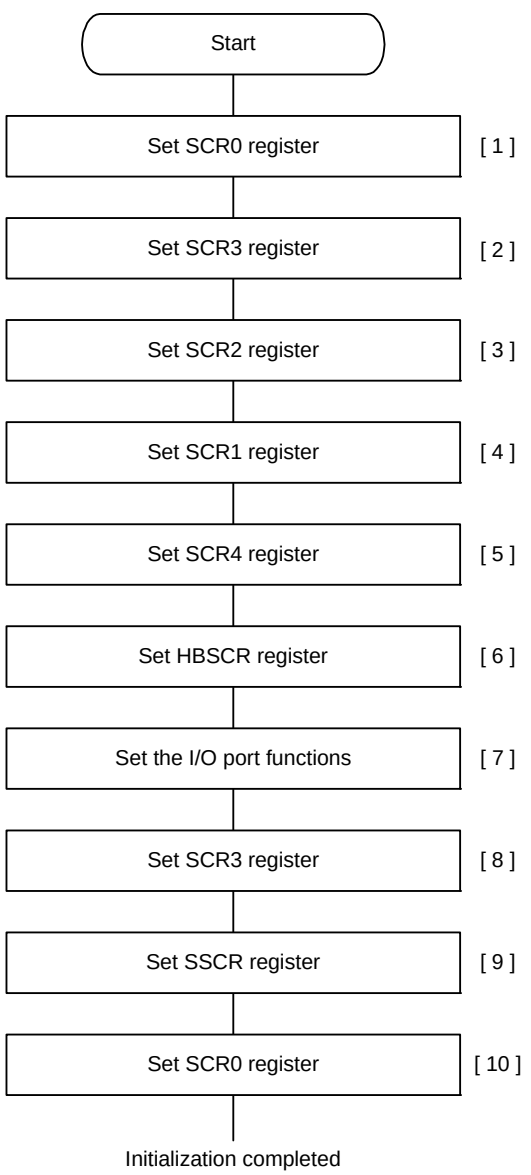
36.3.7 RSCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, start by writing 0 to SCR0.TE and SCR0.RE bits (or writing the initial value to SCR0 register) and then continue through the procedure (select to non-FIFO or FIFO) for RSCI given in Figure 36.8 or Figure 36.9. Whenever the operating mode or transfer format is changed, SCR0.TE and SCR0.RE bits must be initialized before the change is made.

When the external clock is used in asynchronous mode, ensure that the clock signal is supplied even during initialization.

Note that setting the SCR0.RE bit to 0 initializes neither the ORER flags, AFER flags, APER flags, RDRF flags, DR flags, and RDR register. In FIFO mode, even if the TE bit is set as 0, the TEND flag isn't initialized, so please be careful. Please be also careful at the time of change in the operation mode.

Moreover, note that switching the value of the SCR0.TE bit from 0 to 1 while the SCR0.TIE bit is 1 leads to the generation of a TXI interrupt request.



[1] Set the following bits of SCR0 register to 0.
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set up following function and communication mode.
Driver control function for RS-485
FIFO no-use
Multi-Processor mode
Communication mode (MOD[2:0] bits = 000b)
Transmission/reception format
Clock enable (Leave the CKE[1:0] bits at their initial values when outputting the clock)
Leave unused bits at their initial values.

[3] Set up the bit-rate-modulation function*1, 2, select clock, set bit rate*2.

[4] Set up the Noise-filter function, the loop-back function, communication terminal status, the parity check function, and the CTS/RTS function.

[5] Set up the adjust sampling timing function and the adjust transmit timing function.

[6] When using the HBS support mode, enable or set the function.
If this function is not used, leave the default value.

[7] Make I/O port settings to enable input and output functions as required for TXDn (TXDAn/TXDBn), RXDn, and SCKn pins.

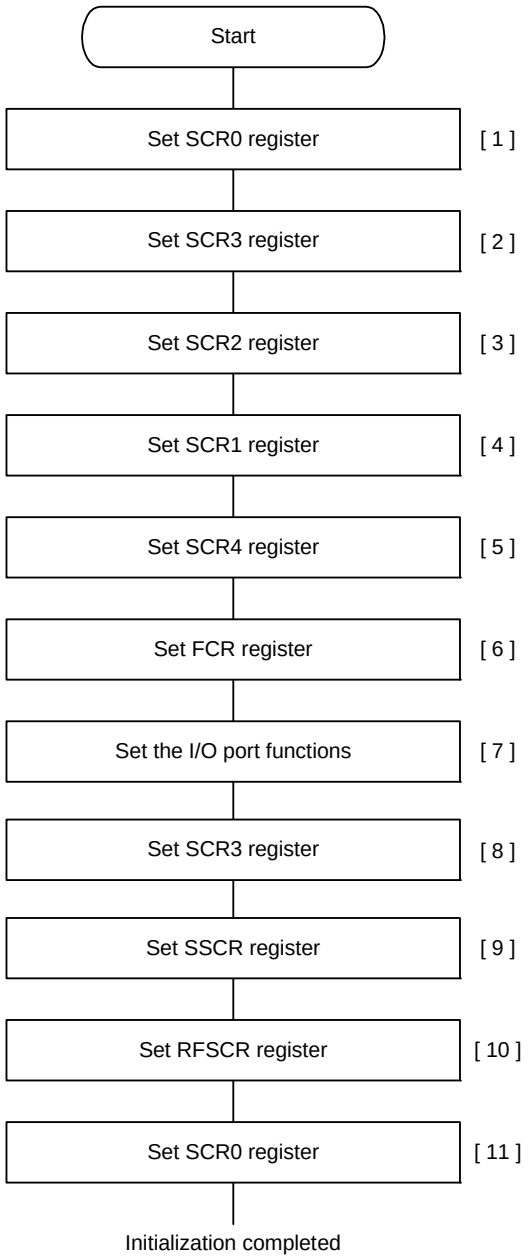
[8] Set clock enable bit (CKE[1:0] bits) at this step when outputting the clock. After setting the bits, the SCKn pin becomes output mode (the clock is output after setting the TE or RE bit to 1).

[9] Write 1 to the following bits and clear the corresponding flag.
SSCR.RDRFC, AFERC, APERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
If it's initialization flow after a reset, you can skip this step.

[10] Set the TE or RE bit to 1.
To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time. Also, set the data match detection function and multiprocessor function.
Setting the TE and RE bits allows TXDn (TXDAn/TXDBn) and RXDn to be used.

Note 1. If you do not use the bit rate modulation function, you do not need to set it.
Note 2. If you use an external clock, you do not need to set it.

Figure 36.8 Sample RSCI Initialization Flowchart (Asynchronous Mode/Non-FIFO Mode)



[1] Set the following bits of SCR0 register to 0.
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set up following function and communication mode.
Driver control function for RS-485
FIFO use
Multi-Processor mode
Communication mode (MOD[2:0] bits = 000b)
Transmission/reception format
Clock enable (Leave the CKE[1:0] bits at their initial values when outputting the clock)
Leave unused bits at their initial values.

[3] Set up the bit-rate-modulation function*1, 2,select clock, set bit rate*2.

[4] Set up the Noise-filter function, the loop-back function, communication terminal status, the parity check function, and the CTS/RTS function.

[5] Set up the adjust sampling timing function and the adjust transmit timing function.

[6] Set the TFRST and RFRST to 1 to empty FIFO.
Set the DRES, TTRG[4:0], RTRG[4:0], and RSTRG[4:0] bits

[7] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[8] Set clock enable bit (CKE[1:0] bits) at this step when outputting the clock. After setting the bits, the SCKn pin becomes output mode (the clock is output after setting the TE or RE bit to 1).

[9] Write 1 to the following bits and clear the corresponding flag.
SSCR.RDRFC, AFERC, APERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
If it is initialization flow after a reset, you can skip this step.

[10] Write 1 to The RFSCR.DRC bit and clear the corresponding flag.
If it is initialization flow after a reset, you can skip this step.

[11] Set the TE or RE bit to 1.
To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time. Also, set the data match detection function and multiprocessor function.
Setting the TE and RE bits allows TXDn and RXDn to be used.

Note 1. If you do not use the bit rate modulation function, you do not need to set it.
Note 2. If you use an external clock, you do not need to set it.

Figure 36.9 Sample RSCI Initialization Flowchart (Asynchronous Mode, FIFO Mode)

Figure 36.10 shows an example of the timing when data is transmitted after reset is released, and RSCI is set to asynchronous mode according to Figure 36.8 or Figure 36.9. As shown in the figure, when the pin function is set to the TXDn pin, the SCR0.TE bit is 0, so the pin is high impedance. When transmit data is written after setting the SCR0.TE bit to 1, data transmission starts. There is a transmission wait time from writing TDR register to data transmission starts. In asynchronous mode, TXDn is high during this period.

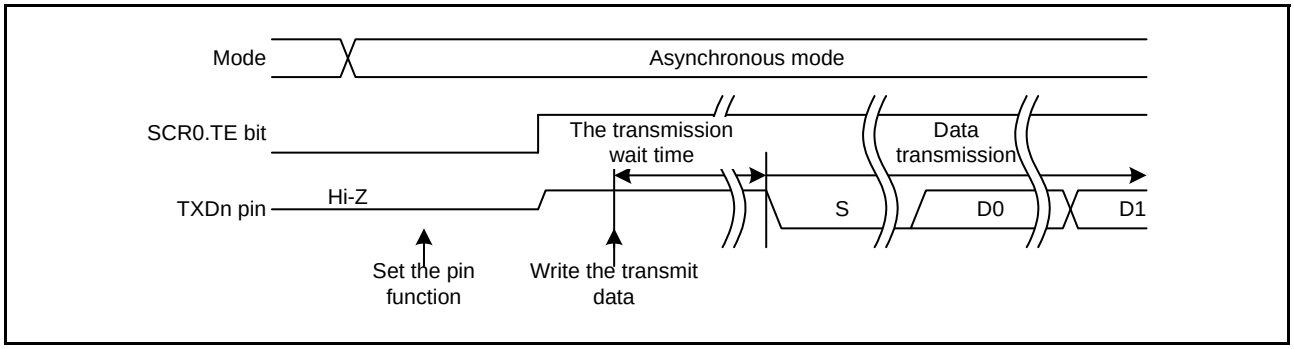


Figure 36.10 Example of Data Transmission Timing in Asynchronous Mode

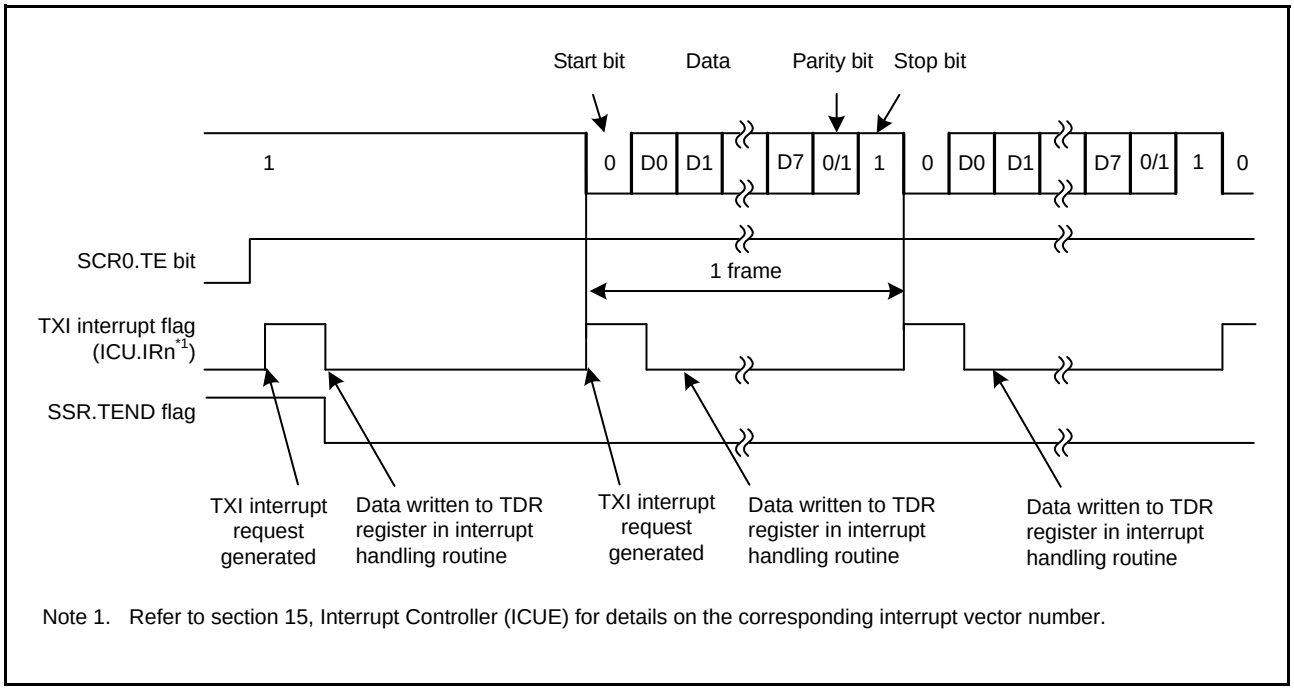
36.3.8 Serial Data Transmission (Asynchronous Mode)

(1) Non-FIFO Mode

Figure 36.11 to Figure 36.13 show an example of the operation for serial transmission in asynchronous mode.
In serial transmission, the RSCI operates as described below.

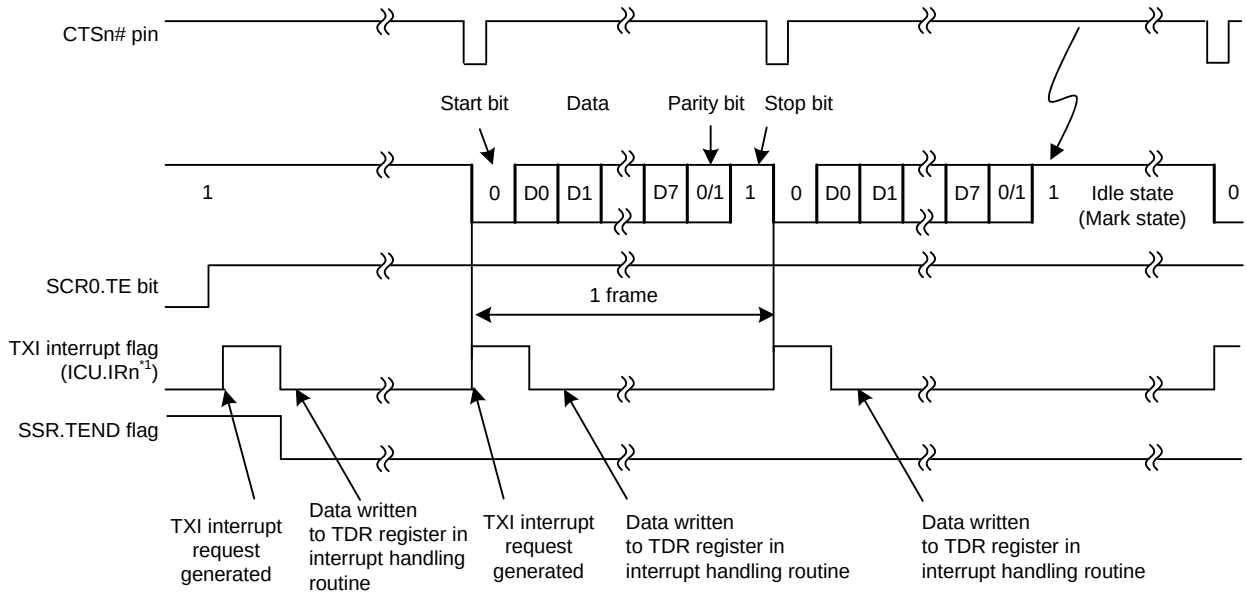
1. The RSCI transfers data from the TDR register to the TSR register when data is written to the TDR register in the TXI interrupt handling routine. At the beginning of transmission, set 1 to the SCR0.TE bit and the SCR0.TIE bit simultaneously. Then the TXI interrupt request is generated.
2. Transmission starts at data transfer from the TDR register to the TSR register when the SCR1.CTSE bit = 0 (CTS function is disabled) or when the CTS# pin level is low. If the SCR0.TIE bit is 1 at this time, a TXI interrupt request is generated. Continuous transmission is obtainable by writing the next transmit data to the TDR register in the TXI interrupt handling routine before transmission of the current transmit data is completed. Write the last transmission data, then the last data transmit start, and TXI is generated. When TEI interrupt requests are in use, before the last data transmit end, set the SCR0.TIE bit to 0 (a TXI interrupt request is disabled) and the SCR0.TEIE bit to 1 (a TEI interrupt request is enabled) using the TXI interrupt handling routine.
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 RSCI checks for updating of (writing to) the TDR register at the time of stop bit output.
5. When the TDR register is updated, setting of the SCR1.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn# pin cause the next transfer of the next transmit data from the TDR register to the TSR register and sending of the stop bit, after which serial transmission of the next frame starts.
6. If the TDR register is not updated, the SSR.TEND flag is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output. If the SCR0.TEIE bit is 1 at this time, a TEI interrupt request is generated.

Figure 36.15 shows the example of Serial Transmission Flowchart.



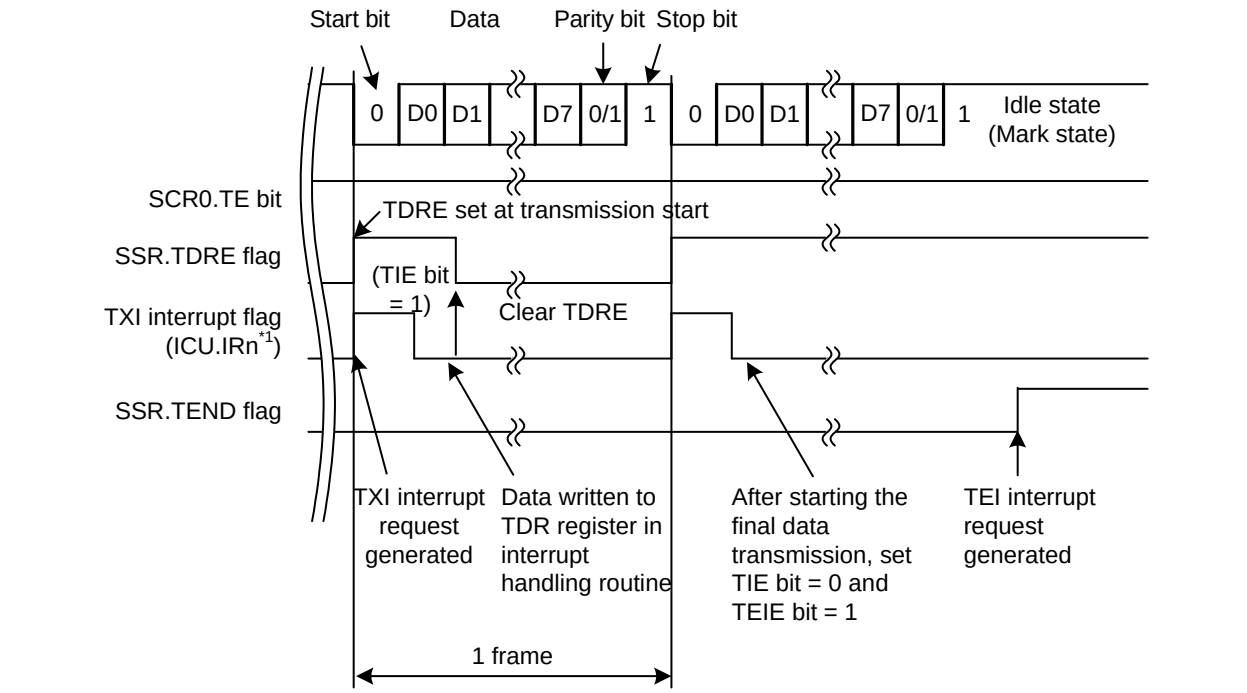
Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 36.11 Example of Operation for Serial Transmission in Asynchronous Mode (1)
(with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, at the Beginning of Transmission)



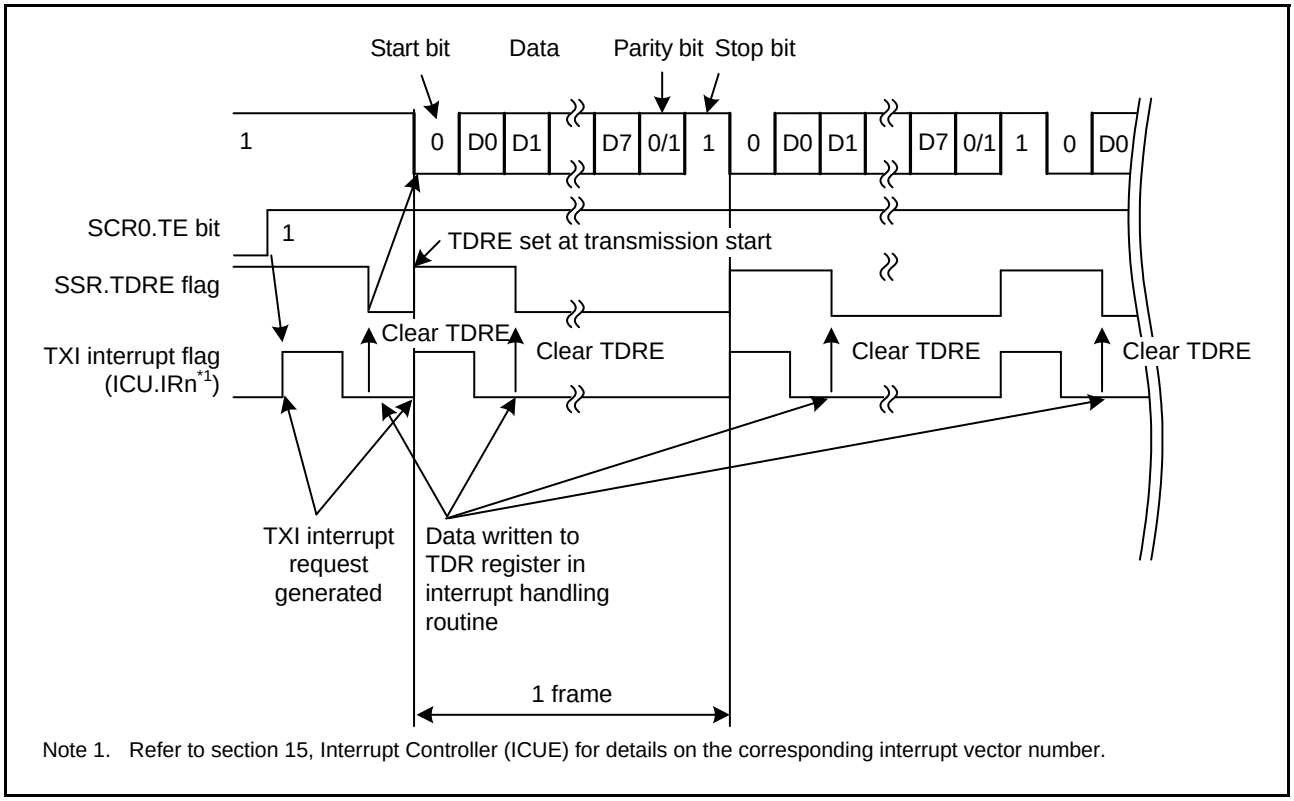
Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 36.12 Example of Operation for Serial Transmission in Asynchronous Mode (2) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Used, at the Beginning of Transmission)



Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 36.13 Example of Operation for Serial Transmission in Asynchronous Mode (3) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, from the Middle of Transmission until Transmission Completion)



Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 36.14 Example of Operation for Serial Transmission in Asynchronous Mode (4) (with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, during Transmission)

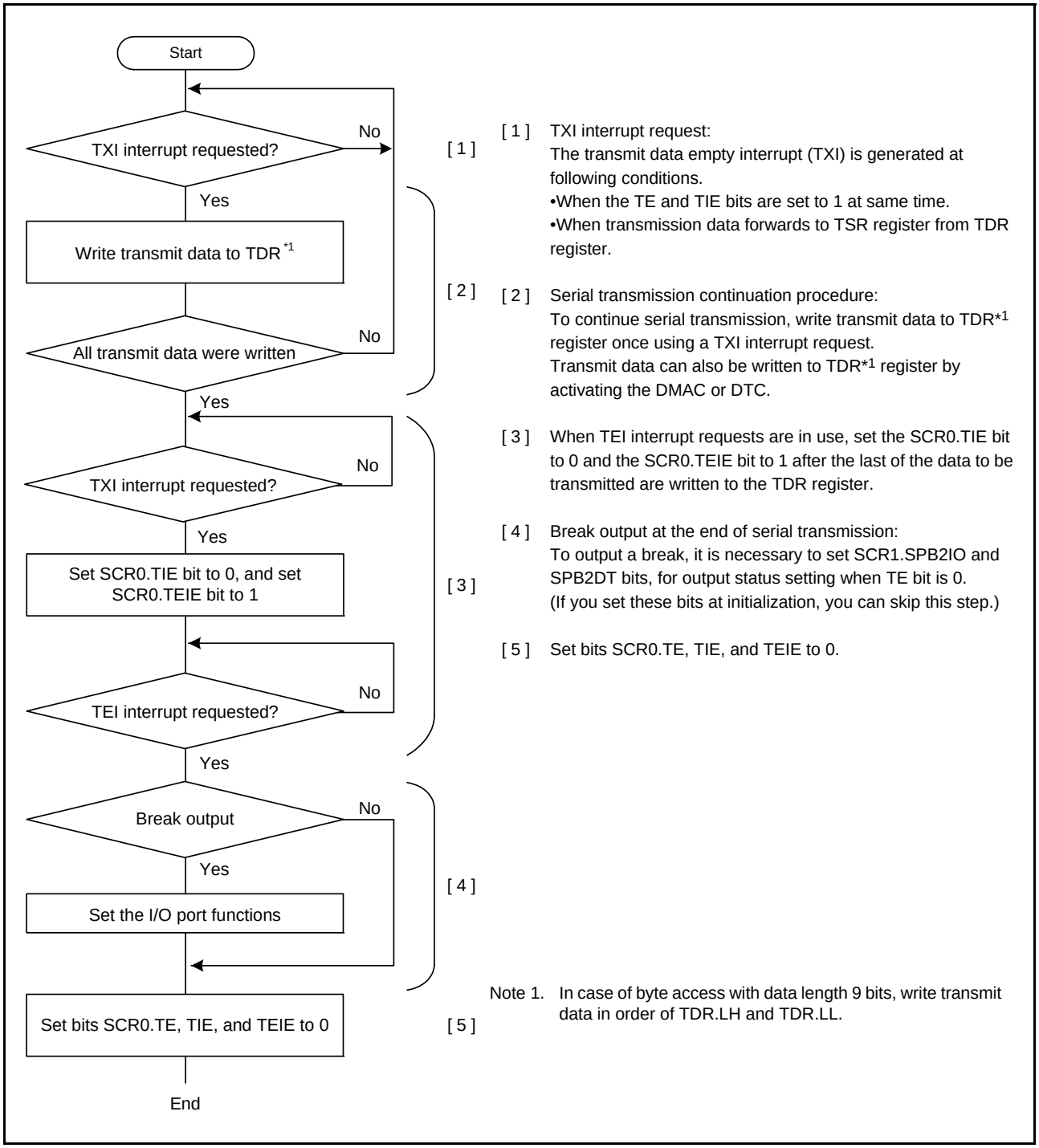


Figure 36.15 Example of Serial Transmission Flowchart in Asynchronous Mode (Non-FIFO Mode)

(2) FIFO Mode

Table 36.28 shows an example of data format that is written to transmit FIFO (TDR register) in asynchronous mode with FIFO selected.
MPBT bit write to transmit FIFO (TDR register) bit9. Data is set to TDR.TDAT[8:0] bits corresponded to data length. It should write to 0 for unused bits. It should write it in order of the TDR.LH and the TDR.LL at byte access.

Table 36.28 Data Format That is Written to Transmit FIFO (TDR) (FIFO Mode)

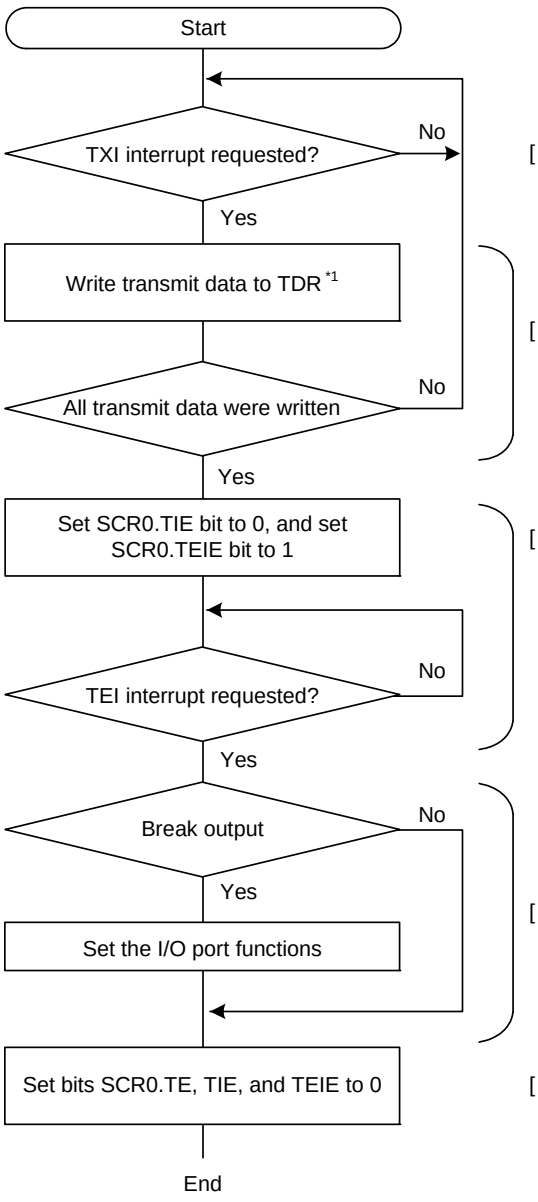
Data Length	Register setting		Transmit Data in TDR.L															
	SCR3	CHR[1:0]	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	1	—	—	—	—	—	—	MPBT	—	—	TDAT[6:0]						
8 bits	1	0	—	—	—	—	—	—	MPBT	—	TDAT[7:0]							
9 bits	0	0 or 1	—	—	—	—	—	—	MPBT	TDAT[8:0]								

—: Do not used. It should write to 0.

In serial transmission, the RSCI operates as described below.

1. The RSCI transfers data from transmit FIFO (TDR register) to TSR register when data is written to transmit FIFO (TDR register) in the TXI interrupt handling routine. The writable transmit data number is until (32 – transmit FIFO (TDR register)) bytes. At the beginning of transmission, set 1 to SCR0.TE and SCR0.TIE bits simultaneously. Then the TXI interrupt request is generated.
2. Transmission starts at data transfer from transmit FIFO (TDR register) to TSR register when SCR1.CTSE bit = 0 (CTS function is disabled) or when the CTS# pin level is low. When the quantity of transmit data written in transmit FIFO (TDR register) is equal to or less than the specified transmit triggering number, SSR.TDRE flag is set to 1. If SCR0.TIE bit is 1 at this time, a TXI interrupt request is generated. Continuous transmission is obtainable by writing the next transmit data to transmit FIFO (TDR register) in the TXI interrupt handling routine before transmission of the current transmit data is completed. Write the last transmission data, then the last data transmit start, and TXI is generated. When TEI interrupt requests are in use, before the last data transmit end, set the SCR0.TIE bit to 0 (a TXI interrupt request is disabled) and the SCR0.TEIE bit to 1 (a TEI interrupt request is enabled) using the TXI interrupt handling routine.
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 RSCI checks whether non-transmitted data in transmit FIFO (TDR register) or not at the time of stop bit output.
5. When data is set to transmit FIFO (TDR register), setting of SCR1.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn# pin cause the next transfer of the next transmit data from transmit FIFO (TDR register) to TSR register and sending of the stop bit, after which serial transmission of the next frame starts.
6. If data is not set to transmit FIFO (TDR register), SSR.TEND flag is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output. If SCR0.TEIE bit is 1 at this time, a TEI interrupt request is generated.

Figure 36.16 shows a sample flowchart for serial transmission in asynchronous mode at FIFO selected.



[1] [1] TXI interrupt request:
The transmit data empty interrupt (TXI) is generated at following conditions.
•When the TE and TIE bits are set to 1 at same time.
•When transmission data forwards to TSR from the transmit FIFO (TDR register) and the number of transmit data written in the transmit FIFO is equal to or less than the transmit FIFO threshold.

[2] Serial transmission continuation procedure:
To continue serial transmission, write all transmit data to the transmit FIFO (TDR register)*1 by using a TXI interrupt and clear the SSR.TDRE flag to 0. (Transmit data can also be written to the transmit FIFO (TDR register)*1 by activating the DMAC or DTC. In this case, the TDRE flag is cleared automatically, so please don't operate the TDRE flag.)
The number of transmission data it's possible to write in is 32 (the number of stored transmit FIFO data).

[3] If you use TEI, set TIE bit = 0 and TEIE bit = 1 after writing the last transmit data to TDR*1 register.

[4] Break output at the end of serial transmission:
To output a break, it is necessary to set SCR1.SPB2IO and SPB2DT bits, for output status setting when TE bit is 0.
(If you set these bits at initialization, you can skip this step.)

[5] Set bits SCR0.TE, TIE, and TEIE to 0.

Note 1. In case of byte access with data length 9 bits, write transmit data in order of TDR.LH and TDR.LL.

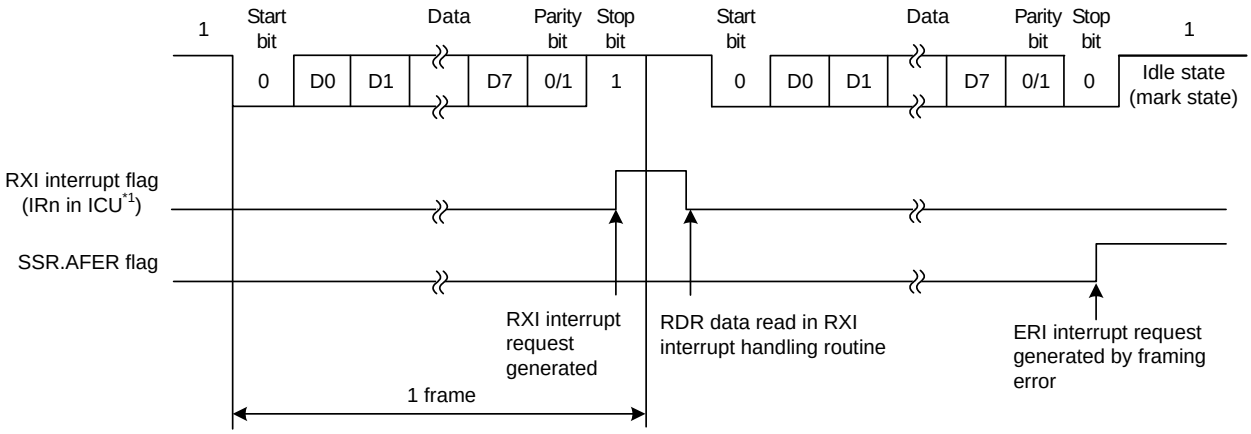
Figure 36.16 Example of Serial Transmission Flowchart in Asynchronous Mode (FIFO Mode)

36.3.9 Serial Data Reception (Asynchronous Mode)

(1) Non-FIFO Mode

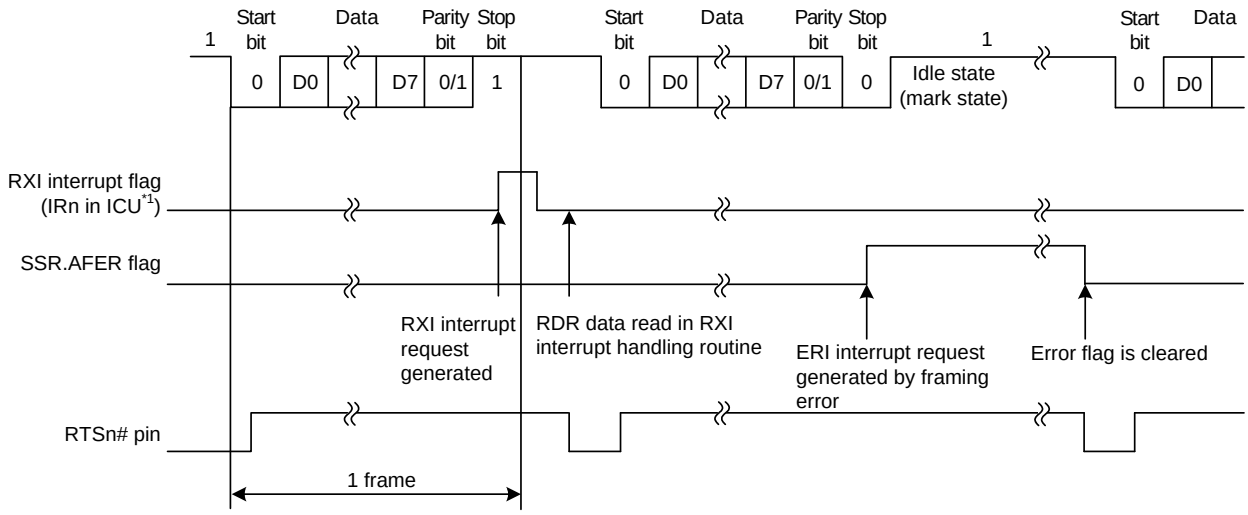
Figure 36.17 and Figure 36.18 show an example of the operation for serial data reception in asynchronous mode. In serial data reception, the RSCI operates as described below.

1. When the value of SCR0.RE bit becomes 1, the output signal on the RTSn# pin goes to the low level in the case of RTS function use.
2. When the RSCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in RSR register, and checks the parity bit and stop bit.
3. If an overrun error occurs, SSR.ORER flag is set to 1. If SCR0.RIE bit is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR register.
4. If a parity error is detected, SSR.APER flag is set to 1 and receive data is transferred to RDR register. If the SCR0.RIE bit is 1 at this time, an ERI interrupt request is generated.
5. If a framing error (when the stop bit is 0) is detected, SSR.AFER flag is set to 1 and receive data is transferred to RDR register. If SCR0.RIE bit is 1 at this time, an ERI interrupt request is generated.
6. When reception finishes successfully, receive data is transferred to RDR register. If SCR0.RIE bit is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to RDR register in this RXI interrupt handling routine before reception of the next receive data is completed. Reading the received data that have been transferred to RDR register causes the RTSn# pin to output the low level in the case of RTS function use. If you do not want to turn the RTSn # pin output low after receiving the last data, set SCR0.RE bit to 0, before reading the RDR register.



Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 36.17 Example of RSCI Operation for Serial Reception in Asynchronous Mode (1) (8-Bit Data, Parity, 1 Stop Bit, RTS Function is Not Used)



Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 36.18 Example of RSCI Operation for Serial Reception in Asynchronous Mode (2) (8-Bit Data, Parity, 1 Stop Bit, RTS Function is Used)

Table 36.29 lists the states of the flags in SSR status register and receive data handling when a receive error is detected. If a receive error is detected, an ERI interrupt request is generated but an RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, AFER, and APER flags to 0 before resuming reception. Moreover, be sure to read the RDR register during overrun error processing. When a reception is forcibly terminated by setting SCR0.RE bit to 0 during operation, read the RDR register because received data which has not yet been read may be left in RDR register.
Figure 36.19 and Figure 36.20 show samples of flowcharts for serial data reception.

Table 36.29 Flags in the SSR Status Register and Receive Data Handling

Flags in the SSR Status Register				
ORER	**AFER**	**APER**	**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

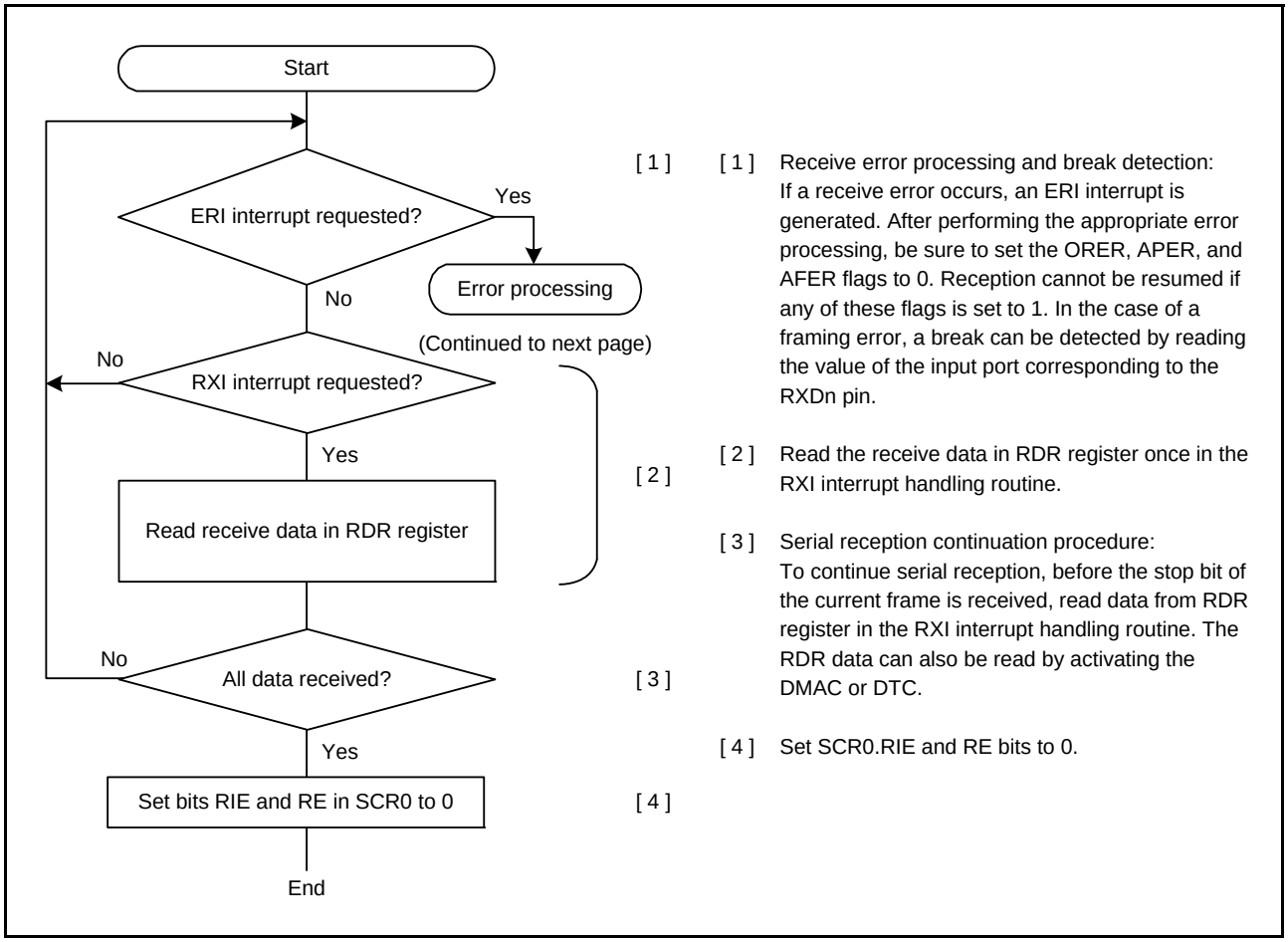


Figure 36.19 Example Flowchart of Serial Reception in Asynchronous Mode (Non-FIFO Mode) (1)

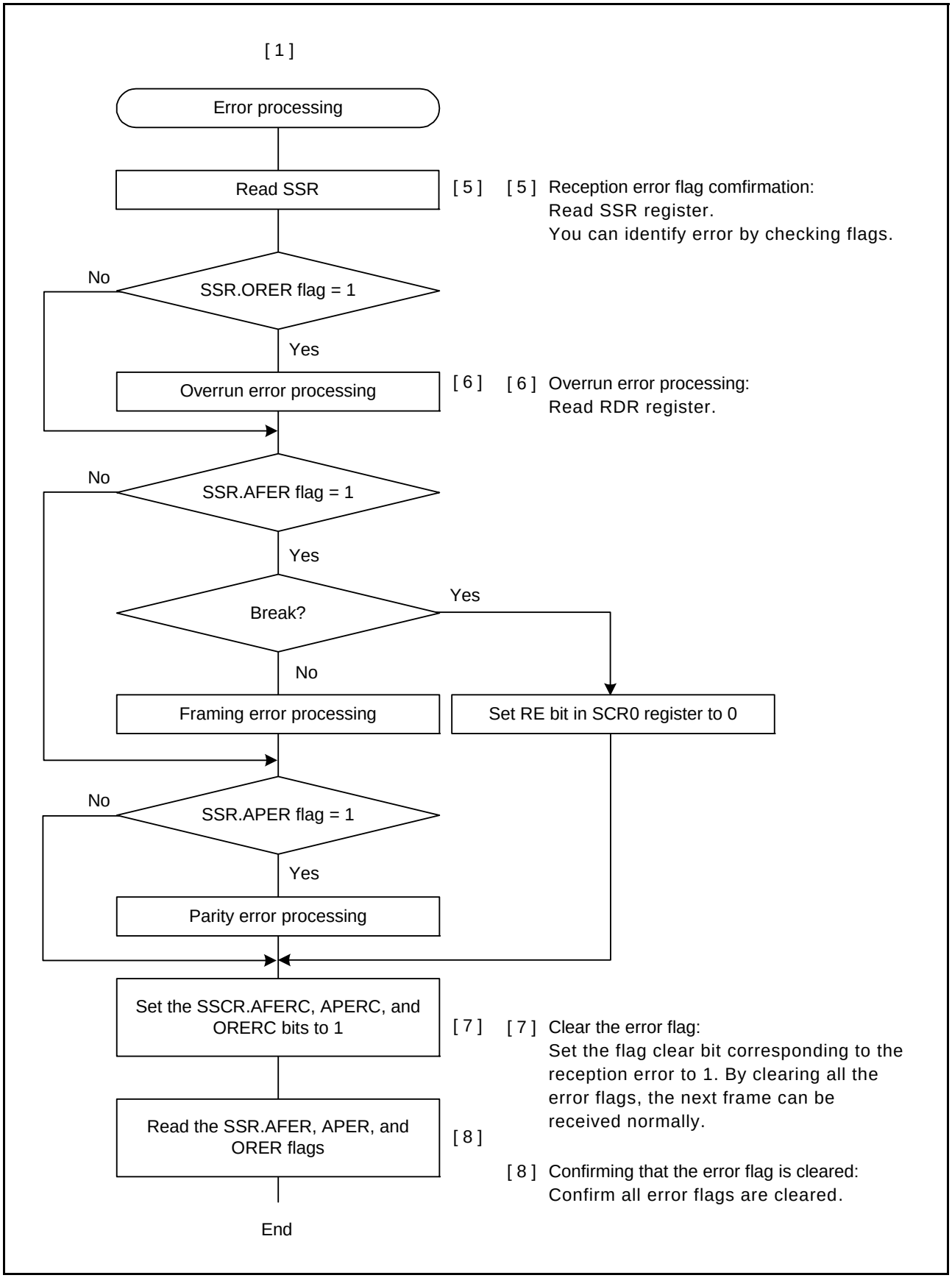


Figure 36.20 Example Flowchart of Serial Reception in Asynchronous Mode (Non-FIFO Mode) (2)

(2) FIFO Mode

Table 36.30 shows an example of data format stored to receive FIFO (RDR register) in asynchronous mode.
MPB flag (Receive FIFO (RDR register) bit9) is stored 0. Data is stored to receive FIFO (RDR register) corresponded to data length. It is stored to 0 for unused bits. If receive FIFO (RDR register) is read, RSCI updates to next data which are FER flag, PER flag, and receive data (RDAT[8:0] bits) in receive FIFO. The flags which are AFER, APER, ORER, and DR flag in receive FIFO, are always indicated to the flags corresponded to SSR register and RFSR register.

Table 36.30 Data Format Stored in the Receive FIFO (RDR) (FIFO Mode)

Data Length	Register setting		Arrangement of Receive Flag, MPB Flag and Received Data in RDR Register															
	SCR3	CHR[1:0]	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	1	—	—	—	FER	PER	DR	MPB	0	0	RDAT[6:0]						
8 bits	1	0	—	—	—	FER	PER	DR	MPB	0	RDAT[7:0]							
9 bits	0	0 or 1	—	—	—	FER	PER	DR	MPB	RDAT[8:0]								
Data Length	**SCR3**	**CHR[1:0]**	**b31**	**b30**	**b29**	**b28**	**b27**	**b26**	**b25**	**b24**	**b23**	**b22**	**b21**	**b20**	**b19**	**b18**	**b17**	**b16**
7 bits	1	1	—	—	—	AFER	APER	—	—	ORER	—	—	—	—	—	—	—	—
8 bits	1	0	—	—	—	AFER	APER	—	—	ORER	—	—	—	—	—	—	—	—
9 bits	0	0 or 1	—	—	—	AFER	APER	—	—	ORER	—	—	—	—	—	—	—	—

Note: 0 is always read from the MPB flag (RDR register bit9).
When a 7-bit data length is selected, 0 is read from the RDAT[8:7] bits.
When a 8-bit data length is selected, 0 is read from the RDAT[8] bit.

Table 36.31 lists the states of the flags in SSR register status register and receive data handling when a receive error is detected in FIFO mode. Figure 36.21 and Figure 36.22 show samples of flowcharts for serial data reception in FIFO mode.
In serial data reception, the RSCI operates as described below.

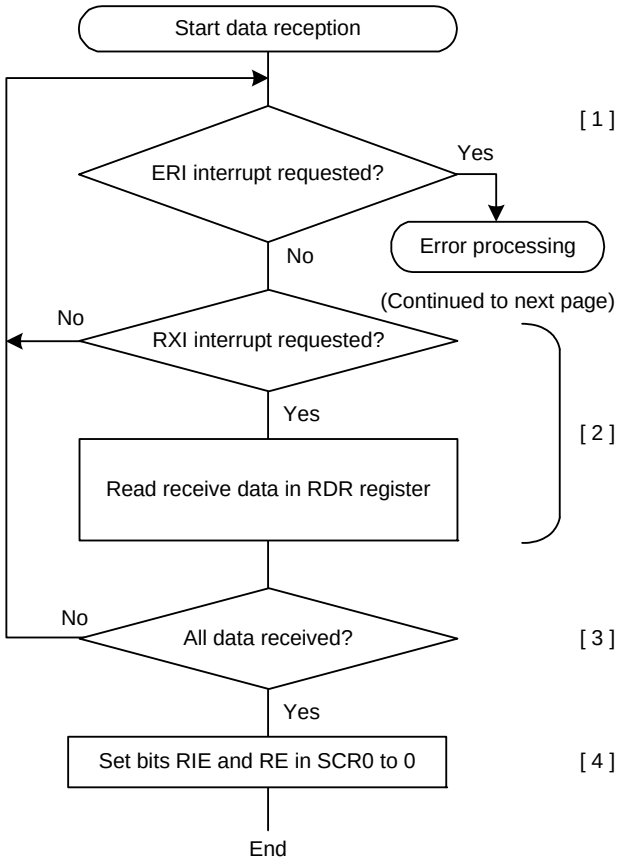
1. When the value of SCR0.RE bit becomes 1, the output signal on the RTSn# pin goes to the low level in the case of RTS function use.
2. When the RSCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in RSR, and checks the parity bit and stop bit.
3. If there is no space in receive FIFO (RDR register), an overrun error occurs and the SSR.ORER flag is set to 1. If SCR0.RIE bit is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to receive FIFO (RDR register).
4. If a parity error is detected, SSR.APER flag is set to 1 and receive data is transferred to receive FIFO (RDR register). If the SCR0.RIE bit is 1 at this time, an ERI interrupt request is generated.
5. If a framing error (when the stop bit is 0) is detected, SSR.AFER flag is set to 1 and receive data is transferred to RDR register. If SCR0.RIE bit is 1 at this time, an ERI interrupt request is generated.
6. After framing error is detected, when RSCI detects that continuous receive data is 0 for 1 frame, the reception stops.
7. When quantity of data stored in the receive FIFO data register (RDR register) falls the specified receive triggering number, and that no next data has been received yet after the elapse of 15 etu from the last stop bit in asynchronous mode, RFSR.DR flag is set to 1. If RIE bit is set to 1, RSCI occurs RXI interrupt request when FCR.DRES bit is 0 and RSCI occurs ERI interrupt request when FCR.DRES bit is 1.
8. When reception finishes successfully, receive data is transferred to receive FIFO (RDR register). The SSR.RDRF flag is set to 1 when the quantity of receive data which is equal to or greater than the specified receive triggering number are stored in receive FIFO (RDR register). If SCR0.RIE bit is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to receive FIFO (RDR register)

in this RXI interrupt handling routine, before overrun error is occurred. Reading the received data that have been transferred to receive FIFO (RDR register), and if it is less than RTS trigger number, causes the RTSn# pin to output the low level. in the case of RTS function use.

Table 36.31 Flags in the SSR Status Register and Receive Data Handling (FIFO Mode)

SSR Register				
ORER	**AFER*1**	**APER*1**	**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	0	0	Lost	Overrun error + framing error + parity error

Note 1. This flag indicates whether there is an error in received data when reception is completed.



[1] Receive error processing and break detection:
If a receive error occurs, an ERI interrupt is generated. After performing the appropriate error processing, be sure to set the ORER flag to 0. Reception cannot be resumed if ORER flag is set to 1. Reception can continue even if APER, AFER, and DR flags are 1.

[2] Read the receive data in the receive FIFO (RDR register) in the RXI interrupt handling routine. The reception data stored in the receive FIFO (RDR register) is read until the number of stored data will be below the FCR.RTRG value at least. You can confirm the number of the reception data in the FIFO by RFSR.R[5:0] bits.

[3] Serial reception continuation procedure:
To continue serial reception, before an overrun error occurs, please read data from the receive FIFO (RDR register) in the RXI interrupt handling routine and clear SSR.RDRF flag and RFSR.DR flag to 0. The receive FIFO data can also be read by activating the DMA or DTC. The RDRF flag is cleared automatically in this case, so please don't operate the RDRF flag.

[4] Set SCR0.RIE and RE bits to 0.

Figure 36.21 Example Flowchart of Serial Reception in Asynchronous Mode (FIFO Mode) (1)

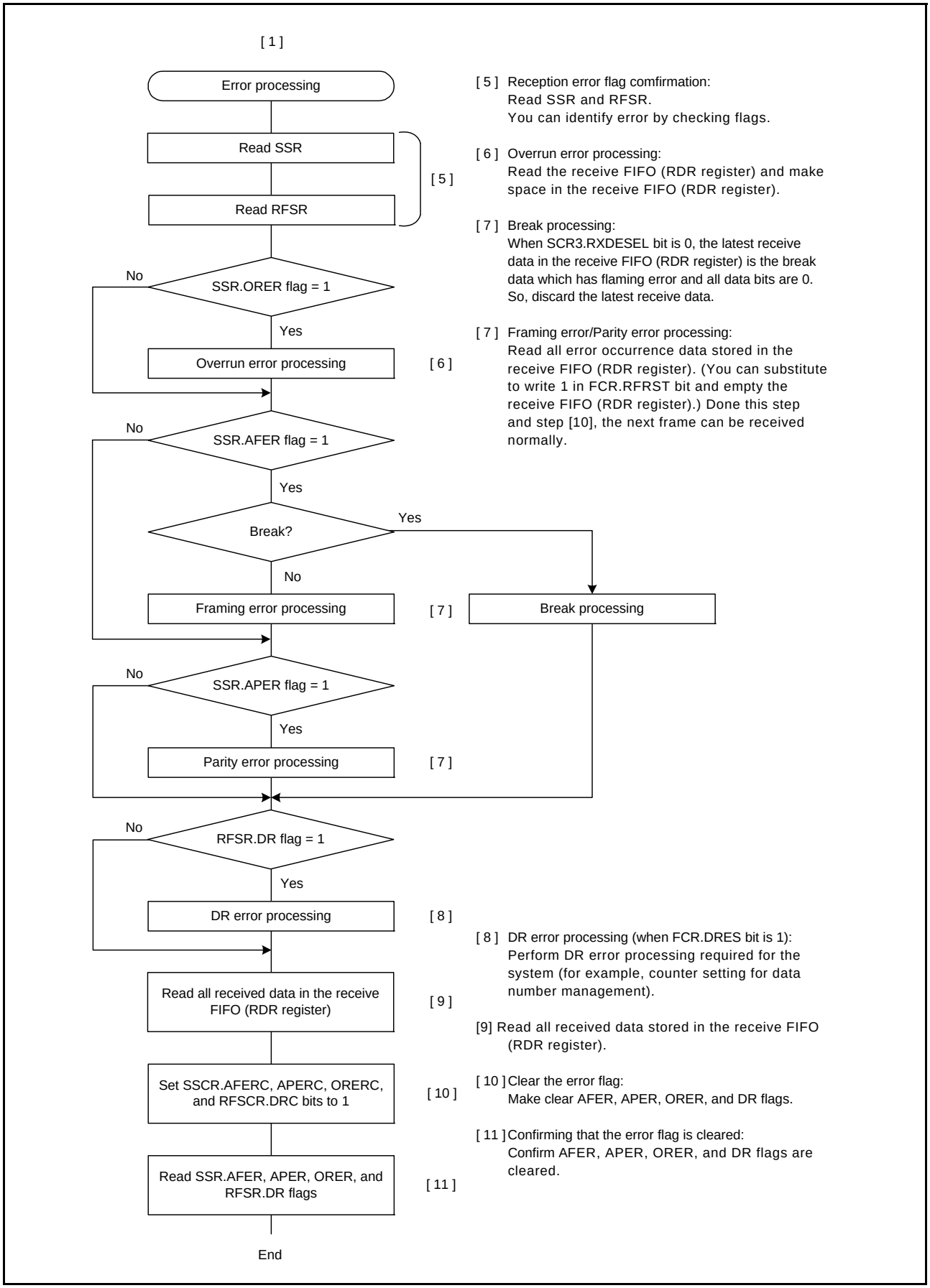


Figure 36.22 Example Flowchart of Serial Reception in Asynchronous Mode (FIFO Mode) (2)

36.3.10 Receive Data Sampling Timing Adjustment

When a waveform such that high width is different from low width because the difference between rising time and falling time is large is received, adjust the timing for data to be sampled at the center of the narrower pulse. When low width is narrower than high width, move the sampling timing forward, and when high width is narrower than low width, move the sampling timing backward.

When the SCR4.RTMG[3:0] bits are set to an offset to the default sampling point and then the SCR4.RTADJ bit is set to 1, received data is sampled at the specified position.

Figure 36.23 shows an example of the sampling timing adjustment.

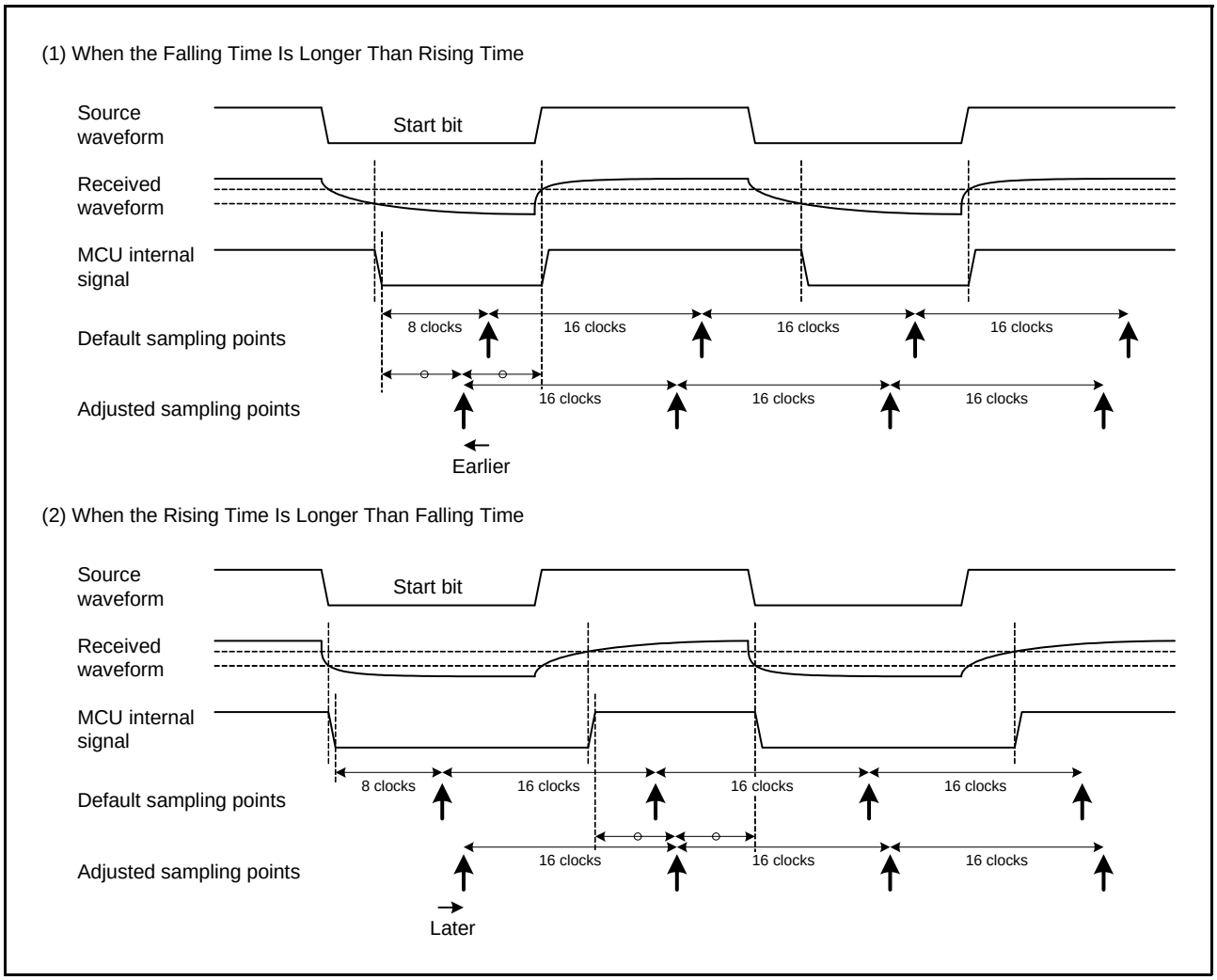


Figure 36.23 Example of Sampling Timing Adjustment (SCR2.ABCSE Bit = 0 and SCR2.ABCS Bit = 0)

36.3.11 Transmit Data Transition Timing Adjustment

When a difference between high width and low width for a waveform transmitted by this MCU arises at the receiver, it is also possible to make a difference between the high width and the low width beforehand at transmission to adjust so that the difference disappears at the receiver. When high width becomes narrower at the receiver, expand the high width of the transmit signal by delaying the falling edges, and when low width becomes narrower at the receiver, expand the low width of the transmit signal by delaying the rising edges.

When the SCR4.TTMG[3:0] bits are set to the transition direction and the delay amount and the SCR4.TTADJ bit is set to 1, transmit data changes at the specified point.

Figure 36.24 shows an example of the transition timing adjustment.

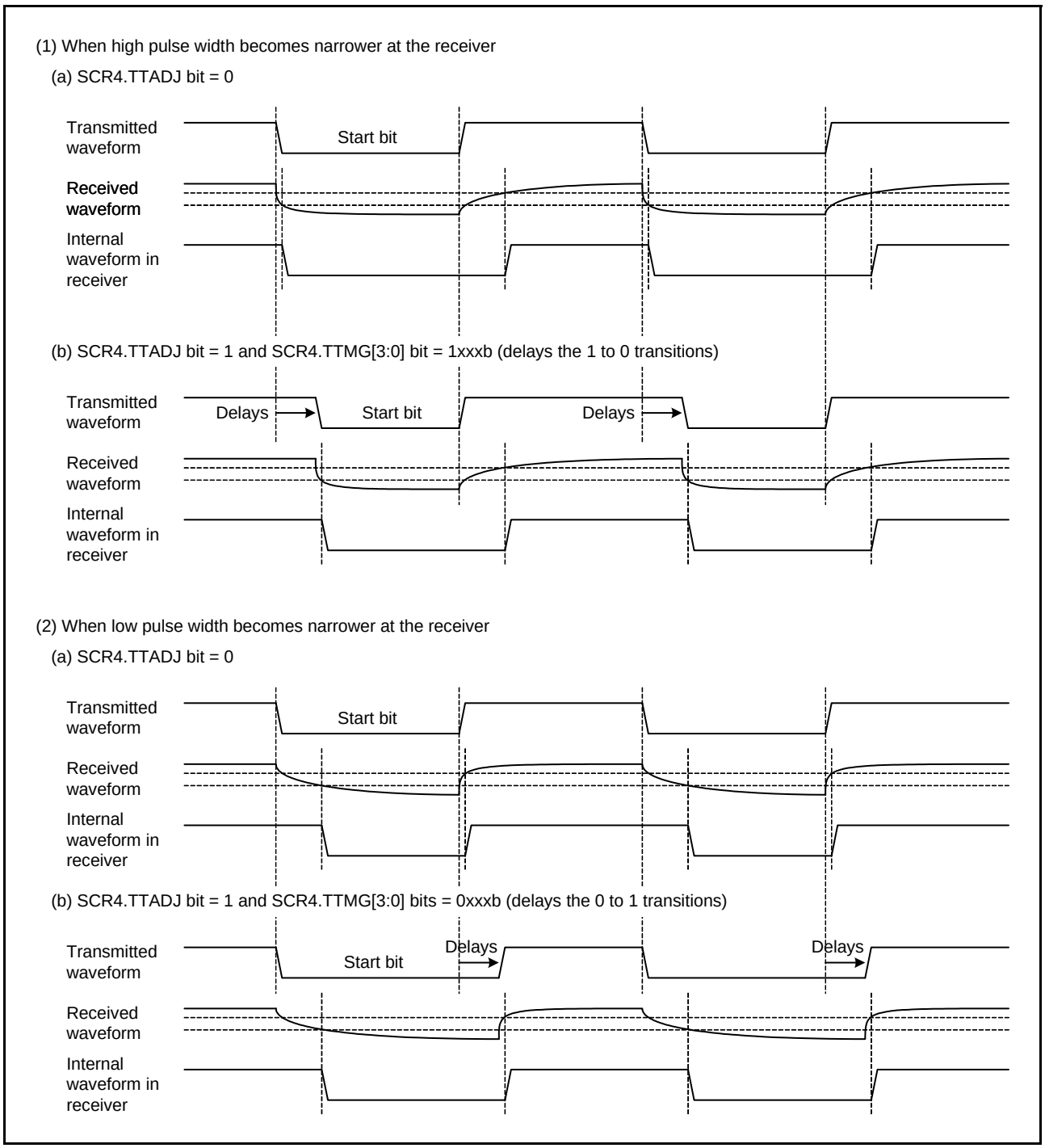


Figure 36.24 Example of Transition Timing Adjustment

36.4 Multi-Processor Communication 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 multiprocessor bit is set to 0, it indicates the data transmission cycle. Figure 36.25 shows an example of communication between processors by using a multi-processor format. First, a transmitting station transmits communication data in which the multi-processor bit set to 1 is added to the ID code of the receiving station. Next, the transmitting station transmits the communication data in which the multi-processor bit set to 0 is added to the transmit data. Upon receiving the communication data in which the multi-processor bit is set to 1, the receiving station compares the received ID with the ID of the receiving station itself and if the two matches, 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.
RTS control cannot be used at the time of multi-processor communication function use, because this is a function corresponding to one-to-many communications.

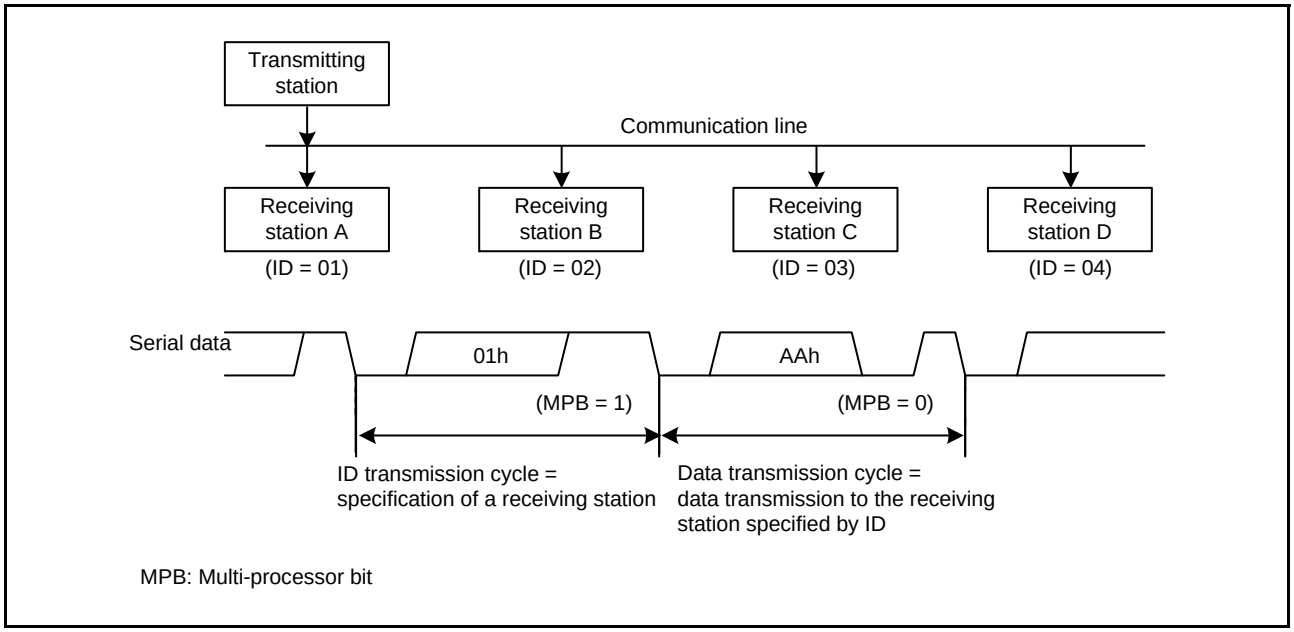


Figure 36.25 An Example of Communication using the Multi-Processor Format (Example of Transmission of Data AAh to Receiving Station A)

(1) Non-FIFO Mode

For supporting this function, the RSCI provides the MPIE bit in SCR0 register. When the MPIE bit is set to 1, transfer of receive data from the RSR register to the RDR register, detection of a receive error, and setting the respective status flags RDRF, ORER and AFER 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 MPB flag in RDR register is set to 1 and the MPIE bit in SCR0 register is automatically cleared, thus returning to a non-multi-processor reception operation. At this time, an RXI interrupt is generated if the RIE bit in SCR0 register 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 non-multi-processor asynchronous mode. A clock which is used for the multi-processor communication is also the same as the clock used in the non-multi-processor asynchronous mode.

(2) FIFO Mode

For transmission, SW should write to TDR.MPBT (Multi-Processor Bit Transfer) which corresponds to transmit data in TDR.TDAT[8:0] bits. For reception, multi-processor bit that is a part of receive data, is stored to RDR.MPB flag, and receive data is stored to RDR.RDAT[8:0] bits. When the MPIE bit is set to 1, transfer of receive data from the RSR to the RDR.RDAT[8:0] bits, detection of a receive error, and detection of DR flag, and setting the respective status flags RDRF, ORER, and AFER 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 MPB flag in RDR register is set to 1,and receive data is stored to receive FIFO (RDR.RDAT[8:0] bits), and the MPIE bit in SCR0 register is automatically cleared, thus returning to a non-multi-processor reception operation. At this time, an RXI interrupt is generated if the RIE bit in SCR0 register 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 non-multi-processor asynchronous mode and FIFO mode.

36.4.1 Multi-Processor Serial Data Transmission

(1) Non-FIFO Mode

Figure 36.26 shows a sample flowchart of multi-processor data transmission. In the ID transmission cycle, the ID should be transmitted with the MPBT bit in TDR register 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.

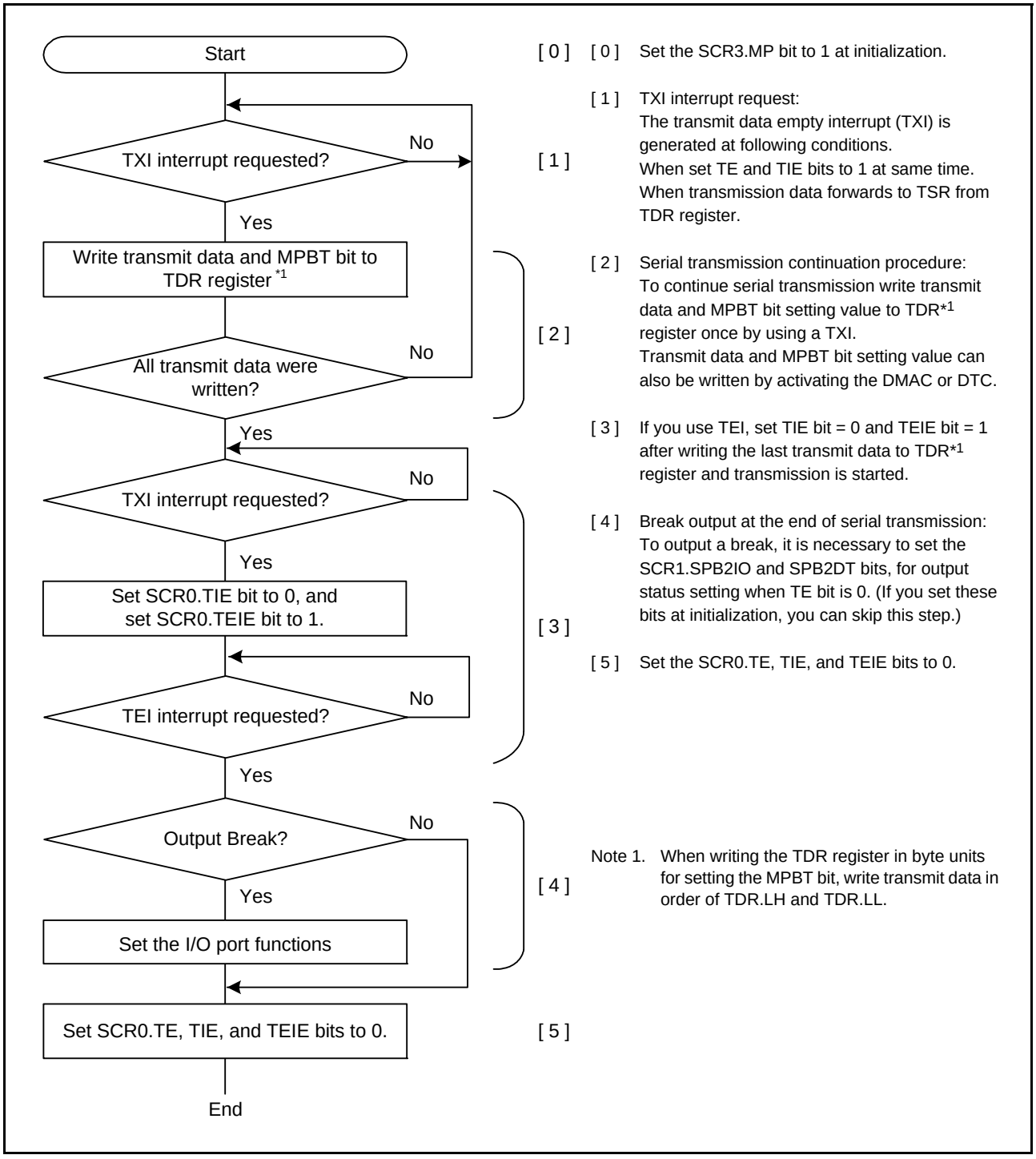


Figure 36.26 Example of Multi-Processor Serial Transmission Flowchart (Non-FIFO Mode)

(2) FIFO Mode

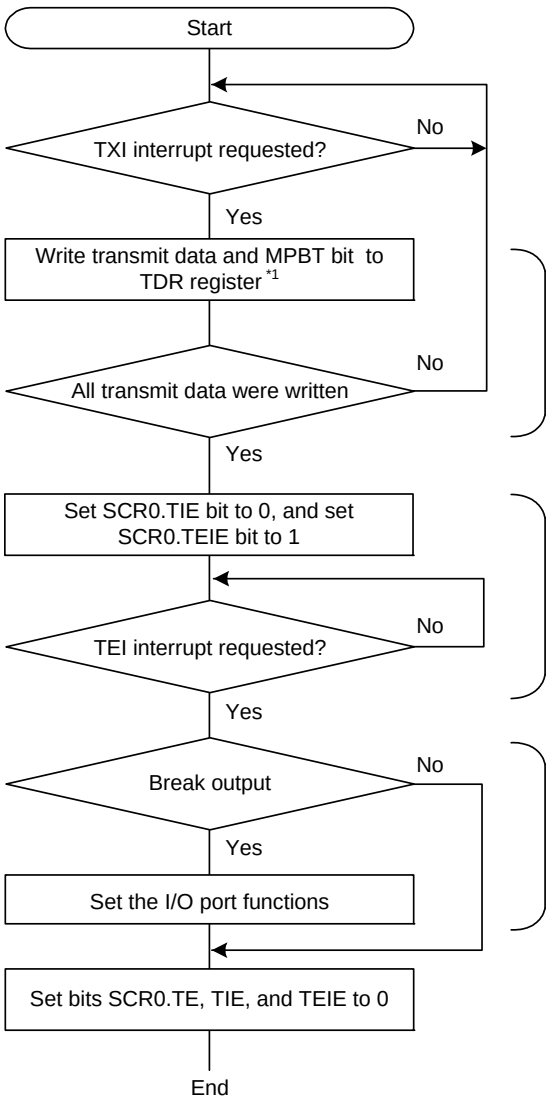
Table 36.32 shows an example of data format that is written to transmit FIFO (TDR register) in multi-processor mode. Write MPBT in bit9 of TDR. And write data to transmit FIFO (TDR register) corresponded to data length. It should write to 0 for unused bits.

Table 36.32 Data Format in Multi-processor Mode that is Written to Transmit FIFO (TDR Register)

Data Length	Register setting		Transmit Data in TDR.L															
	SCR3	CHR[1:0]	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	1	—	—	—	—	—	—	MPBT	—	—	TDAT[6:0]						
8 bits	1	0	—	—	—	—	—	—	MPBT	—	TDAT[7:0]							
9 bits	0	0 or 1	—	—	—	—	—	—	MPBT	TDAT[8:0]								

—: Do not used. It should write to 0.

Figure 36.27 shows a sample flowchart for multi-processor data transmission in FIFO mode. In the ID transmission cycle, the ID should be transmitted with the MPBT bit in TDR register 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 with FIFO enabled.



[0] Set the SCR3.FM and MP bits to 1 at initialization.

[1] TXI interrupt request:
The transmit data empty interrupt (TXI) is generated when transmission data forwards to TSR from the transmit FIFO (TDR register) and the number of transmit data written in the transmit FIFO is equal to or less than the transmit FIFO threshold.

[2] Serial transmission continuation procedure:
To continue serial transmission, write all transmit data and MPBT bit setting value to the transmit FIFO (TDR register) by using a TXI interrupt. Transmit data and MPBT bit setting value can also be written to the transmit FIFO (TDR register) by activating the DMAC or DTC.

[4] If you use TEI, set TIE bit = 0 and TEIE bit = 1 after writing the last transmit data to TDR*1 register.

[5] Break output at the end of serial transmission:
To output a break, it is necessary to set the SCR1.SPB2IO and SPB2DT bits, for output status setting when TE bit is 0.
(If you set these bits at initialization, you can skip this step.)

[6] Set bits SCR0.TE, TIE, and TEIE to 0.

Note 1. When writing the TDR register in byte units for setting the MPBT bit, write transmit data in order of TDR.LH and TDR.LL.

Figure 36.27 Example of Multi-Processor Serial Transmission Flowchart (FIFO Mode)

36.4.2 Multi-Processor Serial Data Reception

(1) Non-FIFO Mode

Figure 36.29 and Figure 36.30 are sample flowcharts of multi-processor data reception. When the MPIE bit in SCR0 register 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 register. At this time, the RXI interrupt request is generated. The other operations are the same as the operations in asynchronous mode.
Figure 36.28 is the example of operation for reception.

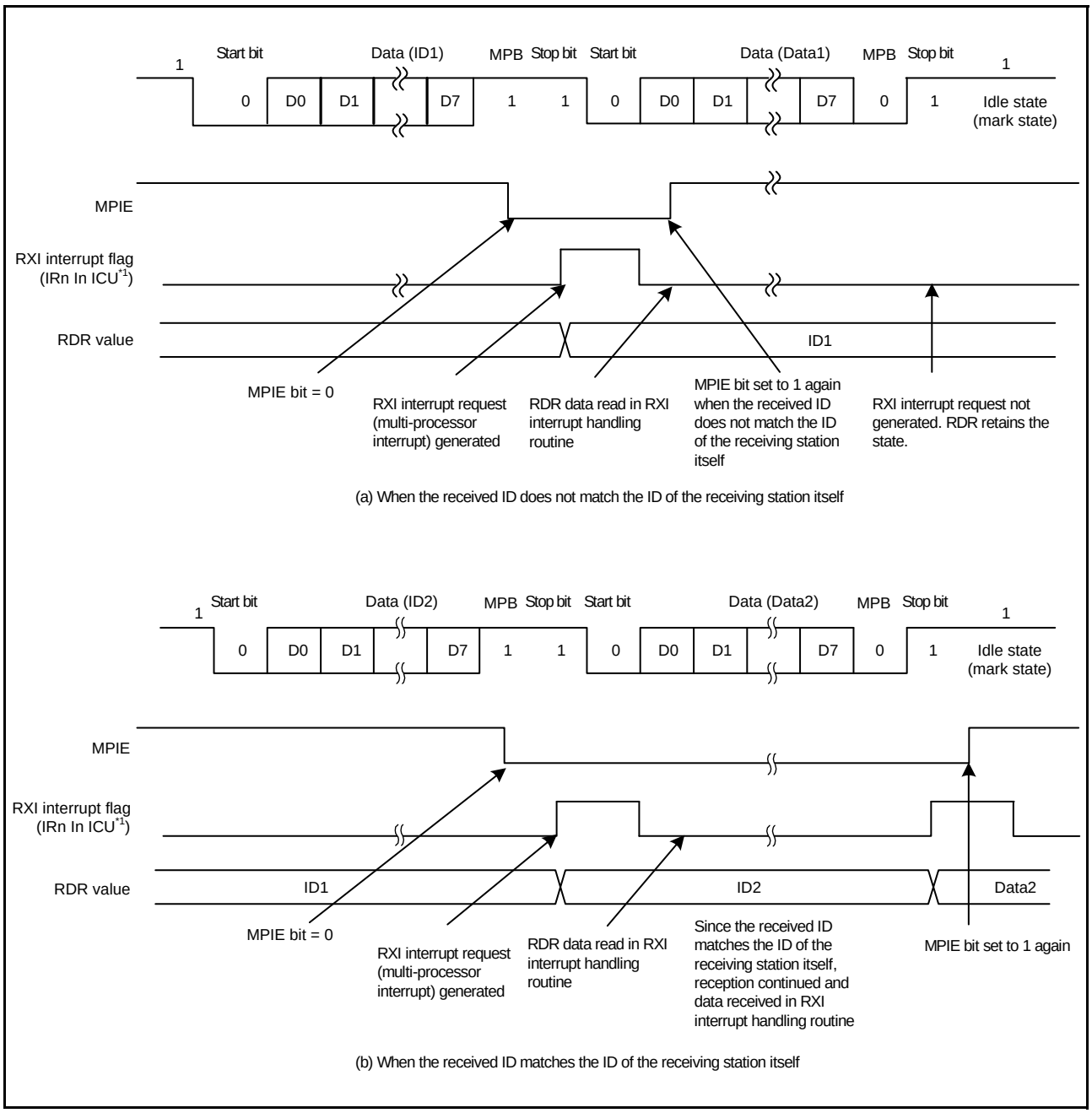
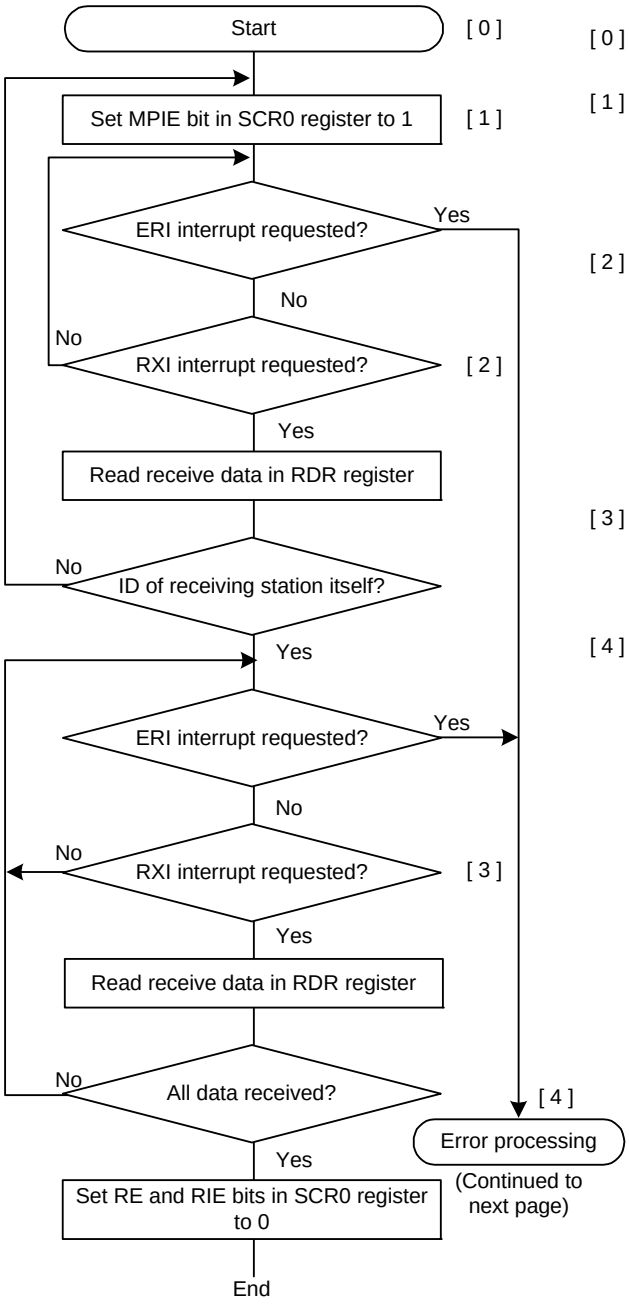


Figure 36.28 Example of RSCI Reception (8-Bit Data/Multi-Processor Bit/1 Stop Bit)



[0] Set the SCR3.MP bit to 1 at initialization.

[1] ID reception cycle:
Set SCR0.MPIE bit to 1 and wait for ID reception.
If you set this bits at initialization, you can skip this Step.

[2] RSCI status confirmation and reception and comparison of ID:
Read data in RDR register at the first RXI interrupt and compare it with the ID of the receiving station itself. If the ID does not match the ID of the receiving station itself, set the MPIE bit to 1 again, and wait for another RXI interrupt request.

[3] Data reception at an RXI interrupt:
Read data in RDR register once in the RXI interrupt routine.

[4] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading ORER and AFER flags in SSR. After performing the appropriate error processing be sure to clear the ORER and AFER flags. Reception cannot be resumed if any of these flags is set to 1.
In the case of a framing error, a break can be detected by reading SSR.RXDMON flag.

Figure 36.29 Example of Multi-Processor Serial Reception Flowchart (1) (Non-FIFO Mode)

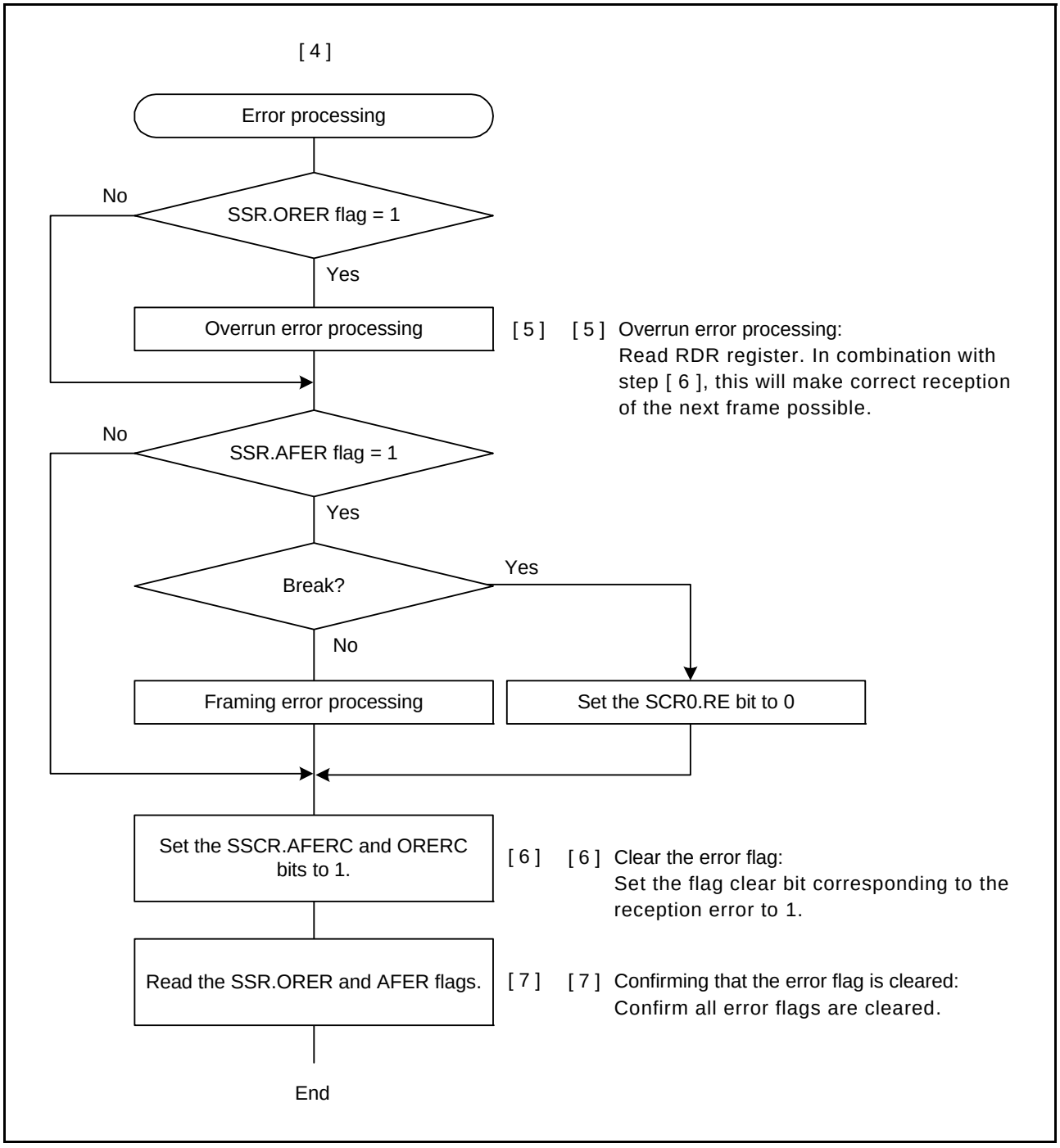


Figure 36.30 Example of Multi-Processor Serial Reception Flowchart (2) (Non-FIFO Mode)

(2) FIFO Mode

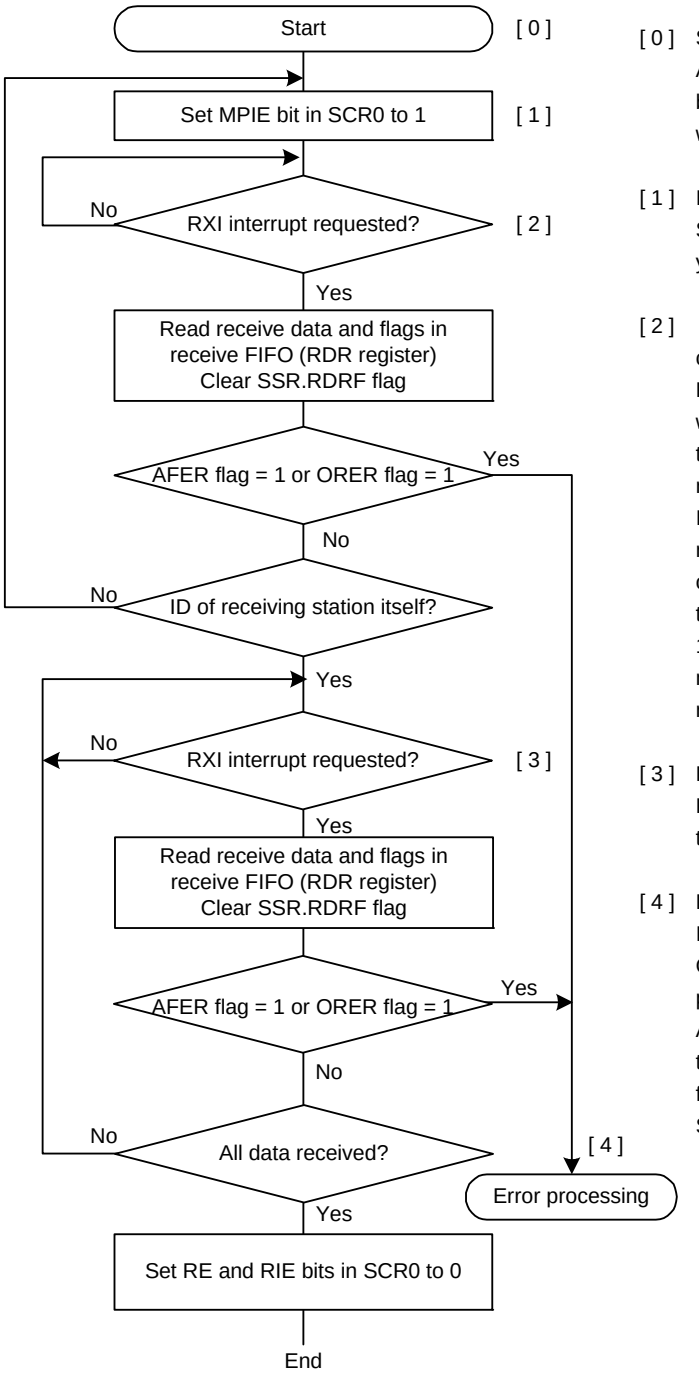
Table 36.33 shows an example of data format that is stored to receive FIFO (RDR register) in multi-processor mode. MPB flag is stored in bit9 of RDR register. 0 is stored to APER and PER flags. Data is stored to receive FIFO RDAT[8:0] bits corresponded to data length. It is stored to 0 for unused bits. Reading the receive FIFO (RDR register) updates the FER, PER, MPB flags and receive data (RDAT[8:0] bits) in the receive FIFO (RDR register) with the next received data. The AFER, APER, and ORER flags in the receive FIFO (RDR register) always reflect the status of the corresponding flags in the SSR and RFSR registers.

Table 36.33 Data Format in Multi-Processor Mode that is Stored to Receive FIFO (FIFO Mode)

Data Length	Register setting SCR3 CHR[1:0]		Received Data in RDR Register b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	1	0	0	0	FER	PER	DR	MPB	0	0	RDAT[6:0]						
8 bits	1	0	0	0	0	FER	PER	DR	MPB	0	RDAT[7:0]							
9 bits	0	0 or 1	0	0	0	FER	PER	DR	MPB	RDAT[8:0]								
Data Length	**SCR3 CHR[1:0]**		**b31**	**b30**	**b29**	**b28**	**b27**	**b26**	**b25**	**b24**	**b23**	**b22**	**b21**	**b20**	**b19**	**b18**	**b17**	**b16**
7 bits	1	1	0	0	0	AFER	APER	0	0	ORER	0	0	0	0	0	0	0	0
8 bits	1	0	0	0	0	AFER	APER	0	0	ORER	0	0	0	0	0	0	0	0
9 bits	0	0 or 1	0	0	0	AFER	APER	0	0	ORER	0	0	0	0	0	0	0	0

Note: When data length is 7bit, it can read always 0 in RDAT[8:7] bits.
When data length is 8bit, it can read always 0 in RDAT[8] bit.

Figure 36.31 shows a sample flowchart for multi-processor data reception in FIFO mode. When the MPIE bit in SCR0 register is set to 1, reading the communication data is skipped until reception of the communication data in which the multiprocessor bit is set to 1. When the communication data in which the multi-processor bit is set to 1 is received, the received data, MPB flag and each errors are transferred to receive FIFO (RDR register), and the MPIE bit in SCR0 register is automatically cleared, thus returning to a normal reception operation. After a framing error occurred and SSR.AFER flag is set to 1, but RSCI continues data reception.
The other operations are the same as the operations in asynchronous mode with FIFO enabled.



[0] Set the SCR3.FM and MP bits to 1.
Also, set the FCR.RTRG[4:0] bits to 00001b and DRES bit to 0 to generate an RXI interrupt request when data with the MPB flag set to 1 is received.

[1] ID reception cycle:
Set SCR0.MPIE bit to 1. And wait for ID reception. If you set this bits at initialization, you can skip this Step.

[2] RSCI status confirmation and reception and comparison of ID:
RSCI skips the received data until it receives the data with the MPB flag set to 1. When receiving data with the MPB flag set to 1, RSCI generates an RXI interrupt request.
In the RXI interrupt routine, read the data from the receive FIFO (RDR register) and compare it with the ID of the receiving station itself. If the ID does not match the ID of the receiving station itself, set the MPIE bit to 1 again and wait for another RXI interrupt request. If it matches the ID of the receiving station itself, set the receive FIFO threshold to the desired value.

[3] Data reception at an RXI interrupt:
Read data in the receive FIFO (RDR register) once in the RXI interrupt routine.

[4] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading ORER and AFER flags in SSR register (There is no problem reading the SSR.ORER and AFER flags again.). After performing the appropriate error processing be sure to clear the ORER and AFER flags. In the case of a framing error, a break can be detected by reading the SSR.RXDMON flag.

Figure 36.31 Example Flowchart of Serial Reception in Multi-Processor Mode (1) (FIFO Mode)

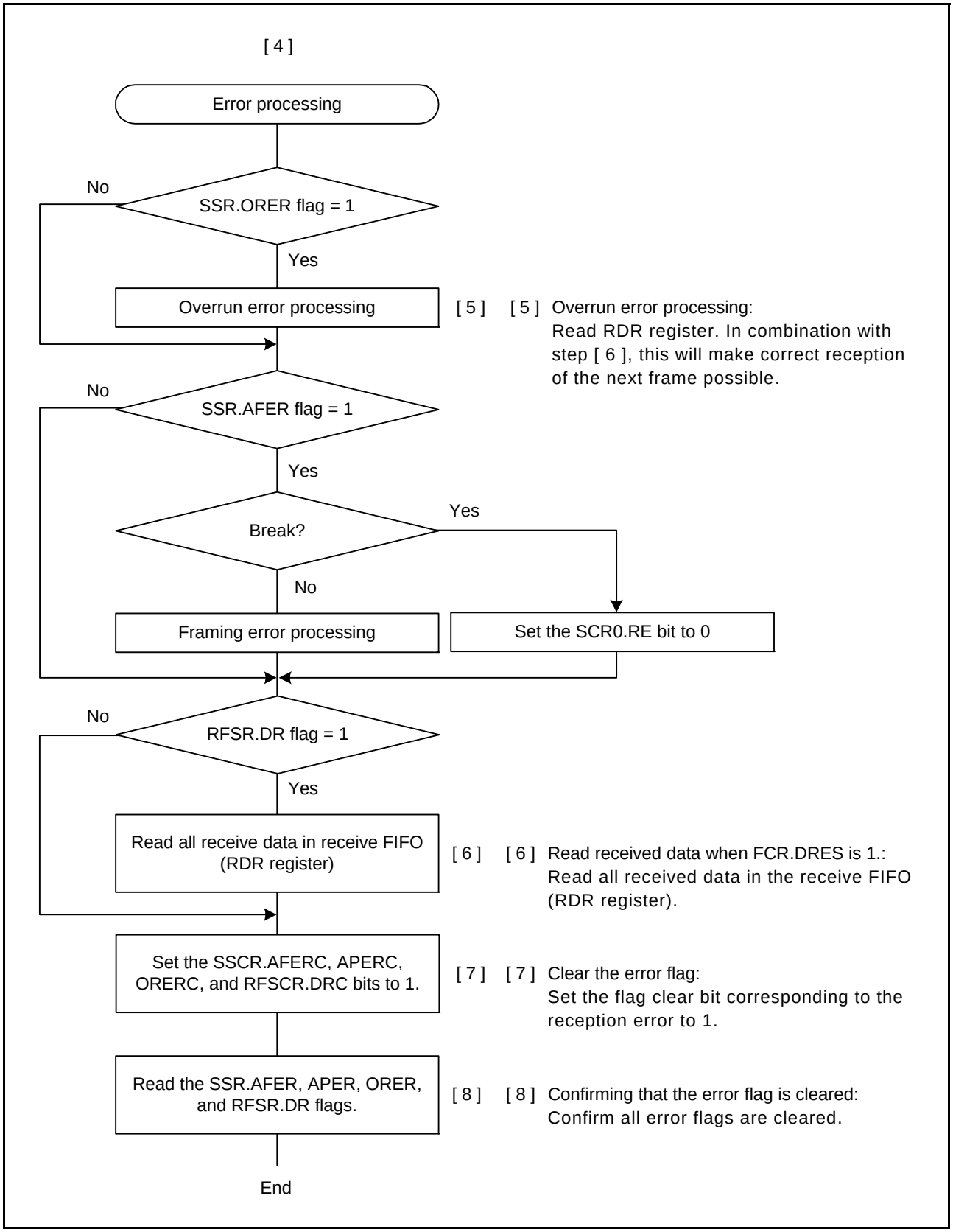


Figure 36.32 Example Flowchart of Serial Reception in Multi-Processor Mode (2) (FIFO Mode)

36.5 Manchester Mode

In manchester mode, the transmit or receive serial data is coded in Manchester encoding.
Figure 36.33 shows the conceptual image of Manchester encoding.

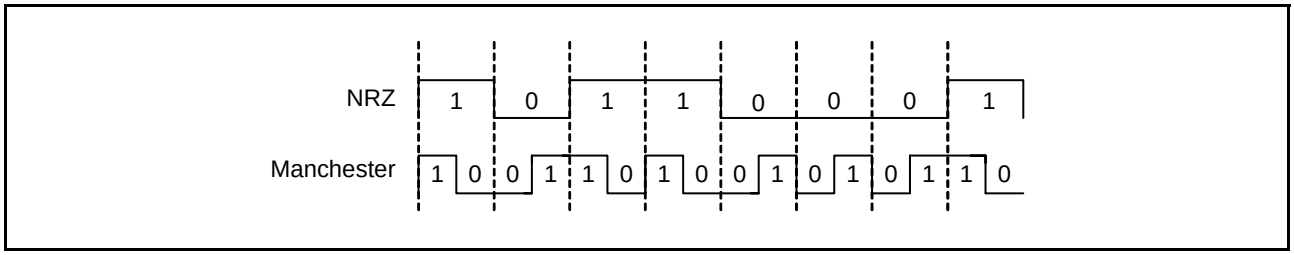


Figure 36.33 Example of Manchester Encoding

In manchester mode, a preface and a start bit area are added to the transmit data in the register to configure a transmit frame. For transmission, data is encoded in Manchester encoding. When data is received, frames having the same format as transmitted frames are detected and Manchester decoding is performed.
For details on the frame format, see section 36.5.1, Frame Format.

36.5.1 Frame Format

Figure 36.34 shows the frame format in manchester mode.
In the upper half of the figure, relevant setting registers are shown.
The preface area and the data area are encoded in Manchester encoding.

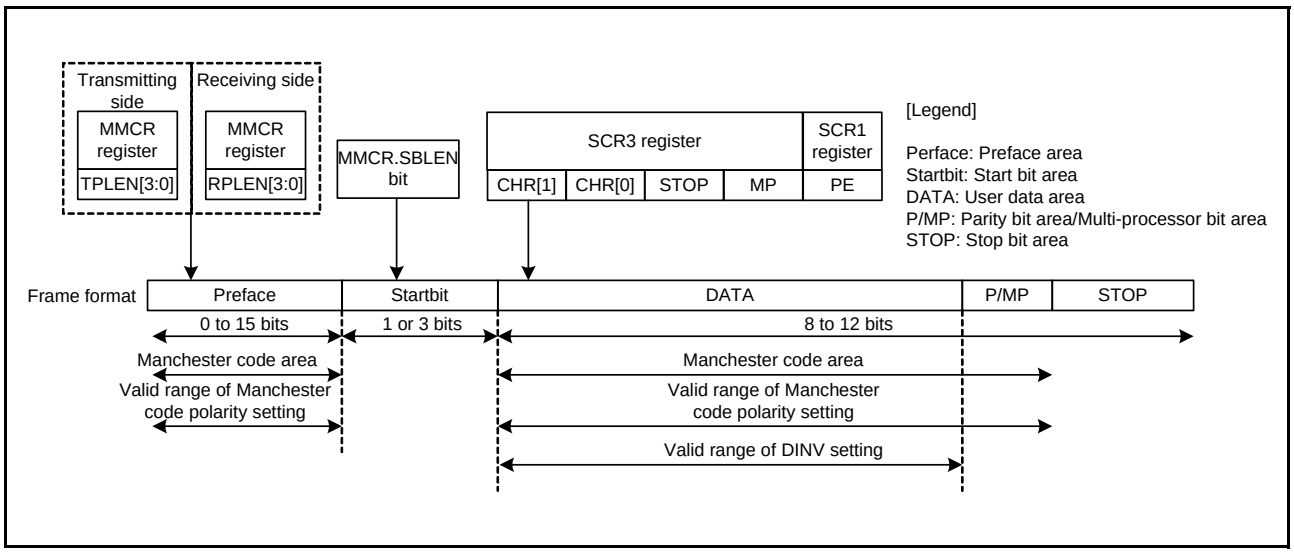


Figure 36.34 Frame Format in Manchester Mode

(1) Preface Area

This is a fixed pattern area located at the beginning of each frame.
Different registers are used to set the preface area for transmission and reception. The preface length is determined by setting MMCR.TPLEN[3:0] bits for transmission. It is determined by setting MMCR.RPLEN[3:0] bits for reception.
If it is set to 0, the transmit preface is disabled and is not added.
If it is set to 1d to 15d, a preface whose length is determined by this setting is added.
(For example, if it is set to 1d, a 1-bit preface is added. If it is set to 15d, a 15-bit preface is added.)
In addition, the preface pattern can be changed by setting, and it can be selected from four types of patterns by setting

MMCR.TPPAT[1:0] bits for transmission and MMCR.RPPAT[1:0] bits for reception.

Figure 36.35 shows how the preface pattern is set. The preface area and the start bit area are added for each communication frame.

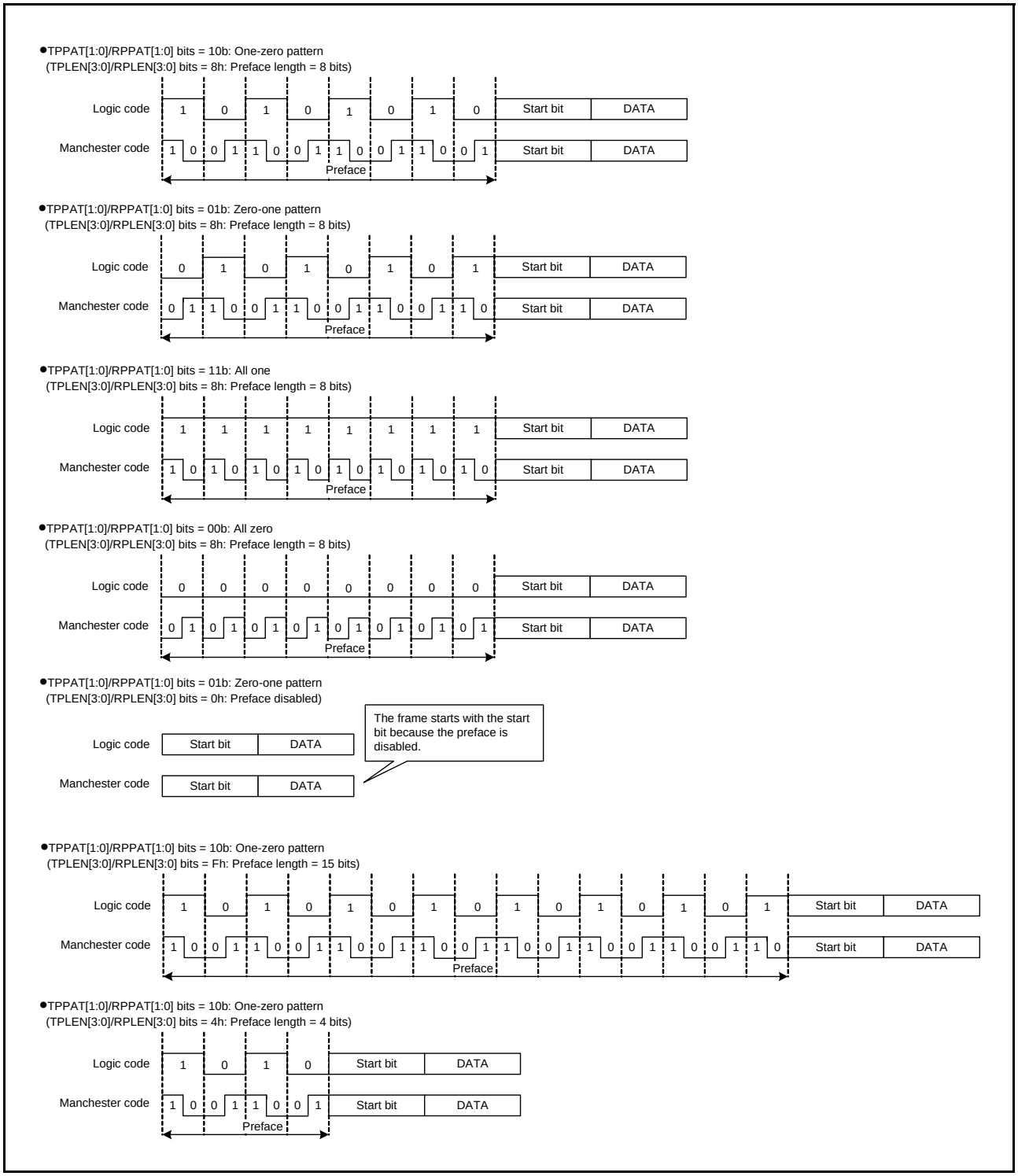
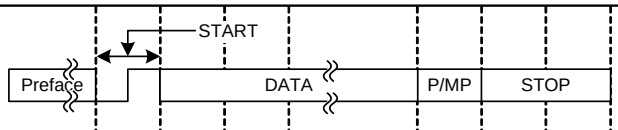
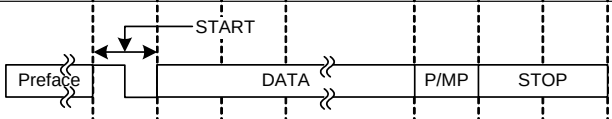
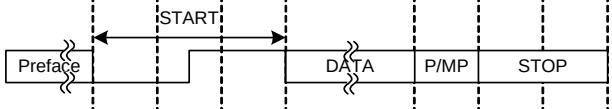
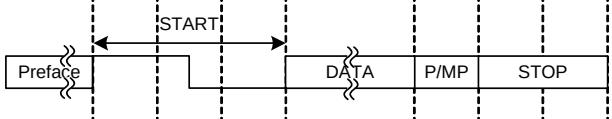
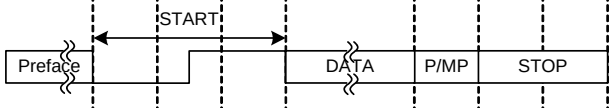
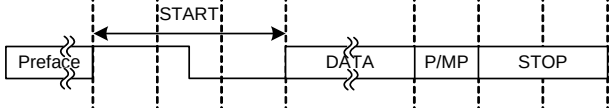


Figure 36.35 Preface Pattern Setting Example

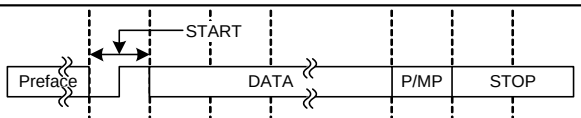
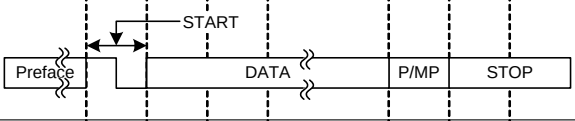
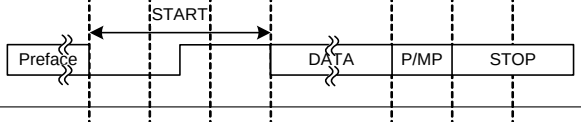
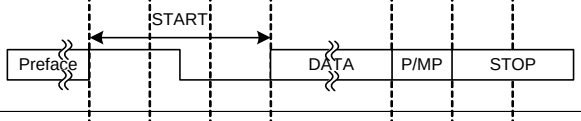
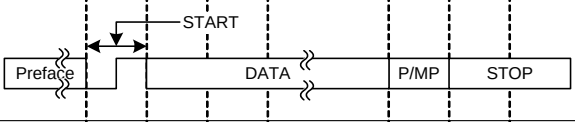
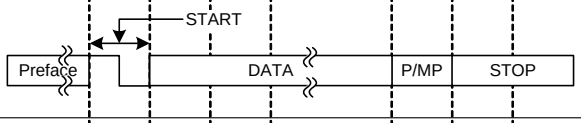
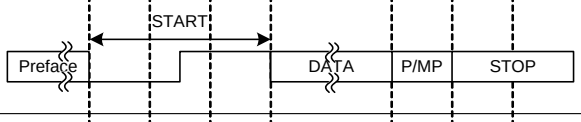
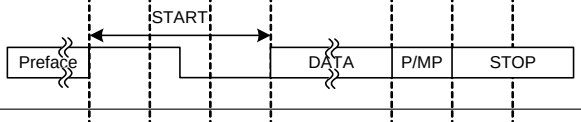
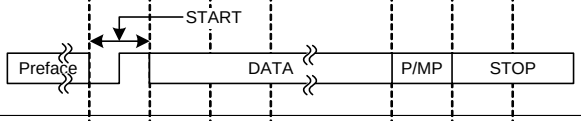
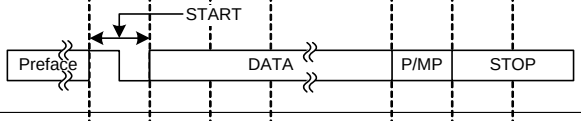
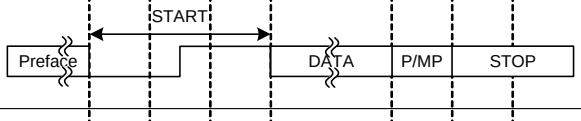
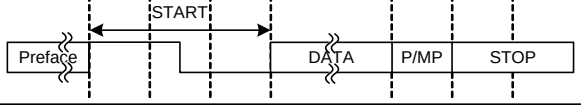
(2) Start Bit Area

This is an area indicating the start of valid data in a frame. It is added after the preface area.
The start bit length is determined by MMCR.SBLEN bit setting. When MMCR.SBLEN bit = 0, the start bit length is 1 bit. When MMCR.SBLEN bit = 1, the start bit length is 3 bits.
When MMCR.SBLEN bit = 1, the Sync type can be selected from command Sync and data Sync.
Command Sync means the three start bits are added as a one-to-zero transition.
Data Sync means the three start bits are added as a zero-to-one transition.
The Sync type is determined by the MMCR.SYNCE, MMCR.SBPTN and TDR.SYNC bits settings. (When receiving, the received result is applied to MMSR.RSYNC bit.)
When MMCR.SBLEN bit = 0, the start bit is added as a zero-to-one or one-to-zero transition. The selection is determined by the MMCR.SBPTN setting.
The MMCR.SYNCE bit specifies the destination to be referred to when setting for transmission. When the MMCR.SYNCE bit is set to 1, the MMCR.SBPTN setting is referred to. When the MMCR.SYNCE bit is set to 0, the TDR.SYNC bit setting is referred to.
Figure 36.36 shows the state of the start bit area according to the settings in the MMCR.SYNCE, MMCR.SBPTN and TDR.SYNC bits in the case of transmission. Figure 36.37 shows that in the case of reception.
The start bit(s) is not affected by the MMCR.ENCS bit or MMCR.DECS bit setting.

Register Setting				Output Signal	Output Start Bit
MMCR			TDR	Output TXD Waveform	
SBLEN	SYNCE	SBPTN	SYNC		
0	d.c.	0	d.c.		1bit 0 to 1 transition
0	d.c.	1	d.c.		1bit 1 to 0 transition
1	0	0	d.c.		3bit 0 to 1 transition
1	0	1	d.c.		3bit 1 to 0 transition
1	1	d.c.	0		3bit 0 to 1 transition
1	1	d.c.	1		3bit 0 to 1 transition

d.c.: don't care
Preface: Preface area
START: Start bit area
DATA: User data area
P: Parity bit area
MP: Multi-processor bit area
STOP: Stop bit area

Figure 36.36 Settings Related to and Format of the Start Bit Area at Transmission

Register Setting				Input Signal	Start Bit Detection Result *1	Register Indication
MMCR			TDR	RXD Input Waveform		MMSR.RSYNC
SBLEN	SYNCE	SBPTN	SYNC			
0	d.c.	0	d.c.		Normal start bit (1 bit: 0 to 1 transition)	0
					Start bit error	0
					Start bit error	0
					Start bit error	0
0	d.c.	1	d.c.		Start bit error	0
					Normal start bit (1 bit: 1 to 0 transition)	0
					Start bit error	0
					Start bit error	0
1	d.c.	d.c.	d.c.		Start bit error	0
					Start bit error	0
					DATA Sync	0
					COMMAND Sync	1

d.c.: don't care
Preface: Preface area
START: Start bit area
DATA: User data area
P: Parity bit area
MP: Multi-processor bit area
STOP: Stop bit area

Note 1. Data other than the start bit is assumed to be normal.

Figure 36.37 Settings Related to and Judgment of the Start Bit Area at Reception

(3) Data

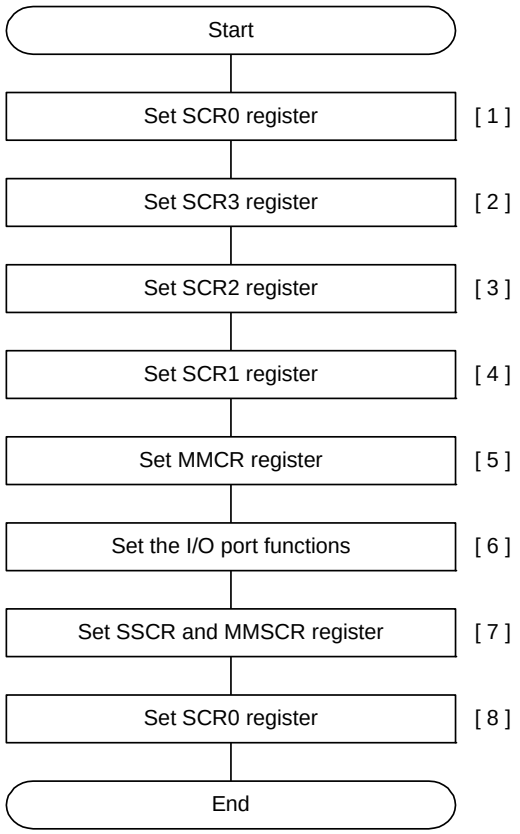
Since the format of the data area is the same as that of the asynchronous mode, see section 36.3.1, Serial Data Transfer Format.
As shown in Figure 36.34, Frame Format in Manchester Mode, the stop bit is not included in the manchester encoding range.

36.5.2 Clock

As the transfer clock in manchester mode, the clock generated by the on-chip baud rate generator is used by setting the SCR2.CKS[1:0] bits.
Also it is possible to set the oversampling (transfer rate of one bit period) by SCR2.ABCS bit. When the SCR2.ABCS bit is set to 0, oversampling ×16 is selected with the one-bit period being 16 cycles of the base clock. When the SCR2.ABCS bit is set to 1, oversampling ×8 is selected with the one-bit period being 8 cycles of the base clock.

36.5.3 RSCI Initialization of Manchester Mode

Before transferring data, write 0 to SCR0.TE and SCR0.RE bits (or write the initial value to SCR0 register), and initialize the RSCI following the example of flowchart shown in Figure 36.38.
Be sure to write 0 to SCR0.TE and SCR0.RE bits before changing the operation mode or communication format.
Note that setting the SCR0.RE bit to 0 initializes none of the SSR.ORER, AFER, APER, RDRF, MMSR.MCER, SYER, PFER and SBER flags, and the RDR registers.
Note also that switching the value of SCR0.TE bit from 0 to 1 when SCR0.TIE bit is 1 generates a TXI interrupt request.



[1] Set the following bits of SCR0 register to 0.
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set the driver control, multi-processor communication, the communication mode, transmission/reception format, and clock enable.
Leave unused bits such as CPOL and CPHA bits at their initial values.

[3] Set the clock selection, bit rate setting.
Leave unused function bits at their initial values.

[4] Set the noise filter, loopback control, communication terminal status, parity check, and CTS/RTS function.

[5] Set manchester transmission/reception setting.
Start bit setting, retiming setting, manchester polarity setting, transmission preface setting, reception preface setting.

[6] Set the I/O port so that the required pin function is valid among the TXDn, RXDn, and SCKn pins.

[7] Set 1 to the following clear bits in the SSCR and MMSCR registers to clear the target flag.
SSCR.RDRFC, AFERC, APERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
MMSCR.MCERC, SBERC, SYERC, PFERC
This procedure can be omitted for the initial setting after reset.

[8] Set the SCR0.TE or RE bit to 1. To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time. Set the multiprocessor function. The TXDn and RXDn pins can be used by setting the TE and RE bits.

Figure 36.38 RSCI Initialization Flow in Manchester Mode

36.5.4 Double Speed Operation

When the ABCS bit in SCR2 register is set to 1 and eight pulses of the base clock for a 1 bit period is selected, the RSCI operates on the bit rate twice that of when ABCS is set to 0.
When the BGDM bit in SCR2 register is set to 1, the cycle of the base clock is reduced to half and the RSCI operates on the bit rate twice that of when BGDM is set to 0.
When the ABCS bit in SCR2 register and the BGDM bit in SCR2 register are set to 1, the RSCI operates on the bit rate four times that of when the ABCS bit in SCR2 register and the BGDM bit in SCR2 register are set to 0.

36.5.5 CTS, RTS Functions

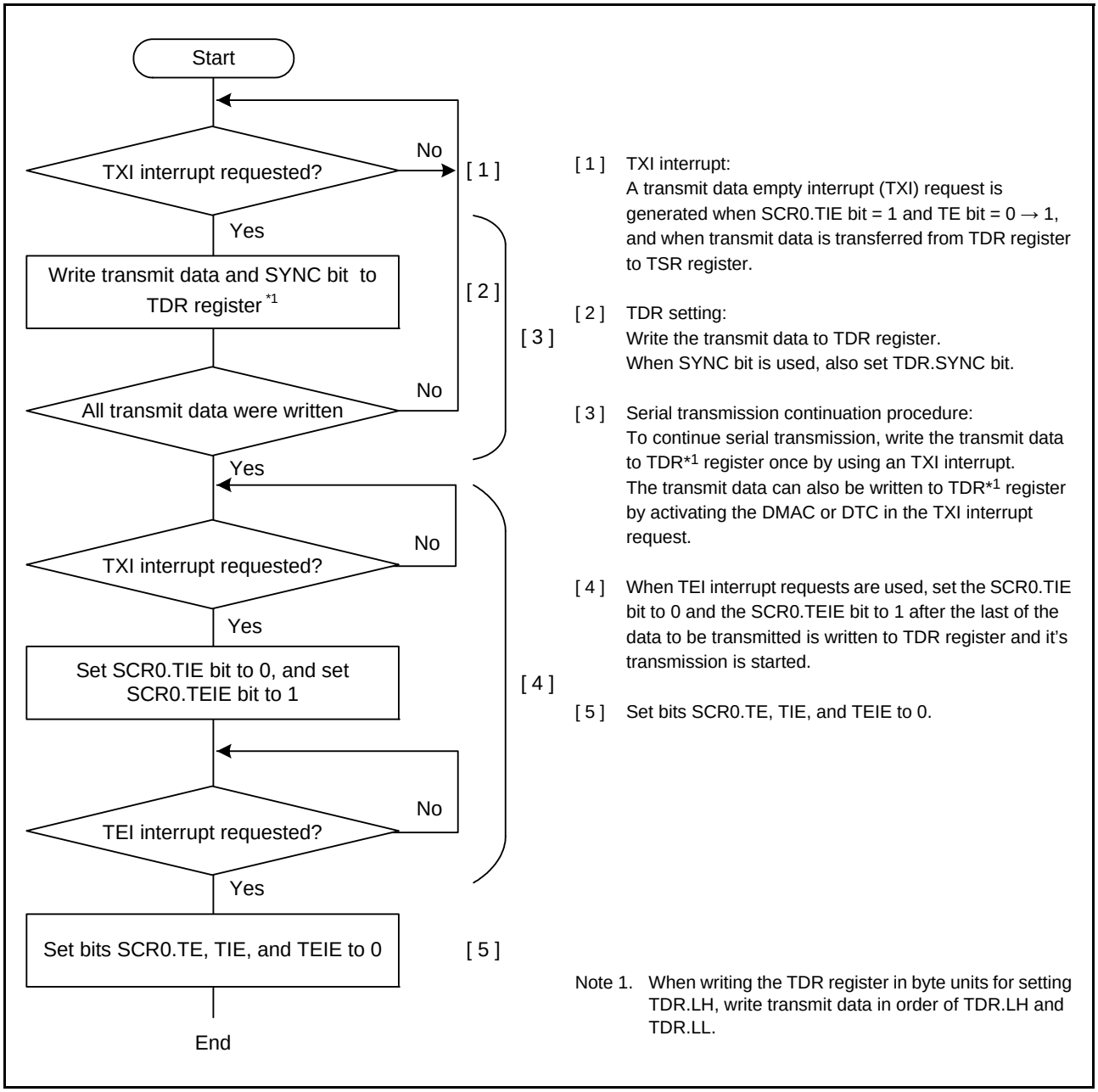
The CTS function controls transmission using the CTSn # pin input. Setting the CTSE bit in SCR1 register to 1 enables the CTS function. The CTSn#/RTSn# pin can be set as a multiplexed pin which allows one pin to be used for either function, or as dedicated pins with each pin for a single function. Use the CRSEP bit in SCR1 register for this setting.
When the CTS function is enabled, transmission starts only when the CTSn# pin is at the low level.
Even if the level of CTSn# pin goes High after transmission starts, does not affect transmission of the current frame, which continues.
The RTS function uses output on the RTSn# pin to request transmission. When the RSCI is ready to receive, it outputs a low level to the RTSn# pin, Conditions for output of the low level and high level are as follows:
[Conditions for low-level output]
When all conditions listed below are satisfied:

- The value of the SCR0.RE bit is 1.
- There are no received data yet to be read and reception is not in progress.
- All of the following flags are set to 0:
  SSR.ORER, AFER, APER and MMSR.MCER, SBER (when SBERIE bit = 1), SYER (when SYERIE bit = 1), PFER (when PFERIE bit = 1)

[Conditions for high-level output]
When the conditions for low output are not satisfied.

36.5.6 Serial Data Transmission in Manchester Mode

The RSCI encodes data in Manchester encoding and sends the resultant data in manchester mode.
When the polarity setting (MMCR.ENCS bit) set to 0, logic 0 is coded as a zero-to-one transition in Manchester code and logic 1 is coded as a one-to-zero transition in Manchester code.
When the polarity setting (MMCR.ENCS bit) set to 1, logic 0 is coded as a one-to-zero transition in Manchester code and logic 1 is coded as a zero-to-one transition in Manchester code.
For this reason, a level transition occurs with the Manchester encoded data in the middle of individual logic data. (See Figure 36.33)
The transmitter constructs transmit frames in a specific format by adding a preface area to data and setting the start bit(s) according to the polarity setting and sends resultant serial data.
For details on the frame format, see section 36.5.1, Frame Format.
Figure 36.39 shows the flowchart in transmission. At transmission starts, set the SCR0.TIE and SCR0.TE bits to 1 simultaneously with one instruction. Then, a TXI interrupt request is generated.
Figure 36.40 to Figure 36.42 show examples of the operation for serial transmission in manchester mode.



[1] TXI interrupt:
A transmit data empty interrupt (TXI) request is generated when SCR0.TIE bit = 1 and TE bit = 0 → 1, and when transmit data is transferred from TDR register to TSR register.

[2] TDR setting:
Write the transmit data to TDR register.
When SYNC bit is used, also set TDR.SYNC bit.

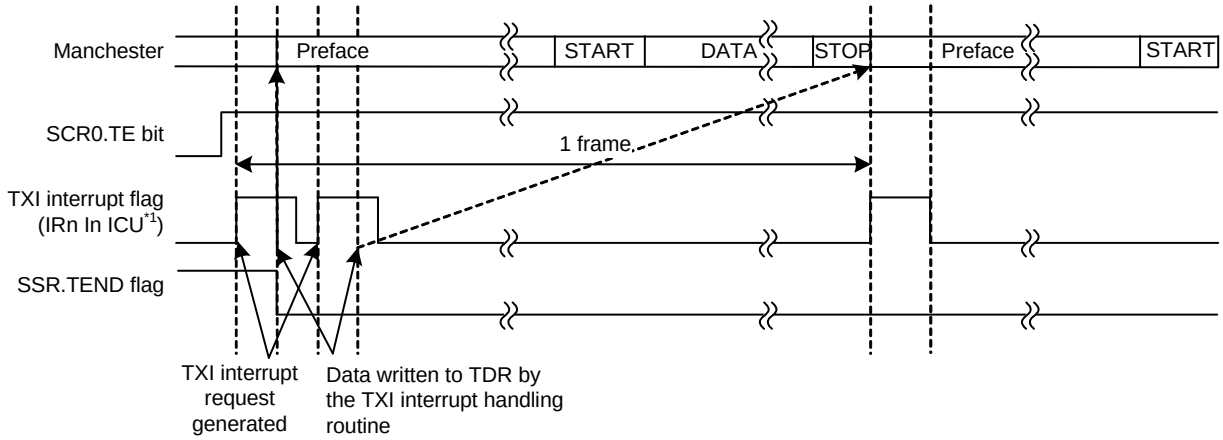
[3] Serial transmission continuation procedure:
To continue serial transmission, write the transmit data to TDR*1 register once by using an TXI interrupt.
The transmit data can also be written to TDR*1 register by activating the DMAC or DTC in the TXI interrupt request.

[4] When TEI interrupt requests are used, set the SCR0.TIE bit to 0 and the SCR0.TEIE bit to 1 after the last of the data to be transmitted is written to TDR register and it's transmission is started.

[5] Set bits SCR0.TE, TIE, and TEIE to 0.

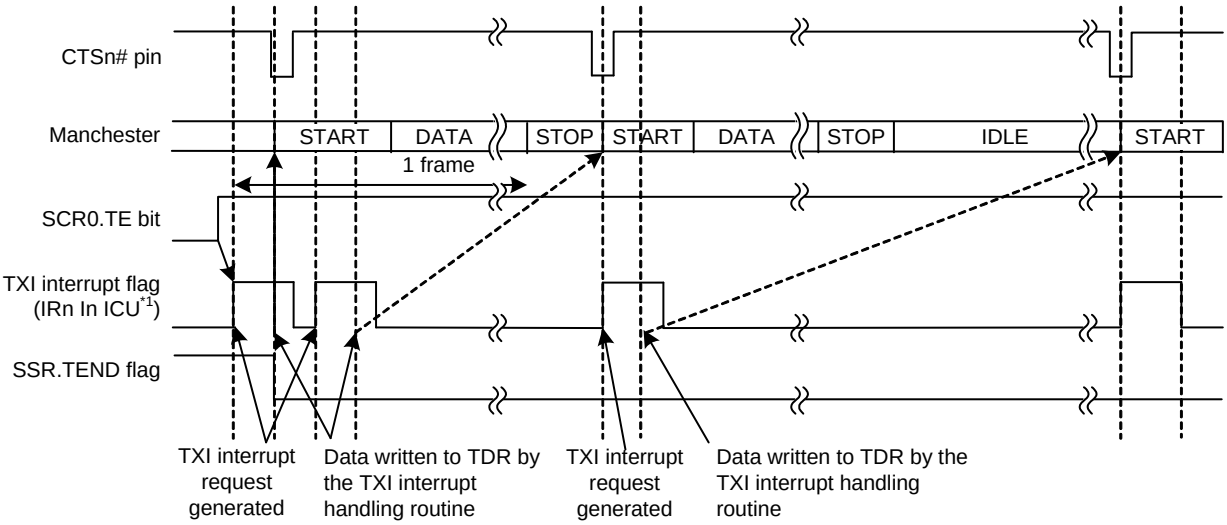
Note 1. When writing the TDR register in byte units for setting TDR.LH, write transmit data in order of TDR.LH and TDR.LL.

Figure 36.39 Serial Transmission Flowchart in Manchester Mode



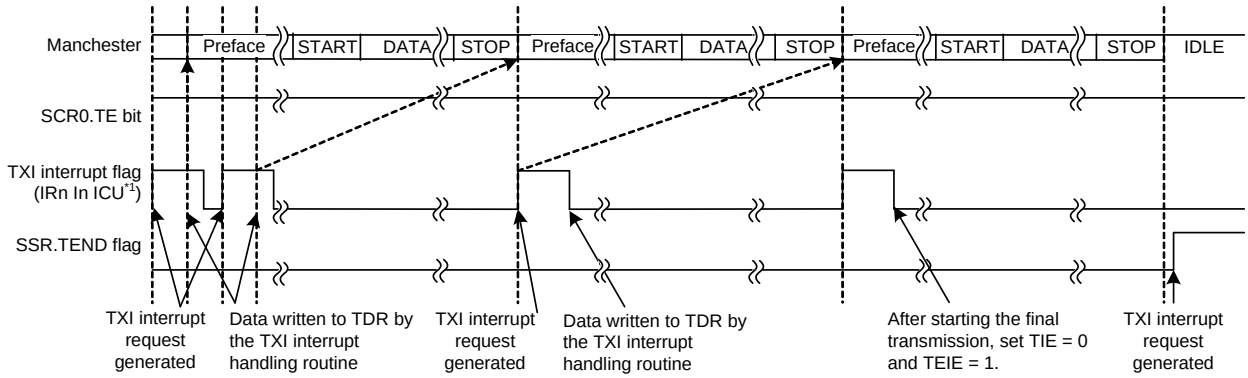
Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 36.40 Example of Start-of-Transmission Operation for Serial Transmission in Manchester Mode (with Preface but without the CTS Function)



Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 36.41 Example of Start-of-Transmission Operation for Serial Transmission in Manchester Mode (without Preface but with the CTS Function)



Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 36.42 Example of End-of-Transmission Operation for Serial Transmission in Manchester Mode (with Preface but without the CTS Function)

36.5.7 Serial Data Reception in Manchester Mode

In manchester mode, the RSCI operates on a base clock with a frequency of 16 times*1 the bit rate. Reception starts by sampling the falling edges of received data at the base clock. As shown in Figure 36.43, reception starts at a falling edge of the received data and it continues if the received data keeps low for the duration of 1/4 bit. If the received data goes high within the duration of 1/4 bit, the RSCI judges it as an error and waits for a falling edge again.
If a high level is expected in the first half of a bit in the received data, the RSCI judges a low level that continues for one base clock cycle as an error and ignores the change to the low level.

Note 1. This is the case when SCR2.ABCS bit = 0. When SCR2.ABCS bit = 1, the RSCI operates on a base clock with a frequency of 8 times the bit rate.

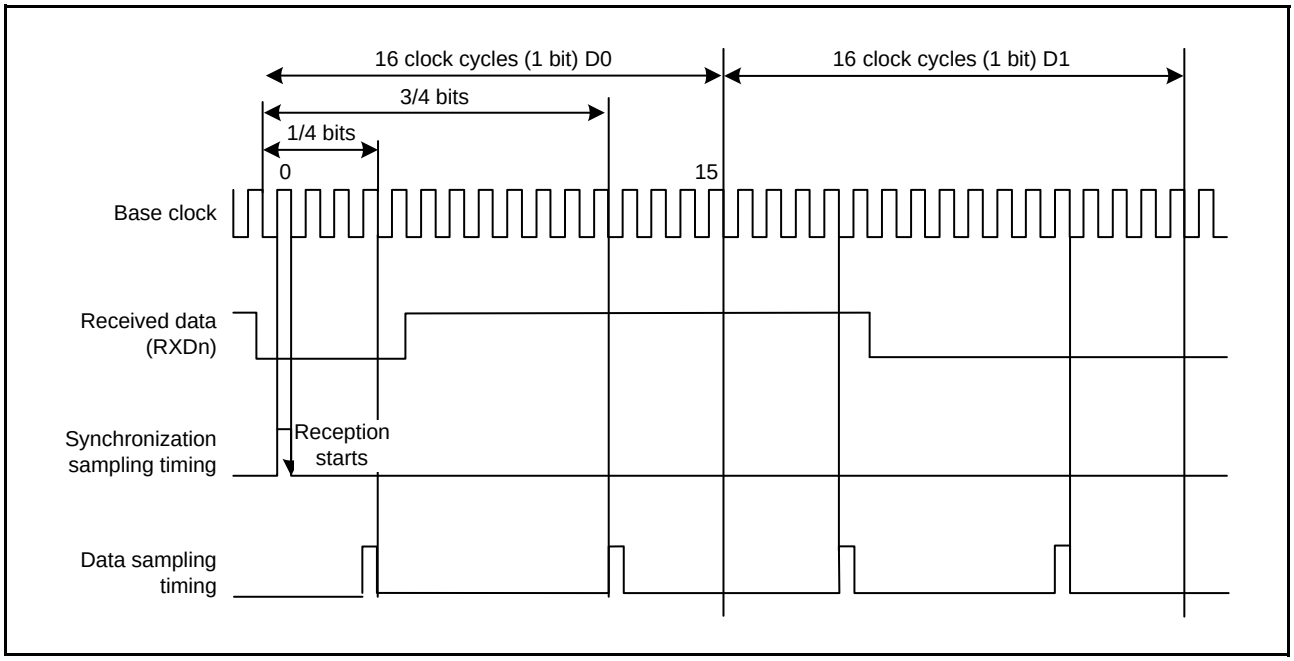
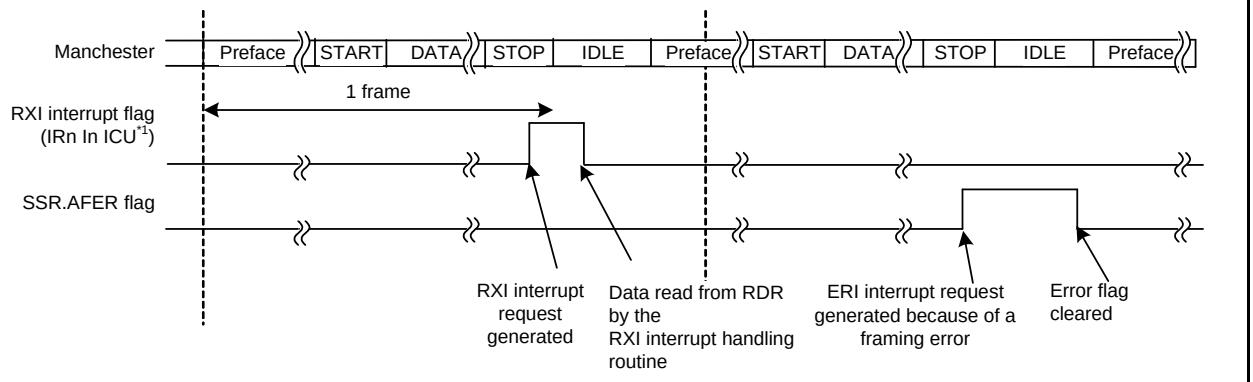


Figure 36.43 Data Reception Sampling Timing in Manchester Mode

In manchester mode, data reception starts with detection of a preface and start bit area.
The RSCI checks the input from the RXDn pin to see whether a preface is added based on the value of MMCR.RPLEN[3:0] bits.
If the preface is disabled (MMCR.RPLEN[3:0] bits = 0), it moves on to the detection of a start bit area without detecting a preface.
When a preface is enabled, it identifies a preface pattern setting according to the set value in MMCR.RPPAT[1:0] bits, and compares it with the RXD input for a pattern match to detect a preface pattern.
Upon detection of a preface pattern match, it judges it as a normal preface and moves on to the detection of a start bit area.
If detecting a preface pattern mismatch or a Manchester code error in the preface area, it judges it as a preface error and asserts a preface error (PFER).
For start bit detection, the RSCI selects an expected value based on the register settings (MMCR.SBLEN and SBPTN bits), compares it with the RXD input for a pattern match to detect a start bit area. Upon detection of a start bit pattern match, it judges it as a normal start bit area and moves on to the data processing.
Only when a preface and a start bit area are detected normally, it moves on to the next phase of data reception.
Upon detection of a start bit pattern mismatch, it asserts a start bit error flag (SBER).
In data processing, the RSCI shifts the data by the expected received data length based on the register settings (SCR3.CHR[1:0] bits) through the RSR register. If two sampling points in a bit of the received data are identical, the RSCI judges this as a Manchester code error. For details, see section 36.5.11, Errors in Manchester Mode (4).
When the parity function is disabled (SCR1.PE bit = 0), the RSCI moves on to the next phase of stop bit detection. When

the parity function is enabled (SCR1.PE bit = 1), the RSCI performs parity checking. If detecting a parity error, it asserts a parity error flag (APER), and then moves on to stop-bit detection.
In stop bit detection, the RSCI checks the following in the stop bit area of the received frame:
It has two sampling points in a bit. If both points are at the high level, the bit is recognized as a normal stop bit and the data is stored in the RDR register. At least one low-level point is judged as an abnormal stop bit, causing a framing error flag (AFER) to be set. Even when an error is detected, the received data is stored in the RDR register as abnormal data.
Figure 36.44 shows an example of the operation for serial data reception in manchester mode.

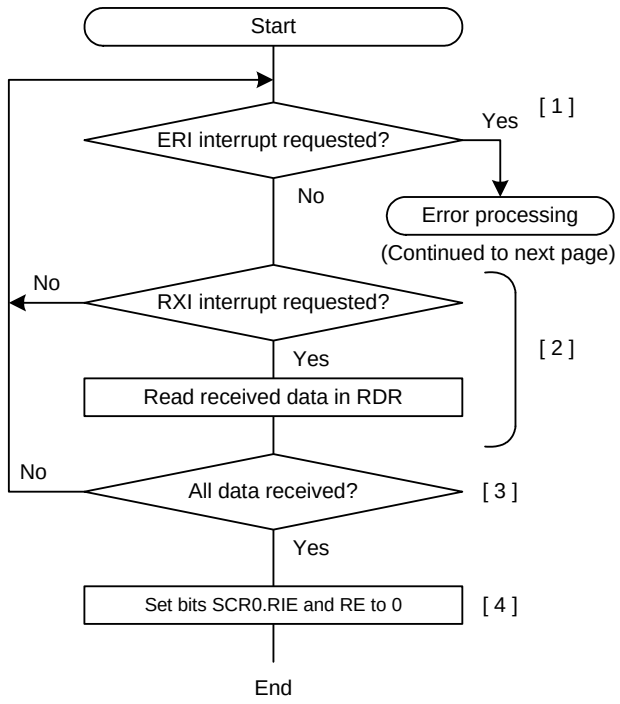


Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 36.44 Example of Operation for Serial Data Reception in Manchester Mode (with a Preface)

For the state of each status flag in the SSR and MMSR registers and RXD input processing when a receive error is detected, see section 36.5.11, Errors in Manchester Mode.
If a receive error is detected, an ERI interrupt request is generated but an RXI interrupt request is not generated.
Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, AFER, APER, MCER, SYER*2, PFER*2, SBER*2 flags to 0 before resuming reception. Also, be sure to read the RDR register during overrun error processing. When a reception is forcibly terminated by setting the SCR0.RE bit to 0 during operation, read the RDR register because received data which has not yet been read may be left in the RDR register.
Figure 36.45 and Figure 36.46 show examples of serial data reception flowchart in manchester mode.

Note 2. Effective when the corresponding bit is enabled.



[1] Receive error processing:
When a receive error occurs, RSCI generates an ERI interrupt. An error is identified by reading the ORER, APER, AFER in SSR register and MCER, SBER, SYER, PFER in MMSR register. After performing the appropriate error processing, always set the ORER, APER, AFER, MCER, SBER, SYER, PFER flags to 0. Reception cannot be resumed if any of these flags is set to 1.

Note: Each of the PFER, SYER, SBER bits in MMSR register has an enable bit (PFERIE, SYERIE, and SBERIE bits in MMCR). If the corresponding enable bit is set to 0 (disable), the flag value does not affect the communication. Perform error processing as required.

[2] Read the received data in RDR register once in the RXI interrupt handling routine.

[3] Serial reception continuation procedure:
To continue serial reception, before the stop bit of a frame is received, read data from RDR register in the RXI interrupt handling routine. The RDR data can also be read by activating the DMAC or DTC.

[4] Write 0 to SCR0.RIE and RE bits.

Figure 36.45 Example of Serial Data Reception in Manchester Mode (Normal)

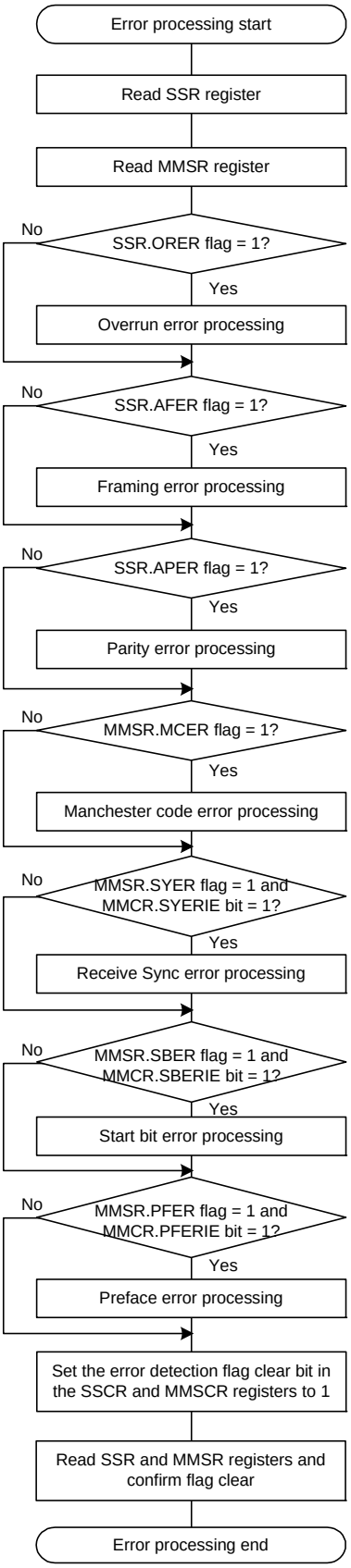


Figure 36.46 Example of Serial Reception Flowchart in Manchester Mode (Error Processing)

36.5.8 Operation When Multi-Processor Bit is Used

See section 36.4, Multi-Processor Communication Function (1) for the operation in manchester mode when using multi-processor mode because the operation is the same.
A preface and a start bit area are added to the frame format in manchester mode.
See Figure 36.46 for error processing in manchester mode for the reception flowchart (Figure 36.29). Refer to Table 36.36 for the operation status when detecting various errors.

36.5.9 Receive Retiming

This function corrects the timing for each central edge of the bit, taking advantage of the fact that each bit has an edge in the center in Manchester code.
The receive retiming function can be turned on or off by setting the SADJE bit in the MMCR register.
When the receive retiming function is turned off (MMCR.SADJE bit = 0), retiming is not performed, causing misalignment between the internal clock and the RXD input to be accumulated and the receive margin to be reduced.
When the receive retiming function is turned on (MMCR.SADJE bit = 1), retiming is performed for the preface area, the start bit area*1, and the data area (excluding the stop bit).

Note 1. Retiming is not performed for the start bit area if the preface length is 0 and the start bit length is 3.

As an example, the receive retiming when oversampling ×16 is selected is shown below.
When detecting an RXD input edge two to four cycles before the expected receive cycle, the receive processing is shortened by one sampling CLK cycle.
When detecting a RXD input edge two to three cycles after the expected receive cycle, the receive processing is extended by one sampling CLK cycle.
(Even if the clock is misaligned with the data by more than two cycles, one cycle is corrected for each bit.)
Figure 36.47 shows the conceptual image of receive retiming range.
When detecting an edge in the tolerance area in the figure, data is received as is without making correction.
When detecting an edge in the SyncJump area in the figure, data is corrected for reception.
When detecting an edge in the SyncError area in the figure, data is received as abnormal data with no correction made.
For a Manchester code error (data matches at the 1/4-phase and 3/4-phase sampling points), the RSCI reports a code error.

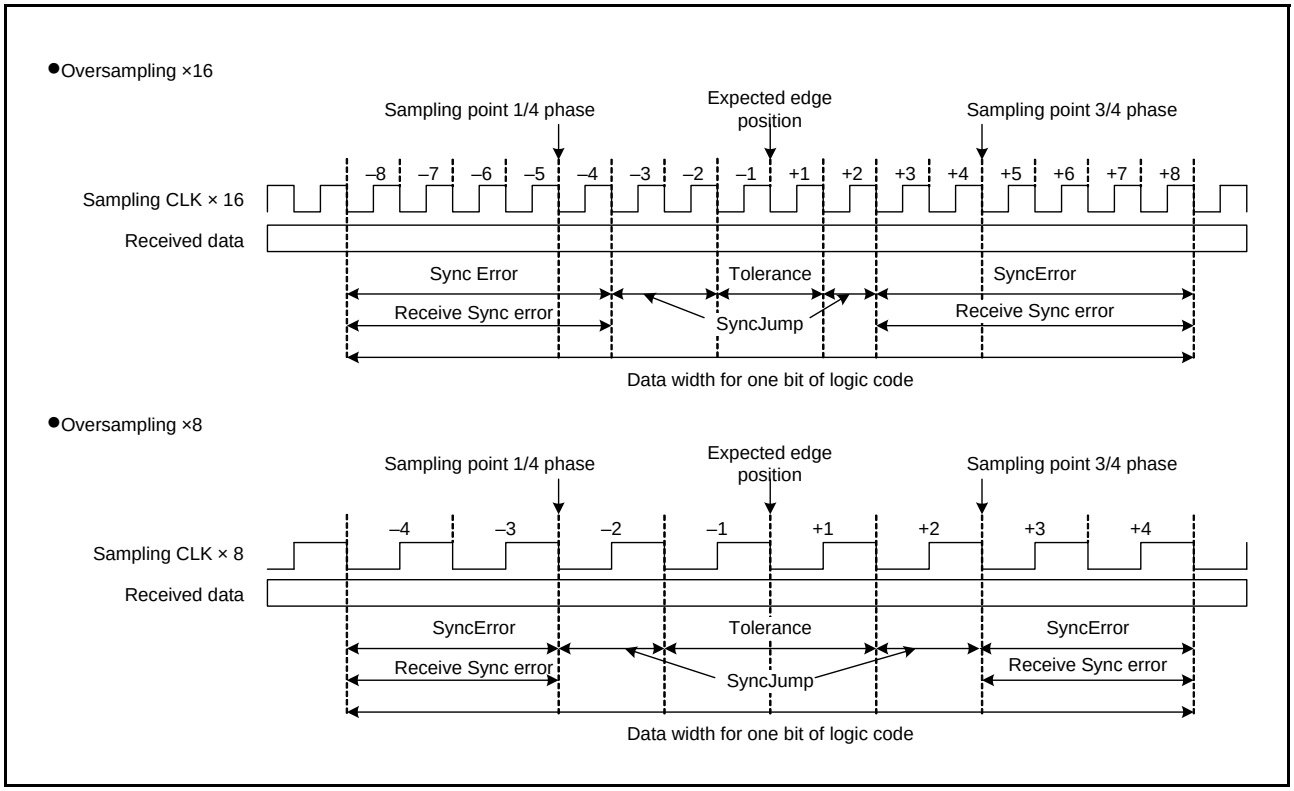


Figure 36.47 Conceptual Image of Reception Retiming Range

36.5.10 Polarity Setting for Manchester Code

The polarity of the Manchester code can be set with the Manchester Control Register (MMCR).

It can be set separately for transmission and reception. Use the MMCR.ENCS bit to set the polarity for transmission and the MMCR.DECS bit to set the polarity for reception.

The Manchester code polarity setting is valid for the preface area, the data area, and the parity or multi-processor area.

When the initial settings (ENCS/DECS bit = 0) are used for the polarity of Manchester code, logic 0 is encoded as a zero-to-one transition in Manchester code and logic 1 is encoded as a one-to-zero transition in Manchester code.

If the settings are changed to ENCS/DECS bit = 1, logic 0 is encoded as a one-to-zero transition in Manchester code and logic 1 is encoded as a zero-to-one transition in Manchester code. Figure 36.48 shows the conceptual image of the settings and operation.

Separately from the function above, the transmitted and received data in the data area can be inverted by the transmitted/received data inversion function (SCR3.DINV bit). Since the polarity of Manchester code (MMCR.ENCS/DECS bit) can be set separately from the transmitted/received data invert function (SCR3.DINV bit), if both are set to inversion (MMCR.ENCS/DECS bit = 1 and SCR3.DINV bit = 1), the transmitted and received data are set to initial state (inversion + inversion = normal).

The polarity of the start bit area can be set by a register different from the ones mentioned above.

Since a different register is used, the polarity of the start bit area is not affected by the polarity setting for Manchester code mentioned above.

For details on the setting for the start bit area, see section 36.5.1, Frame Format (2).

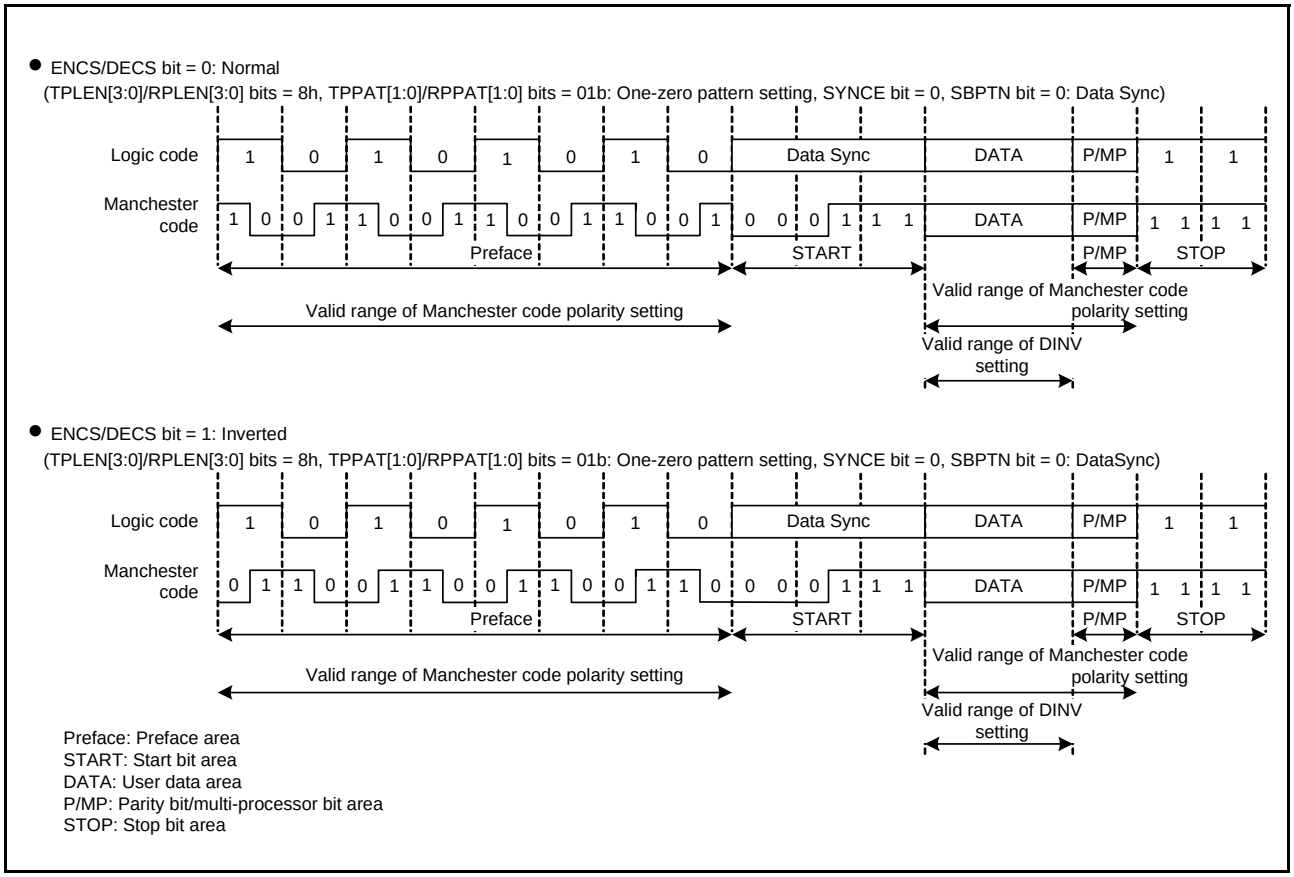


Figure 36.48 Valid Range of the Manchester Code Polarity Setting

36.5.11 Errors in Manchester Mode

There are the following errors in manchester mode:

(1) Parity error
(2) Over run error
(3) Framing error
(4) Manchester code error
(5) Preface error
(6) Start bit error
(7) Receive Sync error

For errors (1) to (3), see section 36.3.9, Serial Data Reception (Asynchronous Mode) (1) because they are the same as in asynchronous mode.
Each error is judged in each area, but they are reflected on flags and operations at the timing of 3/4-bit sampling of the STOP bit area. If a preface error or start bit error is detected, subsequent data will not be received. Therefore, no other error detection is performed, and the error flag holds the previous information.
Table 36.34 lists the states of the serial status register when detecting errors and judgment about whether to store data in the RDR register.
Table 36.35 lists the errors that can be detected in each area of a Manchester frame. If a Preface error or Start bit error is detected, subsequent data will not be received. Therefore, no other error detection is performed, and the error flag holds the result of the previous frame reception. Also, if an error is detected in the previous frame, data will not be received, but errors in the preface area and start bit area will update that flag. Table 36.36 shows the flags and actions in this case.

(4) Manchester Code Error

A Manchester code error is generated when a Manchester code error is detected.
In Manchester code, there must be an edge (transition) in the center of the bit.
In the data area of a received frame (including the parity/multi-processor bit), the values of the 1/4-bit and 3/4-bit sampling points are checked in each received 1-bit data, and a Manchester code error is determined if these two values matches.
If a Manchester code error is detected, the Manchester code error flag (MMSR.MCER flag) is asserted.
If a Manchester code error occurs, it is handled as an interrupt source and event source. If a Manchester error is detected, the next reception is not performed until the corresponding error flag is cleared.

(5) Preface Error

A preface error is generated when the preface pattern does not match or a Manchester code error is detected in the preface area. If a preface error is detected, the preface error flag (MMSR.PFER flag) is asserted.
It is possible to set whether to use this error flag as an interrupt source with the setting of the MMCR register.
When MMCR.PFERIE bit = 1, a preface error is handled as an interrupt source or event source. If a preface error is detected, the next reception is not performed until the corresponding error flag is cleared.
When MMCR.PFERIE bit = 0, a preface error is not handled as an interrupt source or event source, and the next reception is not halted. However, a preface error is notified to MMSR.PFER flag.

(6) Star Bit Error

A start bit error is generated when a mismatch is detected between the start bit area in the received frame and the preset start bit pattern. Upon detection of a start bit error, a start bit error flag (MMSR.SBER) is asserted.
It is possible to set whether to use the start bit error as an interrupt source with the setting of the MMCR register.
When MMCR.SBERIE bit = 1, a start bit error is handled as an interrupt source or event source. If a start bit error is detected, the next reception is not performed until the corresponding error flag is cleared.

When MMCR.SBERIE bit = 0, a start bit error is not handled as an interrupt source or event source, and the next reception is not halted. However, a start bit error is notified to MMSR.SBER flag.

(7) Receive Sync Error (SyncError)

When the receive retiming function described in section 36.5.9, Receive Retiming is enabled, the receive retiming operation is performed.
If no edges are detected within the receive retiming range (SyncError area in Figure 36.47) when receive timing operation is being performed, a receive Sync error is generated. Upon detection of a receive Sync error, a receive Sync error flag (MMSR.SYER) is asserted. In areas not subject to receive retiming, receive Sync errors are not detected.
The preface area *1, the start bit area*1, *2, and the data area (excluding the stop bit) for which receive retiming operation is performed are checked.
It is possible to set whether to use the receive Sync error as an interrupt source with the setting of the MMCR register.
When MMCR.SYERIE bit = 1, a receive Sync error is handled as an interrupt source or event source. If a receive Sync error is detected, the next reception is not performed until the corresponding error flag is cleared.
When MMCR.SYERIE bit = 0, a receive Sync error is not handled as an interrupt source or event source, and the next reception is not halted. However, a receive Sync error is notified to MMSR.SYER flag.

Note 1. In the case of a frame that starts with a pattern that expects the first half of the bit to be High, it is excluded from retiming.
Note 2. In the start bit area, when there is no preface length and 3 bit start bit is set, it is not subject to retiming. Also, the 1st bit and the 2nd bit in the start bit area when 3 bit start bit is set are not subject to retiming.

Table 36.34 Flags in the Status Register and Receive Data Handling in Manchester

Flags in the SSR Register			Flags in the MMSR Register					
ORER	AFER	APER	MCER	SBER *1	PFER *1	SYER	Received Data	Received Error Status (ERI Interrupt/Event Generation)
0	0	0	0	0	0	0	Transfer to RDR	No error
0	1	0	0	0	0	0	Transfer to RDR	Framing error
0	0	1	0	0	0	0	Transfer to RDR	Parity error
0	1	1	0	0	0	0	Transfer to RDR	Framing error + Parity error
0	0	0	1	0	0	0	Transfer to RDR	Manchester code error
0	1	0	1	0	0	0	Transfer to RDR	Framing error + Manchester code error
0	0	1	1	0	0	0	Transfer to RDR	Parity error + Manchester code error
0	1	1	1	0	0	0	Transfer to RDR	Framing error + Parity error + Manchester code error
1	0	0	0	0	0	0	Lost	Overrun error
1	1	0	0	0	0	0	Lost	Overrun + Framing error
1	0	1	0	0	0	0	Lost	Overrun + Parity error
1	1	1	0	0	0	0	Lost	Overrun + Framing error + Parity error
1	0	0	1	0	0	0	Lost	Overrun + Manchester code error
1	1	0	1	0	0	0	Lost	Overrun + Framing error + Manchester code error
1	0	1	1	0	0	0	Lost	Overrun + Parity error + Manchester code error
1	1	1	1	0	0	0	Lost	Overrun + Framing error + Parity error + Manchester code error
0	Combination of the above errors			0	0	1	Transfer to RDR	Error above + Receive Sync error*2
1				0	0	1	Lost	Error above + Receive Sync error*2
hold	hold	hold	hold	0	1	0	Lost	Preface error*3
hold	hold	hold	hold	1	0	0	Lost	Start bit error*3
hold	hold	hold	hold	0	1	1	Lost	Preface error*3 + Receive Sync error*2
hold	hold	hold	hold	1	0	1	Lost	Start bit error*3 + Receive Sync error*2

Note 1. Start bit error and preface error never become 1 at the same time.
Note 2. When MMCR.SYERIE bit = 1, ERI interrupt/event is generated by SYER factor.
Note 3. If MMCR.PFERIE bit = 1 or MMCR.SBERIE bit = 1, an ERI interrupt/event is generated when the corresponding flag is set.

Table 36.35 Errors Detectable in Each Area

	Preface Error (PFER)	Start Bit Error (SBER)	Manchester Code Error (MCER)	Receive Sync Error (SYER)	Parity Error (APER)	Framing Error (AFER)
Preface area	✓	—	—*1	✓*2	—	—
Start bit area	—	✓	—	✓*2	—	—
Data area	—	—	✓	✓	—	—
Parity area	—	—	✓	✓	✓	—
Multi-processor area	—	—	✓	✓	—	—
Stop bit area	—	—	—	—	—	✓

✓: Detected, —: Not detected
Note 1. When an Manchester code error occurs in the preface area, it is defined as a preface error.
Note 2. It may not be subject to receive Sync error detection. For details see section 36.5.11, Errors in Manchester Mode (7)

Table 36.36 Operation Status due to Presence/Absence of Error in Previous Frame and Operation Status List in Multiprocessor Mode (SCR0.MPIE Bit = 0)

Previous Frame	Each Area of the Frame					PFE RIE	SBE RIE	SYE RIE	Received Data	Error Flag	Interrupt Request	Event Signal
	Preface	Start Bit	Data	Parity Bit	Stop Bit							
No error	PFER, No SYER*1	No error	—	—	—	0	—	—	Lost	Set PFER*1	Not output	Not output
						1					Output	Output
	No error	SBER, No SYER*1	—	—	—	—	0	—	Lost	Set SBER*1	Not output	Not output
							1				Output	Output
	SYER, No PFER	No error	—	—	—	—	—	0	Transfer to RDR	Set SYER	Not output	Not output
								1	Lost		Output	Output
	No error	SYER, No SBER	—	—	—	—	—	0	Transfer to RDR	Set SYER	Not output	Not output
								1	Lost		Output	Output
	No error	No error	SYER		No error	—	—	0	Transfer to RDR	Set SYER	Not output	Not output
								1			Output	Output
	No error	No error	MCER		No error	—	—	—	Transfer to RDR	Set MCER	Output	Output
	No error	No error	—	APER	No error	—	—	—	Transfer to RDR	Set APER	Output	Output
	No error	No error	—	—	AFER	—	—	—	Transfer to RDR	Set AFER	Output	Output
	There is some error ORER					—	—	—	Lost	Set some flags*2	Output	Output
	No error	No error	No error	No error	No error, ORER	—	—	—	Lost	Set ORER	Output	Output
Some error*3, *6	PFER, No SYER*1	No error	—	—	—	0	—	—	Lost	Set PFER*1	Output*4	Not output*5
						1						
	No error	SBER, No SYER*1	—	—	—	—	0	—		Set SBER*1		
							1					
	SYER, No PFER	No error	—	—	—	—	—	0		Set SYER		
								1				
	No error	SYER, No SBER	—	—	—	—	—	0		Set SYER		
								1				
	No error	No error	SYER		No error	—	—	0		don't set any flags		
								1				
	No error	No error	MCER		No error	—	—	—				
	No error	No error	—	APER	No error	—	—	—				
	No error	No error	—	—	AFER	—	—	—				
	There is some error ORER					—	—	—				
	No error	No error	No error	No error	No error, ORER	—	—	—				

Note 1. If SYER is detected, the SYER flag is also set. Other operations are as shown in this table.
Note 2. Other detected error flags including ORER are also set.

Note 3. If all the error flags are cleared before the STOP bit is judged, the operation will be the same as the case where there is no error in the previous frame of this table.
Note 4. Since the ERI interrupt request is level output, it remains active due to errors in the previous frame regardless of the presence or absence of errors in the relevant frame.
Note 5. Since the error cause is continuously detected, the ERI event is not newly output regardless of the presence of errors in the relevant frame.
Note 6. For MMSR.PFER, SBER, and SYER, when each enable bit is set to disable, it is treated as no error.

Table 36.37 Operation Status List in Multiprocessor Mode (SCR0.MPIE Bit = 1)

MPB*1	Each Area of the Frame					PFERIE	SBERIE	SYERIE	Received Data	Error Flag	Interrupt Request	Event Signal
	Preface	Start Bit	Data	Parity Bit	Stop Bit							
1	No error	No error	—	—	—	—	—	—	Transfer to RDR	Set some flags	Output	Output
	No PFER, SYER	No SBER, SYER	—	—	—	—	—	0				
								1	Lost	don't set any flags	Not output	Not output
	PFER	No error	—	—	—	—	—	—				
	No error	SBER	—	—	—	—	—					

Note 1. If the received MPB flag is 0, it is not received the frame, and the operation is the same as lost of the reception data of this table.
Note 2. If no error is detected, RXI interrupt request or event is output, and if it is detected, ERI interrupt request or event is output.
Note 3. When SYER is detected in the preface area or the start bit area, the behavior of handling as an error depending on the MMCR.SYERIE bit changes.

36.6 HBS Support Mode

Setting the HBSCR.HBSE bit to 1 supports the negative logic RZI coding to generate waveforms (AMI, 50% duty cycle, negative logic) required by the home bus system. Since this function operates only in the asynchronous mode, refer to the asynchronous mode for the setting, transmission flow, and reception flow.

36.6.1 Reception in HBS Support Mode

When receiving in HBS support mode, the falling edge of the input from the RXDn pin is detected and the signal after the start bit is recognized is received. One frame is sampled according to the set bit rate, and if the stop bit is correctly received without error, the data value is stored in the receive data register RDR.

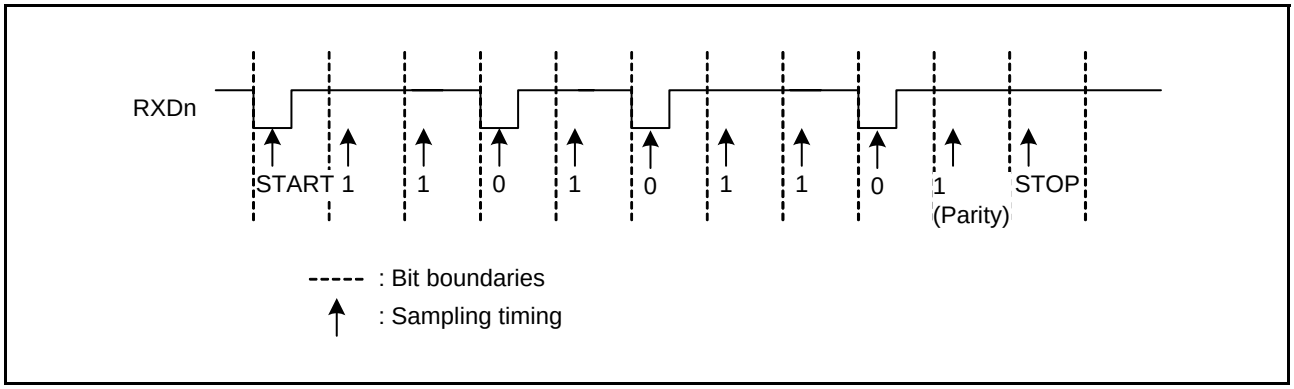


Figure 36.49 HBS Support Mode Reception Timing Chart

To receive in the HBS support mode, it is necessary to sample at the timing of 1/4 of 1 bit in order to capture the pulse in the first half of the 1 bit period. Sampling operates with a frequency 16 times the bit rate*1 as the base clock, similar to the asynchronous mode. The start bit is detected by detecting the Low level from the falling edge of RXD four times continuously with the base clock. When High is detected on the way, it is regarded as noise and waits for the next fall edge.

To set the sampling timing to 1/4 of the 1-bit period, enable the reception sampling timing adjustment function with the SCR4.RTADJ bit, set the SCR4.RTMG[3:0] bits to 1100b, and adjust Adjust from the center of the bit, which is the previous sampling timing, four clocks ahead of the base clock.

Since the sampling timing can be adjusted backward and forward using the reception sampling timing adjustment function, this timing can be adjusted according to the reception status. Increasing the SCR4.RTMG[3:0] bits value from 1100b will move the sampling timing forward, and decreasing it will move it backward. Refer to section 36.3.2, Receive Data Sampling Timing and Reception Margin in Asynchronous Mode for the details of adjustment.

After recognizing the start bit, sampling is performed at the timing according to the set bit rate, but the low width and high width of the waveform are not checked. Therefore, it is possible to receive even a normal asynchronous waveform.

Note 1. HBS support mode supports only following setting: SCR2.ABCS bit = 0 and SCR2.ABCSE bit = 0.

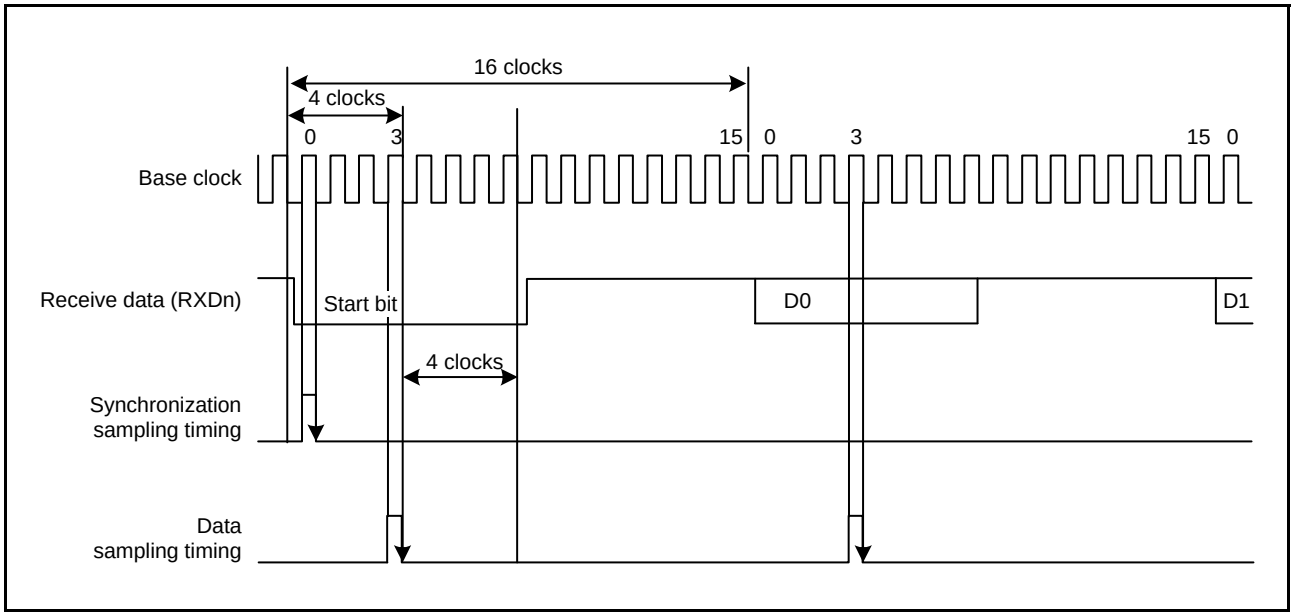


Figure 36.50 Details of Reception Sampling Timing in HBS Support Mode

36.6.2 Transmission in HBS Support Mode

Transmission in HBS support mode, data 0 is output as a low pulse for the first half of the 1-bit period.

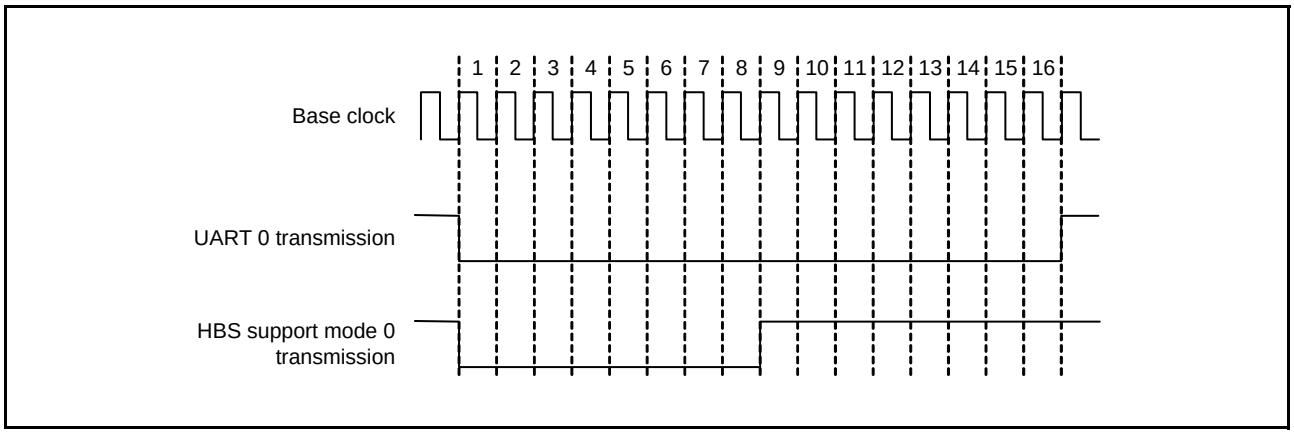


Figure 36.51 Transmission Waveform of HBS Support Mode

When HBSCR.AOE bit = 0, the all bits are output from TXDn pins, and when HBSCR.AOE bit = 1, data 0 is output alternately from TXDAn and TXDBn pins. Use the HBSCR.LPS bit to select which transmission pin starts output of the start bit.

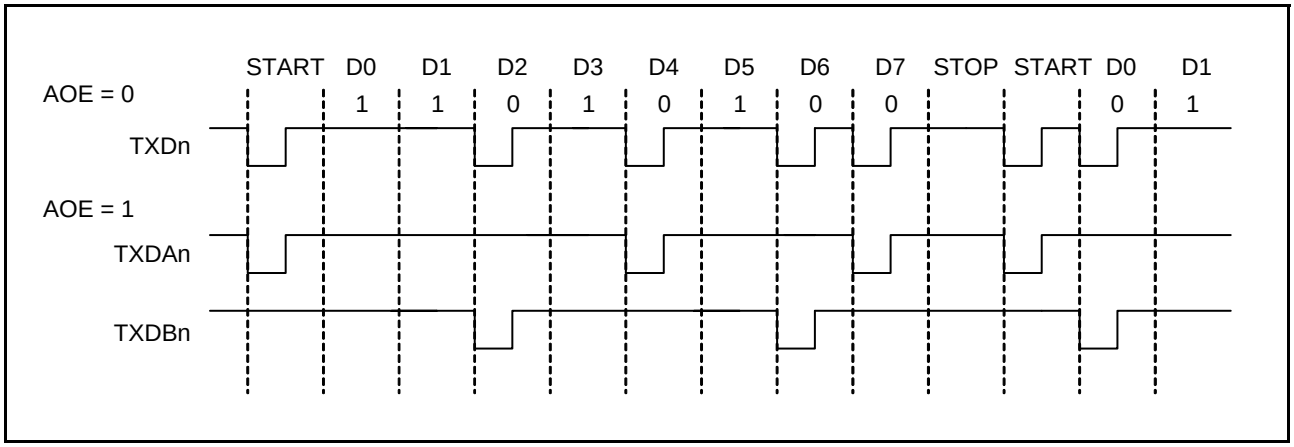


Figure 36.52 Difference in Transmission Waveform Depending on AOE Bit (When LPS Bit = 0)

Figure 36.52 shows an example of the transmission waveform for each HBSCR.AOE bit value. When the AOE bit is set to 0, the waveform is output from the TXDn pin, but when the AOE bit is set to 1, data 0 including start bit is output alternately from the TXDAn pin and the TXDBn pin.

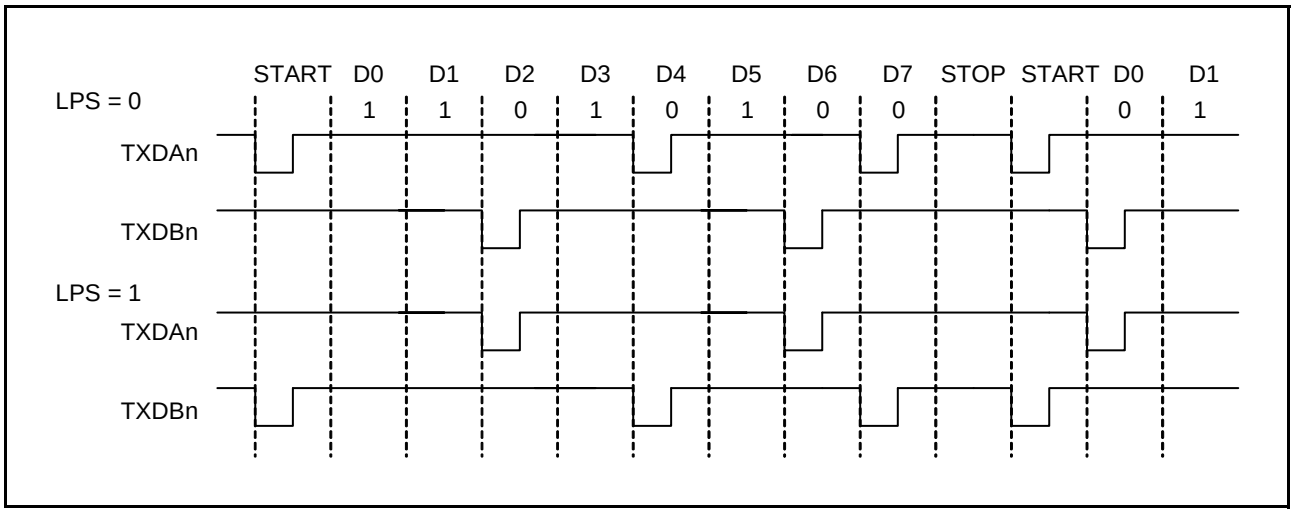


Figure 36.53 Difference in Transmission Waveform Depending on LPS Bit (When AOE Bit = 1)

Figure 36.53 shows an example of the transmission waveform for each HBSCR.LPS bit.
When the HBSCR.LPS bit = 0, the start bit is output from the TXDAn pin, and when the HBSCR.LPS bit is 1, the start bit is output from the TXDBn pin, and data 0 is output to each pin alternately. The start bit of the next frame starts output again from the pin specified by the HBSCR.LPS bit.
If the HBSCR.HBSE bit = 0, the TXDBn pin becomes High regardless of the settings of other bits.
When SCR0.TE bit = 0, both TXDn/TXDAn/TXDBn pins become high impedance, but can be controlled by SCR1.SPB2IO bit and SCR1.SPB2DT bit. At this time, the same output is applied to the TXDn/TXDAn/TXDBn pins.

36.6.3 Register Setting for HBS Support Mode

The HBS support mode is a part of asynchronous mode function, but there are some settings that are not supported when using this function. Set each bit of the control register as shown in Table 36.38 before use. Register bits not described can be set in the same way as the asynchronous mode.

Table 36.38 Control Register Setting Value for HBS Support Mode

Register Bit Name	Value	Remarks
SCR0.DCME	0	Use it when the data match detection function is disabled.
SCR1.NFCS[2:0]	000b	Use this setting when using the noise filter.
SCR1.HDSEL	0	Half-duplex communication with the TXDn pin cannot be used.
SCR1.CTSE	0	Please use it without the CTS function.
SCR2.BRME	0	Bit rate modulation function cannot be used.
SCR2.ABCSE	0	The setting that 6 cycles of the base clock becomes 1 bit cannot be used.
SCR2.ABCS	0	Only the setting that 16 cycles of the base clock becomes 1 bit can be used.
SCR3.CKE[1:0]	00b	Use with internal clock and without clock output.
SCR3.DEEN	0	Use without the RS-485 driver function.
SCR3.FM	0	Use without FIFO function.
SCR3.MOD[2:0]	000b	Set to asynchronous mode.
SCR3.RXDESEL	1	Detect the start bit at the falling edge of the RXDn pin input.
SCR3.STOP	0	Use with stop bit 1.
SCR3.DINV	0	Use without data inversion.
SCR3.DDIR	1	Use with LSB first.
SCR3.CHR[1:0]	10b	Use with 8 bit length.
SCR4.RTMG[3:0]	1100b	Use this setting when receiving in HBS support mode.*1
SCR4.TTADJ	0	Use without adjust transmit timing function.
SCR4.RTADJ	1	Use this setting when receiving in HBS support mode.

Note 1. This is the timing to sample at the center of the effective pulse. It can be adjusted if needed.

36.7 Smart Card Interface Mode

The RSCI supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards), as an extended function.
Smart card interface mode can be selected using the appropriate register.

36.7.1 Sample Connection

Figure 36.54 shows a sample connection between a smart card (IC card) and this MCU.
As in the figure, since this MCU communicates with an IC card using a single transmission line, interconnect the TXDn and RXDn pins and pull up the data transmission line to VCC using a resistor.
Setting the TE and RE bits in the SCR0 register to 1 with an IC card disconnected enables closed transmission/reception allowing self-diagnosis.
To supply an IC card with the clock pulses generated by the RSCI, input the SCKn pin output to the CLK pin of an IC card.
The output port can be used to output the reset signal.

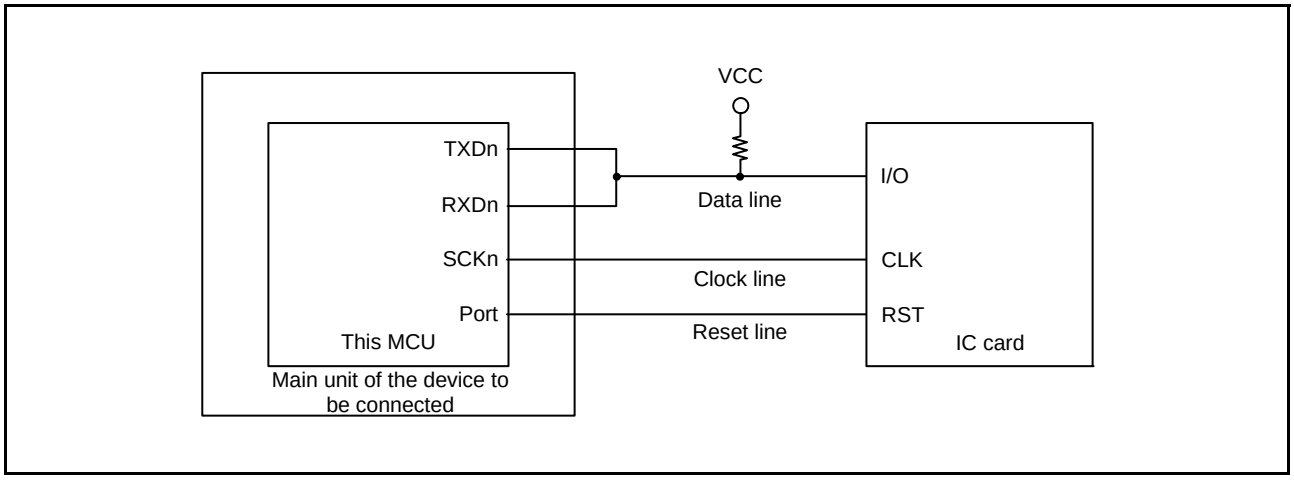


Figure 36.54 Sample Connection with a Smart Card (IC Card)

36.7.2 Data Format (Except in Block Transfer Mode)

Figure 36.55 shows the data transfer formats in smart card interface mode.

- One frame consists of 8-bit data and a parity bit in asynchronous mode.
- During transmission, at least 2 etu (elementary time unit: time required for transferring 1 bit) is secured as a guard time from the end of the parity bit until the start of the next frame.
- If a parity error is detected during reception, a low-level error signal is output for 1 etu after 10.5 etu has passed from the start bit.
- If an error signal is sampled during transmission, the same data is automatically retransmitted after at least 2 etu.

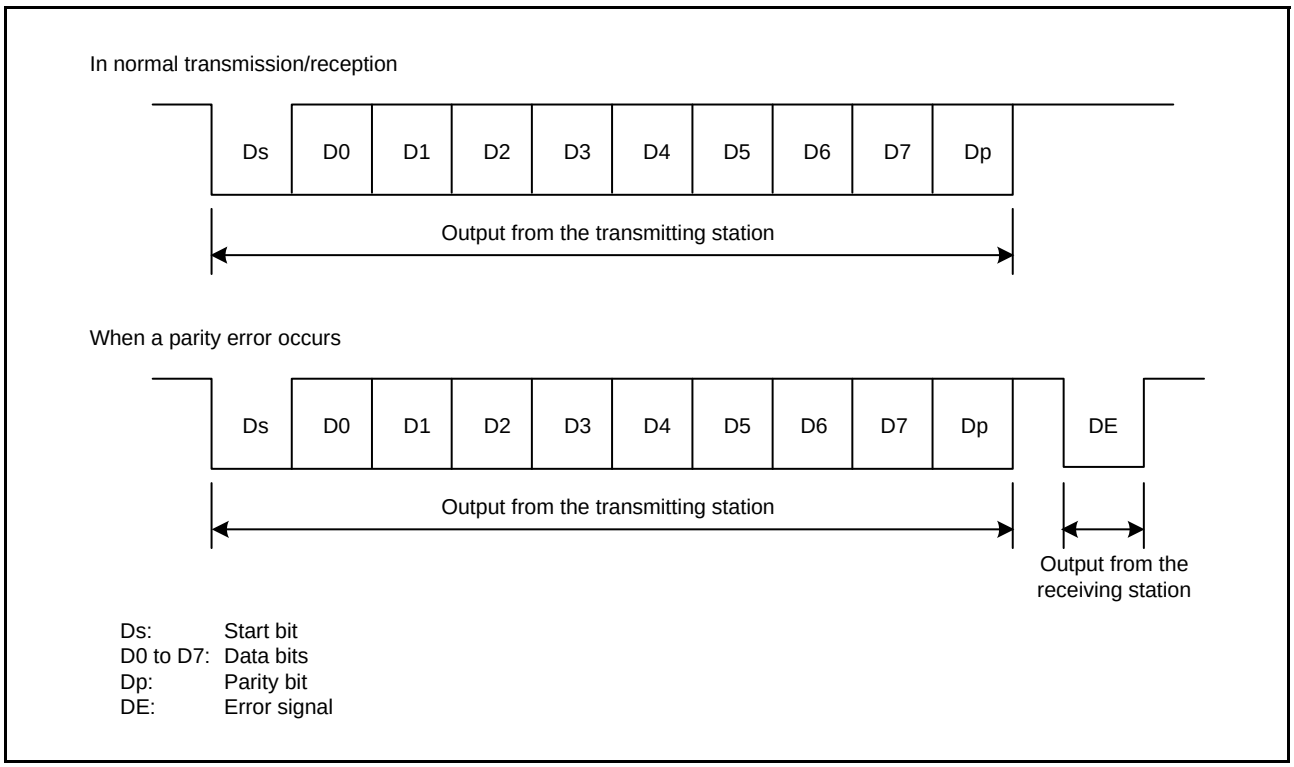


Figure 36.55 Data Formats in Smart Card Interface Mode

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 36.56. Therefore, data in the start character in the figure is 3Bh. When using the direct convention type, write 1 to the SCR3.DDIR bit and 0 to the SCR3.DINV bit. Write 0 to the PM bit in the SCR1 register in order to use even parity, which is prescribed by the smart card standard.

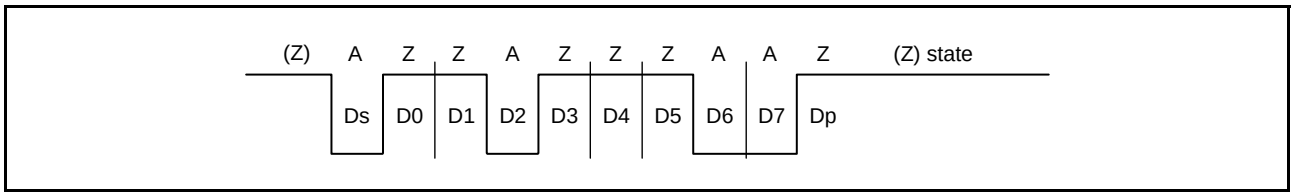


Figure 36.56 Direct Convention (SCR3.DDIR Bit = 1, SCR3.DINV Bit = 0, SCR1.PM Bit = 0)

(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 36.57. Therefore, data in the start character in the figure is 3Fh. When using the inverse convention type, write 0 to the SCR3.DDIR bit and 1 to the SCR3.DINV bit. 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 DINV bit of the this MCU only inverts data bits D7 to D0, write 1 to the PM bit in the SCR1 register to invert the parity bit for both transmission and reception.

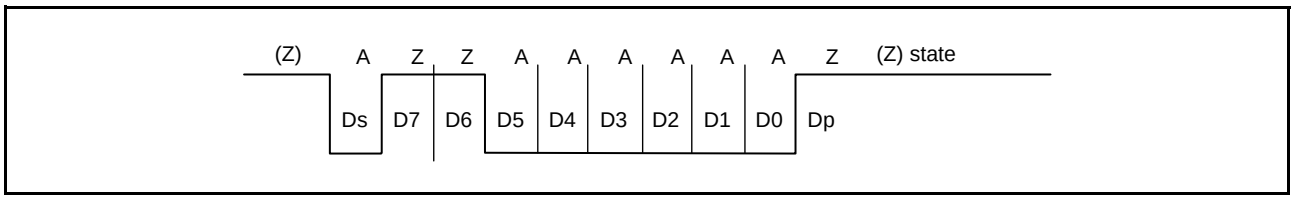


Figure 36.57 Inverse Convention (SCR3.DDIR Bit = 0, SCR3.DINV Bit = 1, SCR1.PM Bit = 1)

36.7.3 Block Transfer Mode

Block transfer mode is different from non-block transfer mode in the following respects.

- Even if a parity error is detected during reception, no error signal is output. Since the APER flag in the SSR register is set by error detection, clear the APER flag before receiving the parity bit of the next frame.
- During transmission, at least 1 etu is secured as a guard time from the end of the parity bit until the start of the next frame.
- Since the same data is not retransmitted during transmission, the TEND flag in the SSR register is set 11.5 etu after transmission start.
- In block transfer mode, the ERS flag in the SSR register indicates the error signal status as in non-block transfer mode, but the flag is read as 0 because no error signal is transferred.

36.7.4 Receive Data Sampling Timing and Reception Margin

Only the base clock generated by the on-chip baud rate generator can be used as a transmit/receive clock in smart card interface mode.

In this mode, the RSCI 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 SCR2.BCP[2:0] bits.

For data reception, the falling edge of the start bit is sampled with the base clock to perform 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 36.58. The reception margin here is determined by the following formula.

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ (\%)$$

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 × 372)} × 100 (%) = 49.866 (%)

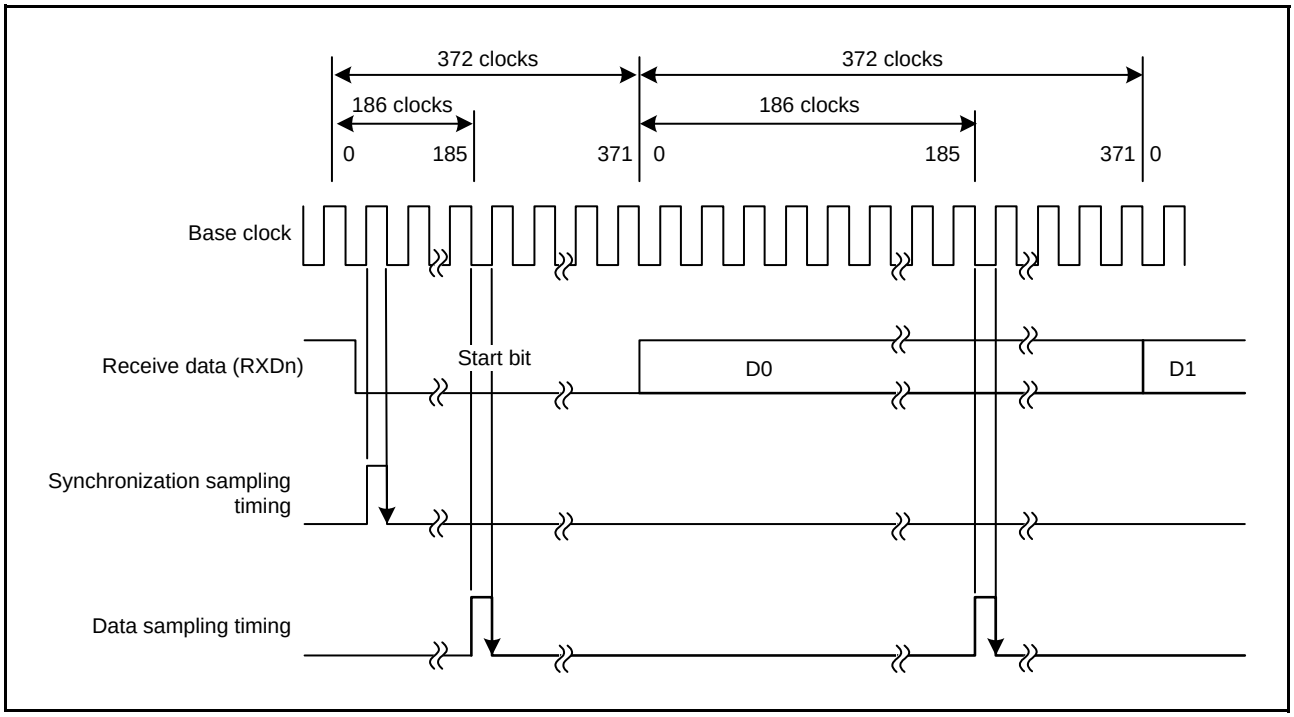


Figure 36.58 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)

36.7.5 RSCI Initialization (Smart Card Interface Mode)

Before transmitting and receiving data, write 0 to SCR0.TE bit and SCR0.RE bit (or write the initial value to SCR0 register). And initialize RSCI following example flowchart in Figure 36.59.
Be sure to set the initial value in the TIE, RIE, TE, RE, and TEIE bits in SCR0 register before switching from transmission mode to reception mode and vice versa. Even if the RE bit is set to 0, the RDR register is not initialized. In transmission mode, set 1 to the TE bit and TIE bit simultaneously, then the TXI interrupt request is generated. To change reception mode to transmission mode, first check that reception has completed, and then initialize RSCI. At the end of initialization, set TE bit = 1 and RE bit = 0. Reception completion can be verified by RXI interrupt request, SSR.ORER, or SSR.APER flag. To change transmission mode to reception mode, first check that transmission has completed, and then initialize RSCI. At the end of initialization, set TE bit = 0 and RE bit = 1. Transmission completion can be verified by reading SSR.TEND flag.

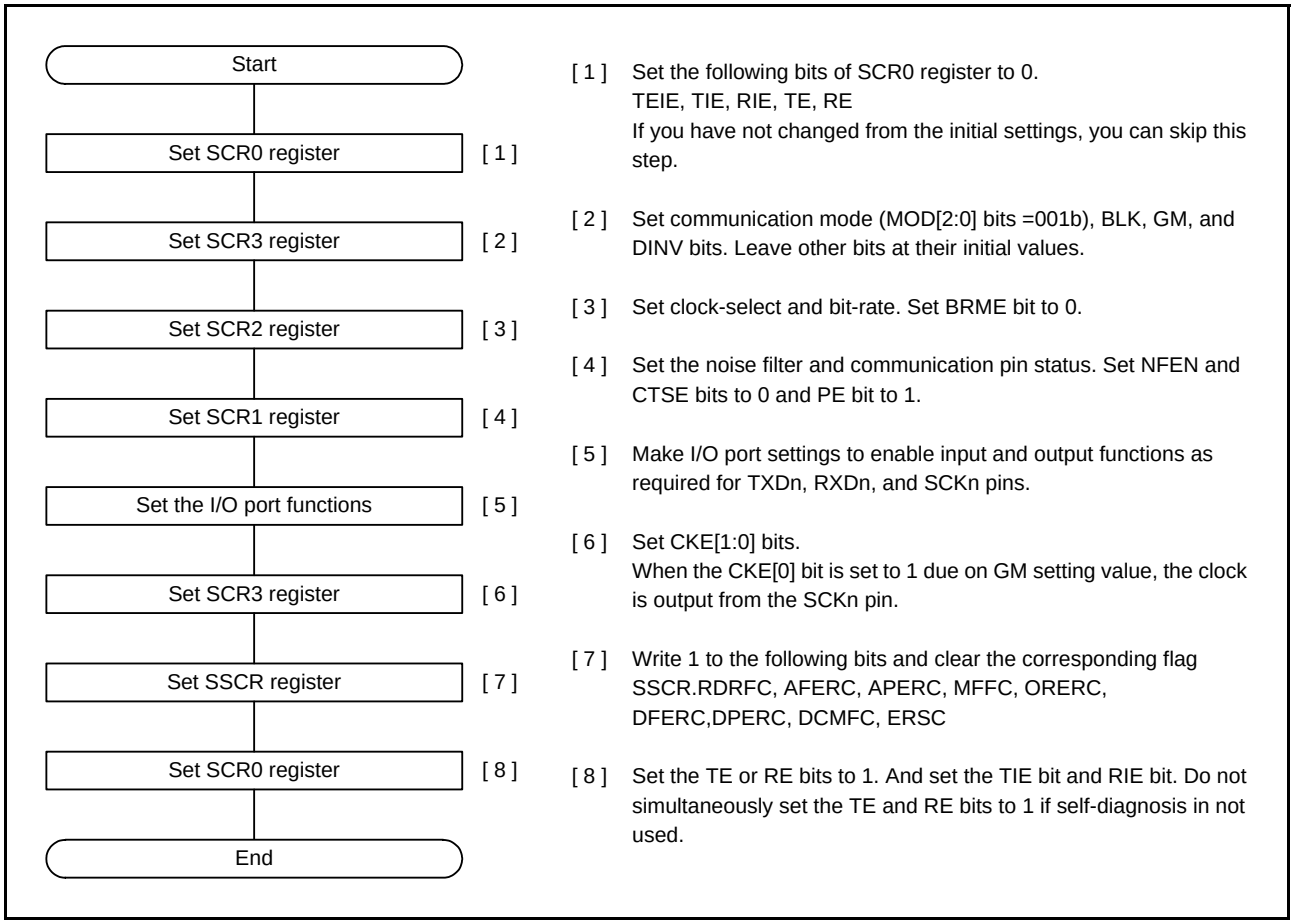


Figure 36.59 Example of RSCI Initialization Flowchart (Smart Card Interface Mode)

Figure 36.60 is a timing chart when data transmission is performed by making transition to the Smart Card Interface mode according to the above flow chart. The figure shows the case when SCR3.GM bit is 0. As shown in the figure, when the pin function is set to the SCKn pin, the SCKn pin is high impedance because the SCR3.CKE[0] bit is 0. When the TXDn pin is set, the TXDn pin is high impedance because the SCR0.TE bit is 0. Start clock output to the SCKn pin with the clock output setting SCR3.CKE[0] bit to 1, start data transmission by writing transmit data after setting SCR0.TE bit to 1.
In the smart card interface mode, even if not communicating at SCR0.TE bit = 0 and SCR0.RE bit = 0, the clock is continuously output if the clock output setting is used.

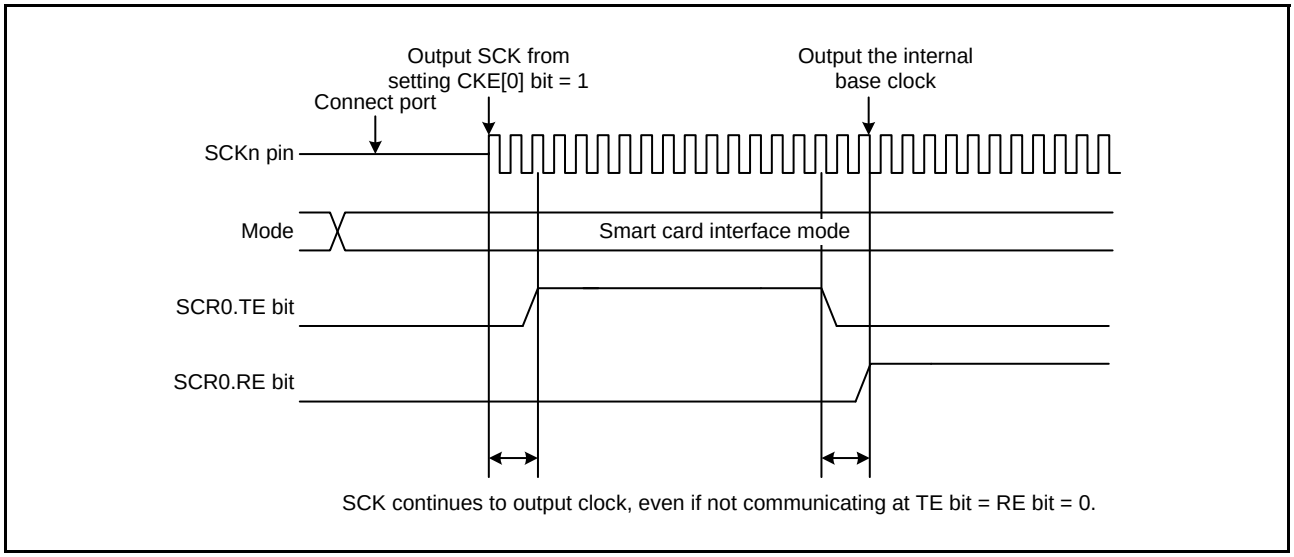
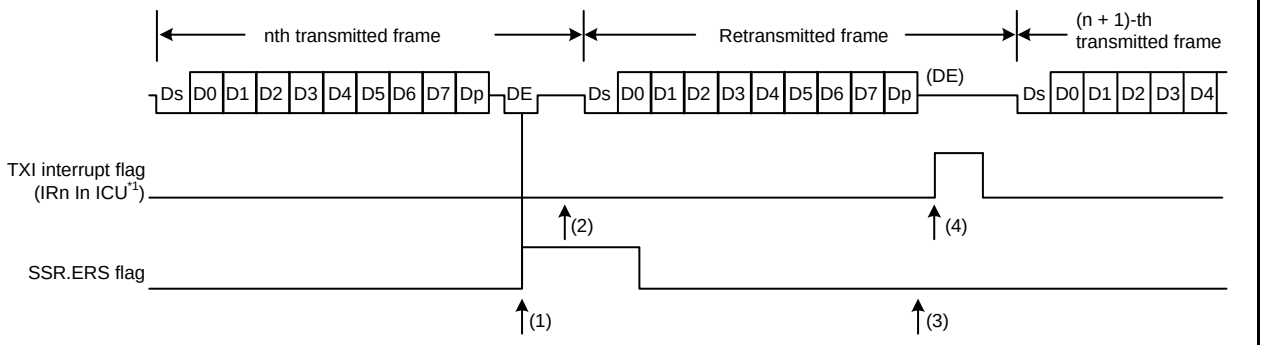


Figure 36.60 Example of Data Transmission Timing in Smart Card Interface Mode

36.7.6 Serial Data Transmission (Except in Block Transfer Mode)

Serial data transmission in smart card interface mode (except in block transfer mode), in that an error signal is sampled and data can be retransmitted, is different from that in non-smart card interface mode. Figure 36.61 shows the data retransmit operation during transmission.

(1) When an error signal from the receiver end is sampled after one-frame data has been transmitted, the ERS flag in the SSR register is set to 1. If the RIE bit in the SCR0 register is 1 at this time, an ERI interrupt request is generated. Clear the ERS flag to 0 before the next parity bit is sampled.
(2) For a frame in which an error signal is received, the TEND flag in the SSR register is not set. Data is retransferred from the TDR register to the TSR register allowing automatic data retransmission.
(3) If no error signal is returned from the receiver, the ERS flag is not set to 1.
(4) In this case, the RSCI judges that transmission of one-frame data (including retransmission) has been completed, and the TEND flag is set. If the TIE bit in the SCR0 register is 1 at this time, a TXI interrupt request is generated. Writing transmit data to the TDR register starts transmission of the next data.



Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 36.61 Data Retransmit Operation in RSCI Transmit Mode

Figure 36.63 shows a sample flowchart of serial transmission. All the processing steps are automatically performed by using a TXI interrupt request to activate the DTC or DMAC. When SSR.TEND flag is set to 1 in transmission, if the SCR0.TIE bit is 1, a TXI interrupt request is generated. The DTC or DMAC is activated by a TXI interrupt request if the TXI interrupt request is specified as a source of DTC or DMAC activation beforehand, allowing transfer of transmit data. The TEND flag is automatically set to 0 when the DTC or DMAC transfers the data. If an error occurs, the RSCI automatically retransmits the same data. During this retransmission, the TEND flag is kept to 0 and the DTC or DMAC is not activated. Therefore, the RSCI 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 RSCI settings. For DTC or DMAC settings, refer to section 18, DMA Controller (DMACAb) and section 20, Data Transfer Controller (DTCb).
Note that SSR.TEND flag is set in different timings depending on the SCR3.GM bit setting. Figure 36.62 shows the TEND flag generation timing.

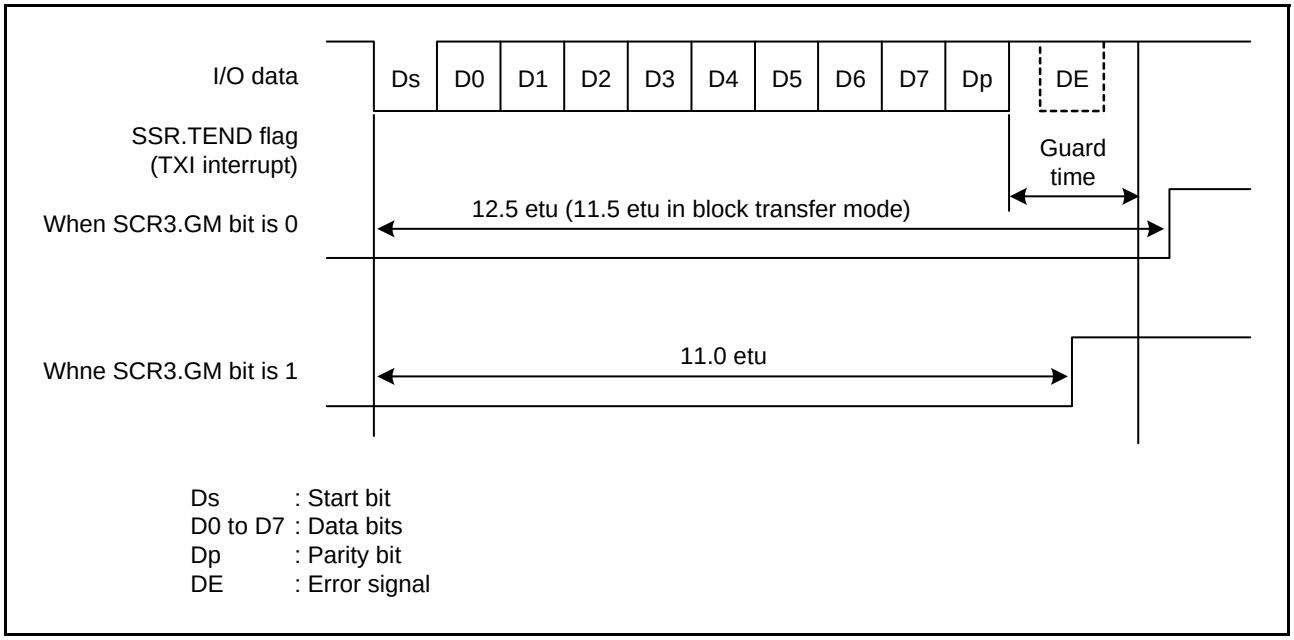


Figure 36.62 SSR.TEND Flag Generation Timing during Transmission

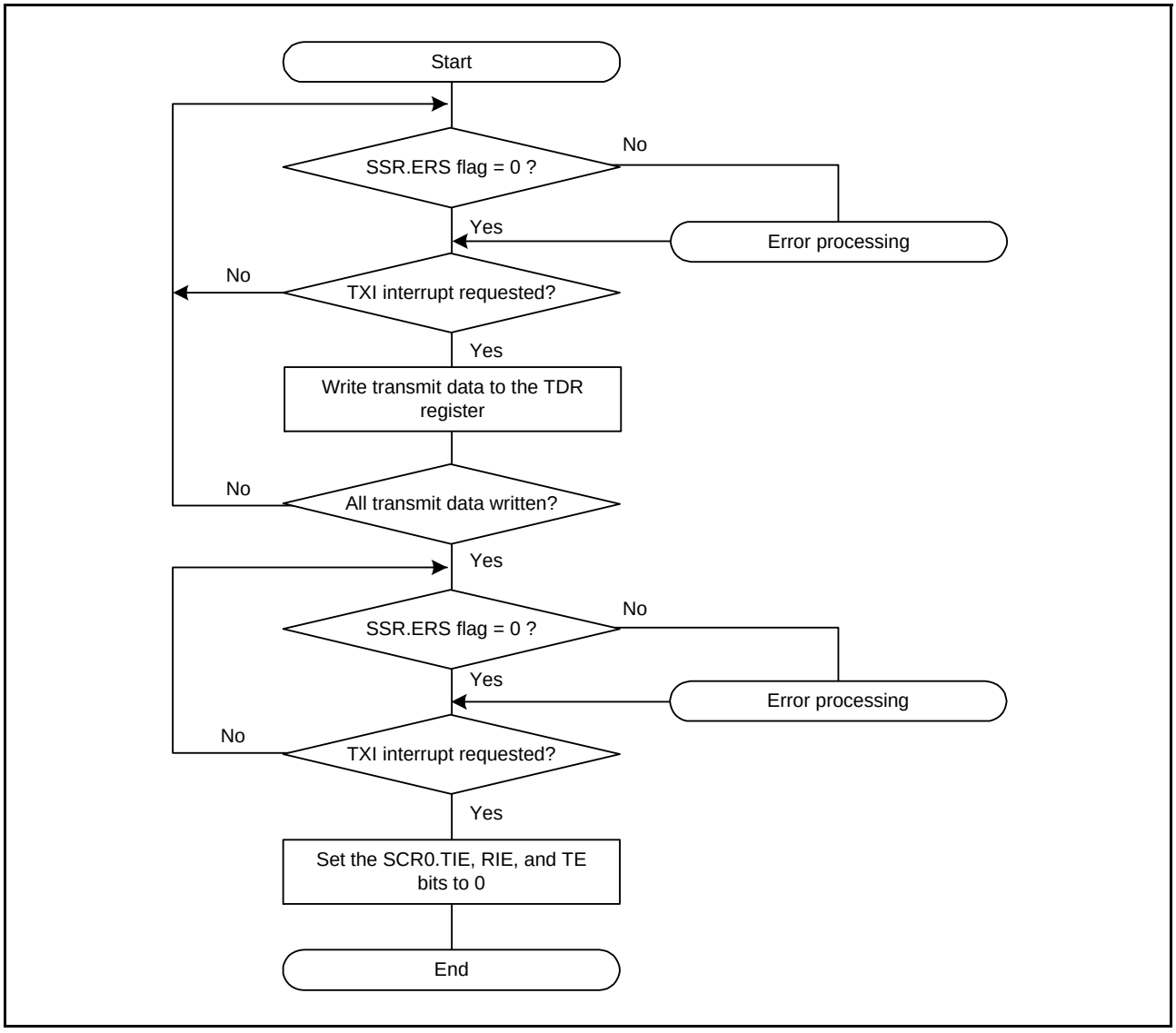
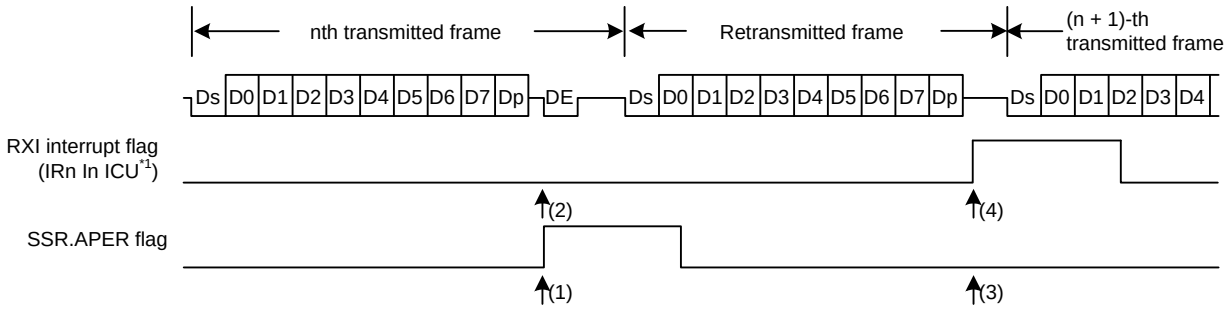


Figure 36.63 Sample Smart Card Interface Transmission Flowchart

36.7.7 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is similar to that in non-smart card interface mode. Figure 36.64 shows the data retransmit operation in receive mode.

(1) If a parity error is detected in receive data, the APER flag in the SSR register is set to 1. When the RIE bit in the SCR0 register is 1 at this time, an ERI interrupt request is generated. Clear the APER 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 APER flag in the SSR register is not set to 1.
(4) In this case, data is determined to have been received successfully. When the RIE bit in the SCR0 register is 1, an RXI interrupt request is generated.



Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 36.64 Data Retransmit Operation in RSCI Receive Mode (Data Retransmit Operation during Reception)

Figure 36.65 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 SSR.ORER or APER flag 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 APER flag is set to 1 during reception, receive data is transferred to RDR register, thus allowing the data to be read. When a reception is forcibly terminated by setting the SCR0.RE bit to 0 during operation, read the RDR register because the received data which has not yet been read may be left in RDR register.

Note: For operations in block transfer mode, refer to section 36.3, Operation in Asynchronous Mode.

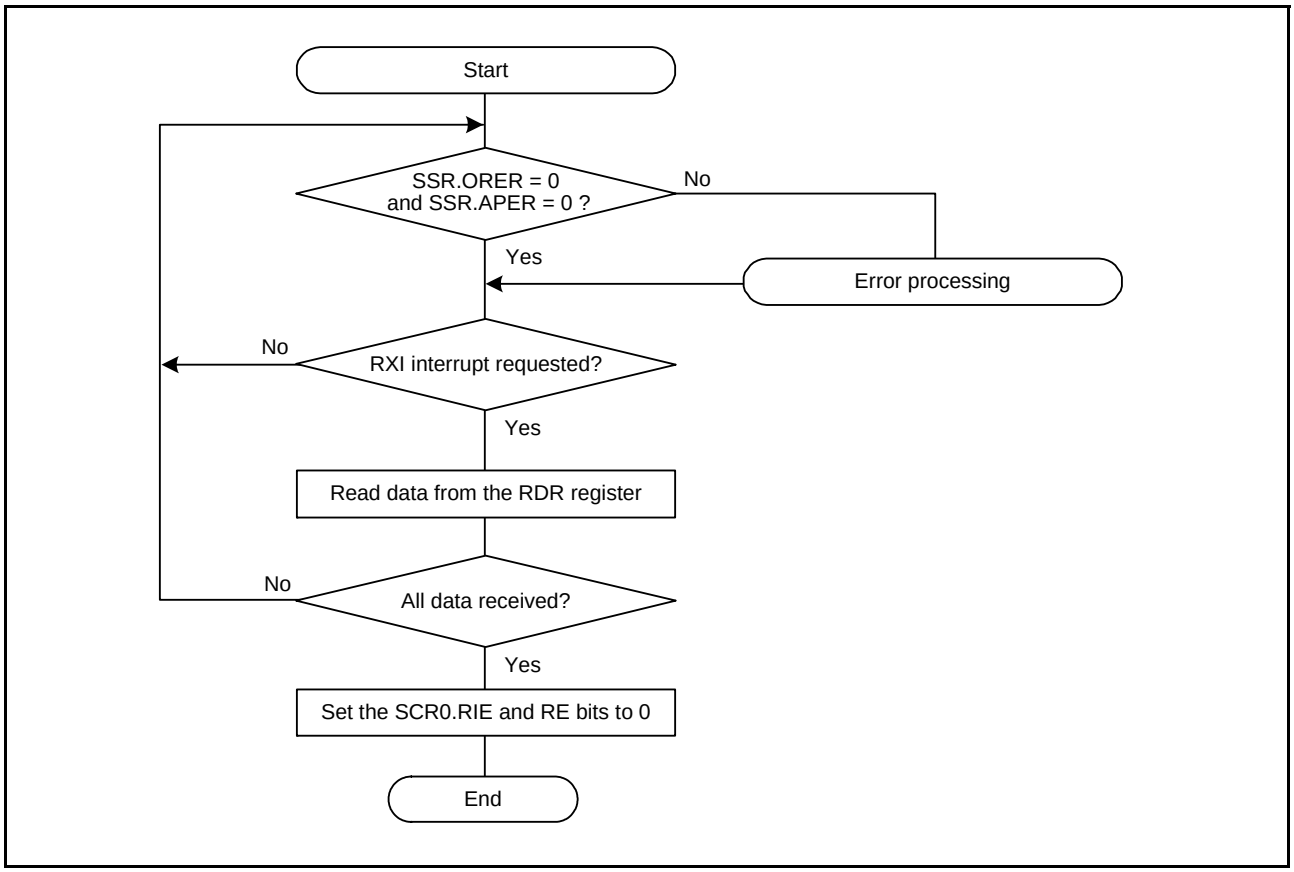


Figure 36.65 Sample Smart Card Interface Reception Flowchart

36.7.8 Clock Output Control

When the SCR3.GM bit is set to 1, the clock output can be controlled by the SCR3.CKE[1:0] bits. Refer to the description of the SCR3.CKE[1:0] bits in section 36.2.8, Control Register 3 (SCR3). When setting the clock output, the base clock described in section 36.7.4, Receive Data Sampling Timing and Reception Margin is output, so the width of the clock pulse can be kept to the width specified by setting the bit rate. It is described in section 36.2.7, Control Register 2 (SCR2), the bit rate is set by SCR2.CKS[1:0] bits, SCR2.BCP[2:0] bits and BRR[7:0] bits.
Figure 36.66 shows the timing chart for explaining clock output control. This is an example when the SCR3.CKE[1] bit is set to 0 and the SCR3.CKE[0] bit is controlled.
When the SCR3.GM bit is 0, output control by the SCR3.CKE[0] bit is immediately reflected on the SCKn pin, so there is a possibility that pulses with an unintended width may be output from the SCKn pin.
When the SCR3.GM bit is 1, the output pulse control by the SCR3.CKE[0] bit controls the pulse width set to be based on the state of the base clock.

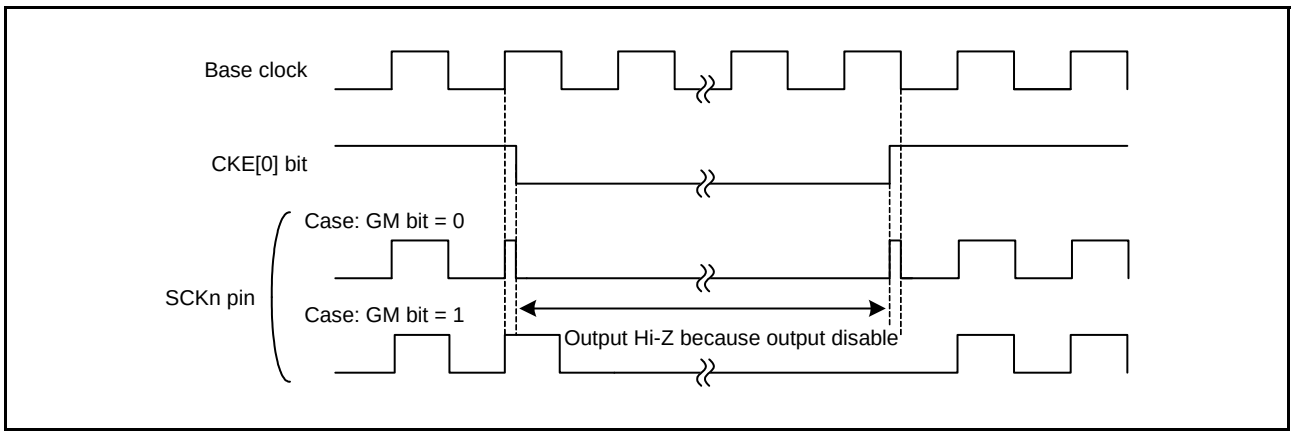


Figure 36.66 Clock Control Timing Chart by SCR3.GM Bit

36.8 Extended Serial Mode

36.8.1 Serial Transfer Protocol

As an extended function of the RSCI, the RSCI supports the serial communication protocol (Figure 36.67) consisting of a Start Frame and an Information Frame. Extended serial mode is enabled by the SCR3.MOD[2:0] bits = 110b. Since the extended serial mode uses the same circuit as the asynchronous mode for transmission/reception control other than Break Field, the basic communication settings required for the asynchronous mode are also required for the extended serial mode (however, set the SCR3.RXDESEL bit to 1).
The Start Frame consists of a Break Field, Control Field 0, and Control Field 1. The Information Frame can be configured with some Data Fields, a CRC16 Upper Field, and a CRC16 Lower Field.

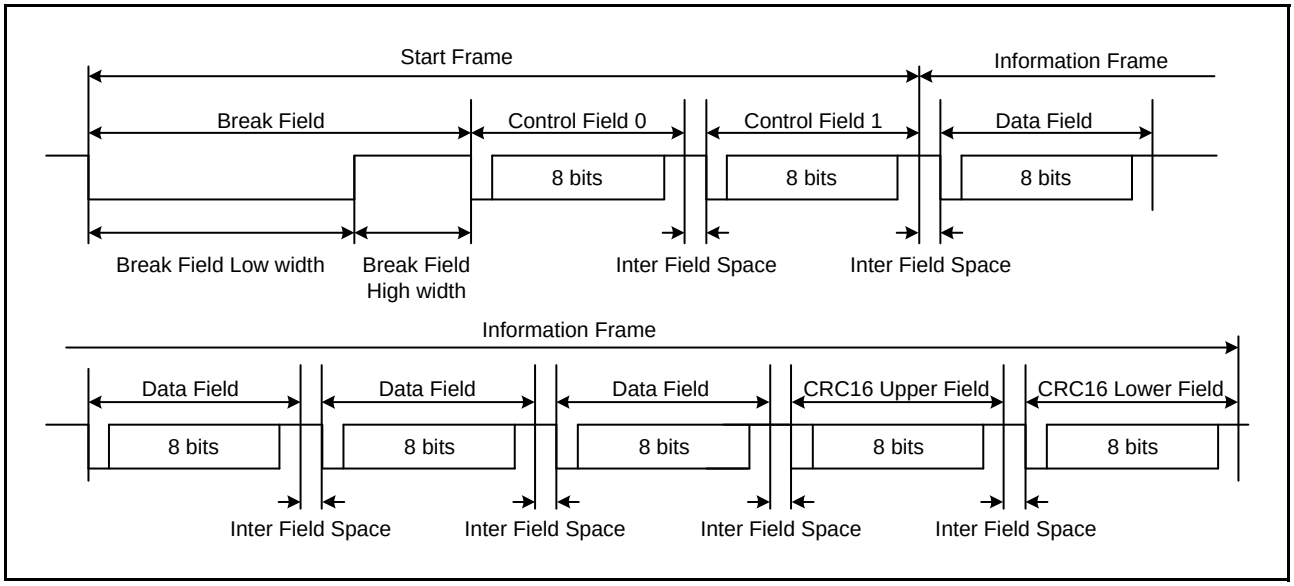


Figure 36.67 Protocol for Serial Transfer by the Extended Serial Mode

The following describes operations in extended serial mode. In this section, operations are described with the following conditions:
Communication pin (RXDn/TXDn) level inversion function: OFF (RINV bit = TINV bit = 0)
When using the communication pin (RXDn/TXDn) level inversion function enabled, replace the RXD and TXD signal levels with their inverted levels.

36.8.2 Transmitting a Start Frame

Figure 36.68 shows an example of transmission of the Start Frame consisting of a Break Field, Control Field 0, and Control Filed 1. (Omit Break Field and Control Field 0 according to the Start Frame configuration.)
Figure 36.69 shows a flowchart for Start Frame transmission.
The RSCI operates as follows during Start Frame transmission.

(1) Make the initial settings for the RSCI according to the RSCI initialization flow (Figure 36.8) in asynchronous mode. In extended serial mode, do not set SCR0.TE and TIE bits to 1 at the same time to avoid TXI output before the Break Field. Therefore, perform the following two steps sequentially to set the RSCI initialization flow (asynchronous mode) procedure [10].

- Set the bits except SCR0.TIE bit. (SCR0.TIE bit = 0, SCR0.TE bit = 1, and SCR0.RE bit = 0)
- Set SCR0.TIE bit to 1.

(2) When 1 is written to TCST, the timer in the extended serial module starts counting and outputs a low level (Break Field) from the TXDn pin for the period set in XCR2.BFLW[15:0] bits. A timer count clock source can be selected by XCR0.TCSS[1:0] bits.
Writing 0 to XCR1.TCST bit suspends output of the Break Field. After the suspension, set SCR0.TE bit = 0 and turn off the transmission.

(3) When the extended serial module timer count value matches the set XCR2.BFLW[15:0] bits value, the timer stops counting and inverts the TXDn pin output level, and the XSR0.BFOF flag is set to 1*1. Furthermore, if XCR0.BFOIE bit has been set to 1 at this time, a TXI interrupt is generated.

(4) After confirming the BFOF flag is set to 1, send the Control Field 0 data.*2

(5) After the Control Field 0 data has been transmitted, write the Control Field 1 data to TDR. And it is transmitted.

(6) After the Control Field 1 data has been transmitted, the Information Frame data is transmitted.

Note 1. After XSR0.BFOF flag is set to 1, if 1 is written to XCR1.TCST bit without clearing it, no TXI interrupt is output at the end of Break Field transmission. Clear XSR0.BFOF flag before writing 1 to XCR1.TCST bit.

Note 2. LIN communication requires a Break delimiter (IDLE period) of 1 bit or more from the end of Break Field transmission until the next data transmission starts. For this reason, the Break delimiter length is counted upon completion of Break Field transmission. If transmit data is written while the Break delimiter length is being counted, transmission does not start until the Break delimiter length counting is completed. When transmit data is written after the Break delimiter length has been counted, transmission starts at the same timing as normal data transmission.
Break delimiter length count time after Break Field transmission: 1-bit to 2-bit length (SCR3.STOP bit = 0) 2-bit to 3-bit length (SCR3.STOP bit = 1)

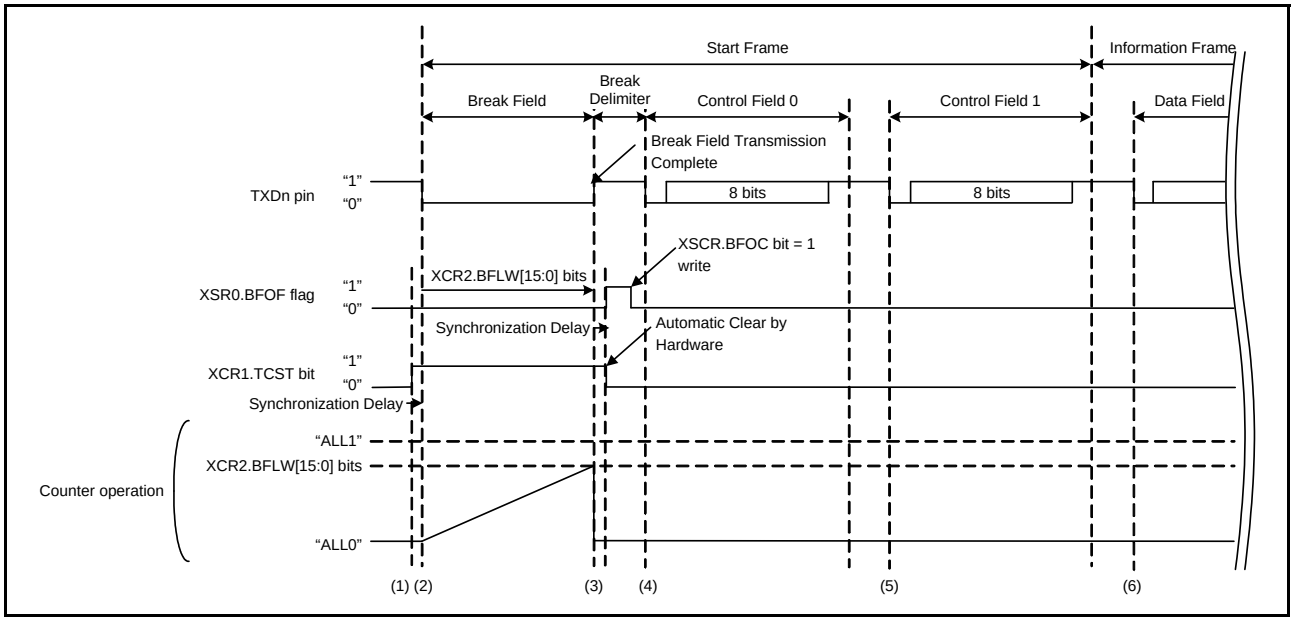


Figure 36.68 Example of Operations When Transmitting a Start Frame

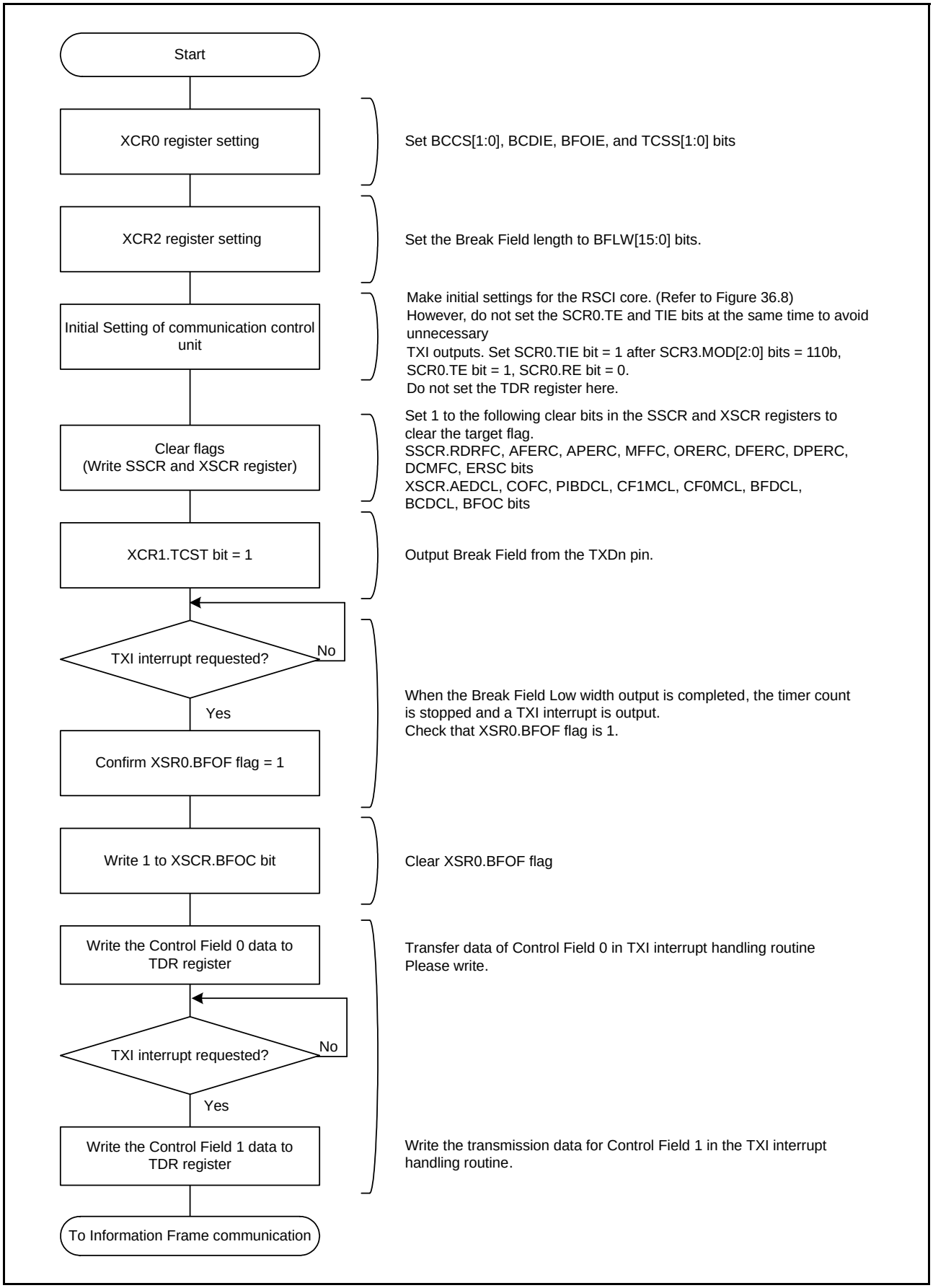


Figure 36.69 Example of Start Frame Transmission

36.8.3 Receiving a Start Frame

The extended serial mode control section is capable of receiving Start Frames with the structures listed in Table 36.39.

Table 36.39 Structures of Start Frames

XCR0		Start Frame Configuration
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

36.8.3.1 Extended Serial Normal Reception of Start Frame (PIB not Used)

Figure 36.70 shows an example of normal reception of the Start Frame consisting of a Break Field, Control Field0, and Control Field1. Figure 36.71 shows an example of reception to detect the Break Field during Control Field 1. Figure 36.72 shows a flowchart to receive the Start Frame, and Figure 36.73 shows a state transition diagram.
When receiving the Start Frame, the RSCI operates as follows. Omit the processing of Break Field and Control Field0 according to the Start Frame configuration.

(1) Writing 1 to XCR1.SDST bit makes it possible to detect the Start Frame. When XCR0.BFE bit = 1, RXD input to the RSCI core is disabled until the Break Field is detected (because XSR0.RXDSF flag is set to 1). Once the Break Field is detected, RXD input can be received to the RSCI core (XSR0.RXDSF flag = 0).
(2) When a low level is input from the RXDn pin, the Break Field detection count starts. A timer count clock source can be selected by XCR0.TCSS[1:0] bits.
(3) When a low level that is equal to or longer than the period set in XCR2.BFLW[15:0] bits is input from the RXDn pin, it is determined as Break Field. At this time, XSR0.BFDF flag is set to 1. If XCR0.BFDIE bit has been set to 1 at this time, a BFD interrupt is generated.
The timer continues counting until the RXD rising edge or counter overflow.
(4) After the Break Field is detected, when the input level from the RXDn pin becomes high, the count value is captured to XSR1.CCV[15:0] bits when XCR1.BRME bit = 0. At this time, XSR0.RXDSF flag is cleared to 0 and the RSCI core starts receiving the RXD input.
(5) The RSCI core starts receiving Control Field 0. Because the extended serial continuously counts the edge interval, it determines a low level that is equal to or longer than the period set in XCR2.BFLW[15:0] bits as detection of the Break Field. When the Break Field is detected in the Control Field 0 phase, the RSCI core waits for reception of Control Field 0 again (Figure 36.71).
(6) When Control Field 0 has been received, an RXI interrupt is generated and the Control Field 0 data is stored in XSR0.CF0RD[7:0] bits. When the received data matches the set XCR2.CF0D[7:0] bits value, XSR0.CF0MF flag is set to 1. If the received data differs from the set XCR2.CF0D[7:0] bits value, the RSCI transitions to the state before the Break Field is detected.
(7) The RSCI core starts receiving Control Field 1. When BFE bit = 1, the Break Field detection function is continuously enabled while SDST bit = 1 as in the case of Control Field 0. When the Break Field is detected in the Control Field 1 phase, the RSCI core waits for reception of Control Field 0 again.
(8) When Control Field 1 has been received, an RXI interrupt is generated and the Control Field 1 data is stored in XSR0.CF1RD[7:0] bits. When the received data matches the set XCR1.PCF1D[7:0] bits value or the set

XCR1.SCF1D[7:0] bits value, XSR0.CF1MF flag is set to 1. If the received Control Field 1 data matches neither the set XCR1.PCF1D[7:0] bits value nor the set XCR1.SCF1D[7:0] bits value, the RSCI transitions to the state before the Break Field is detected.

(9) The RSCI core performs Information Frame communication.

(10) When communication is completed, write 0 to XCR1.SDST bit and 0 to SCR0.RE bit to stop reception.

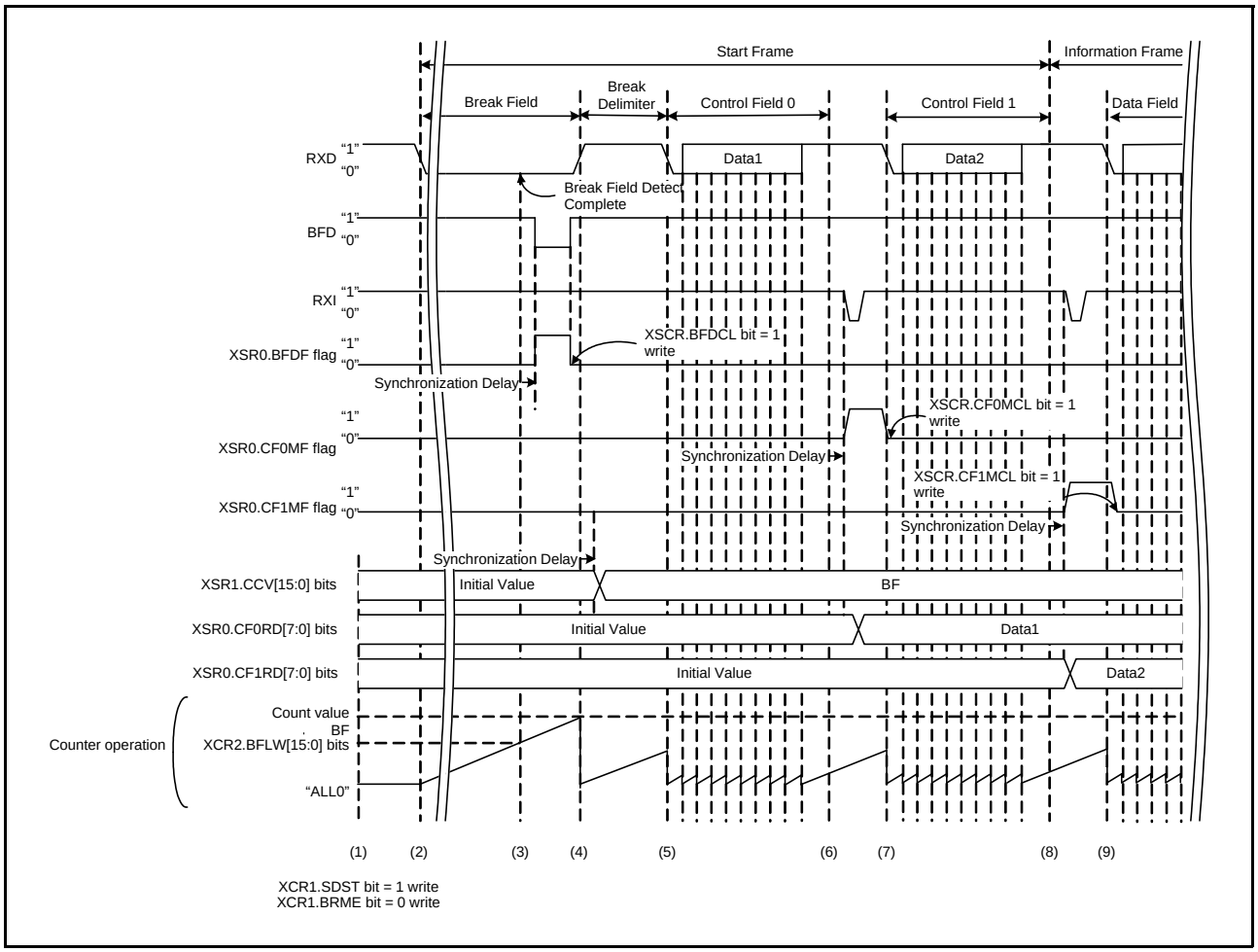


Figure 36.70 Normal Reception Example of Start Frame (PIB Not Used)

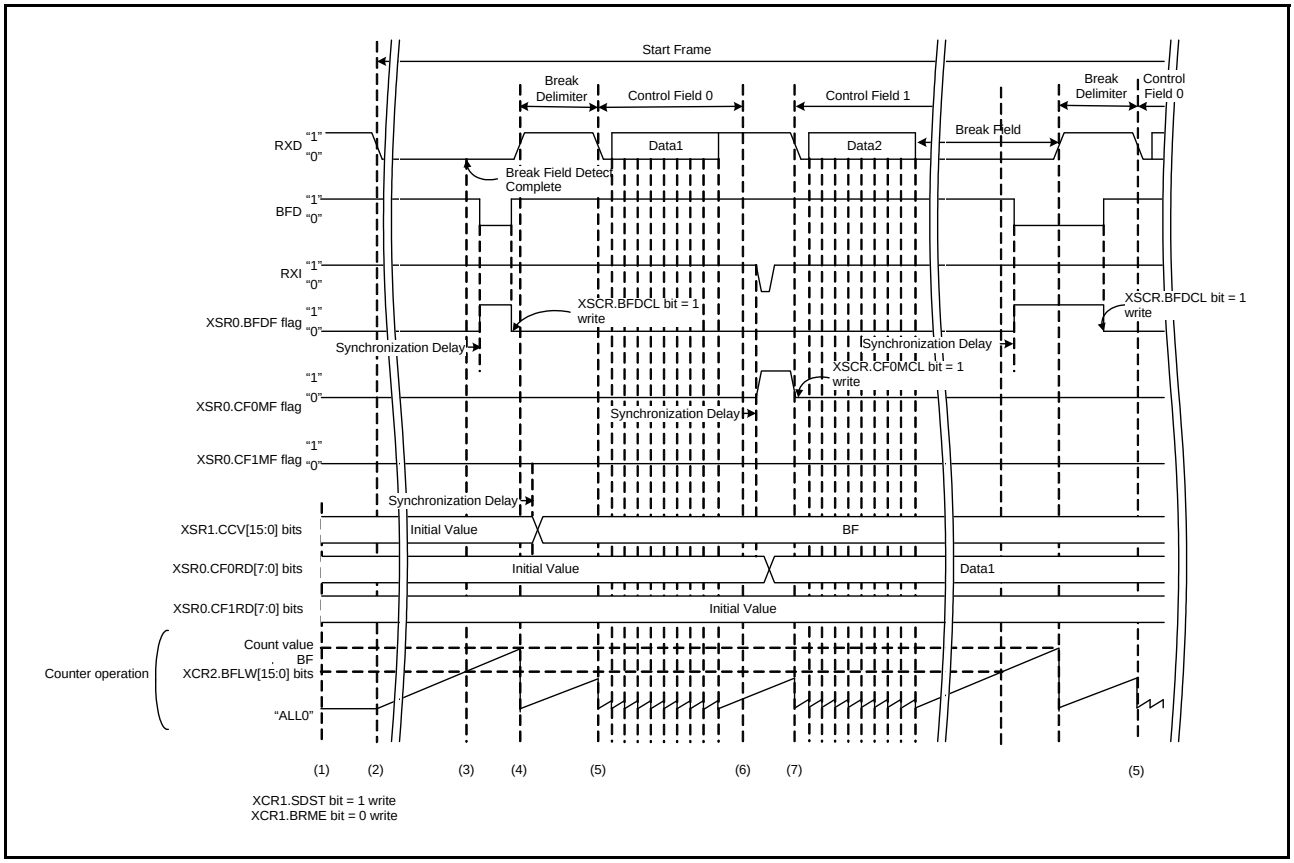


Figure 36.71 Start Frame Reception Example (PIB Not Used) Break Field Detected during Control Field 1

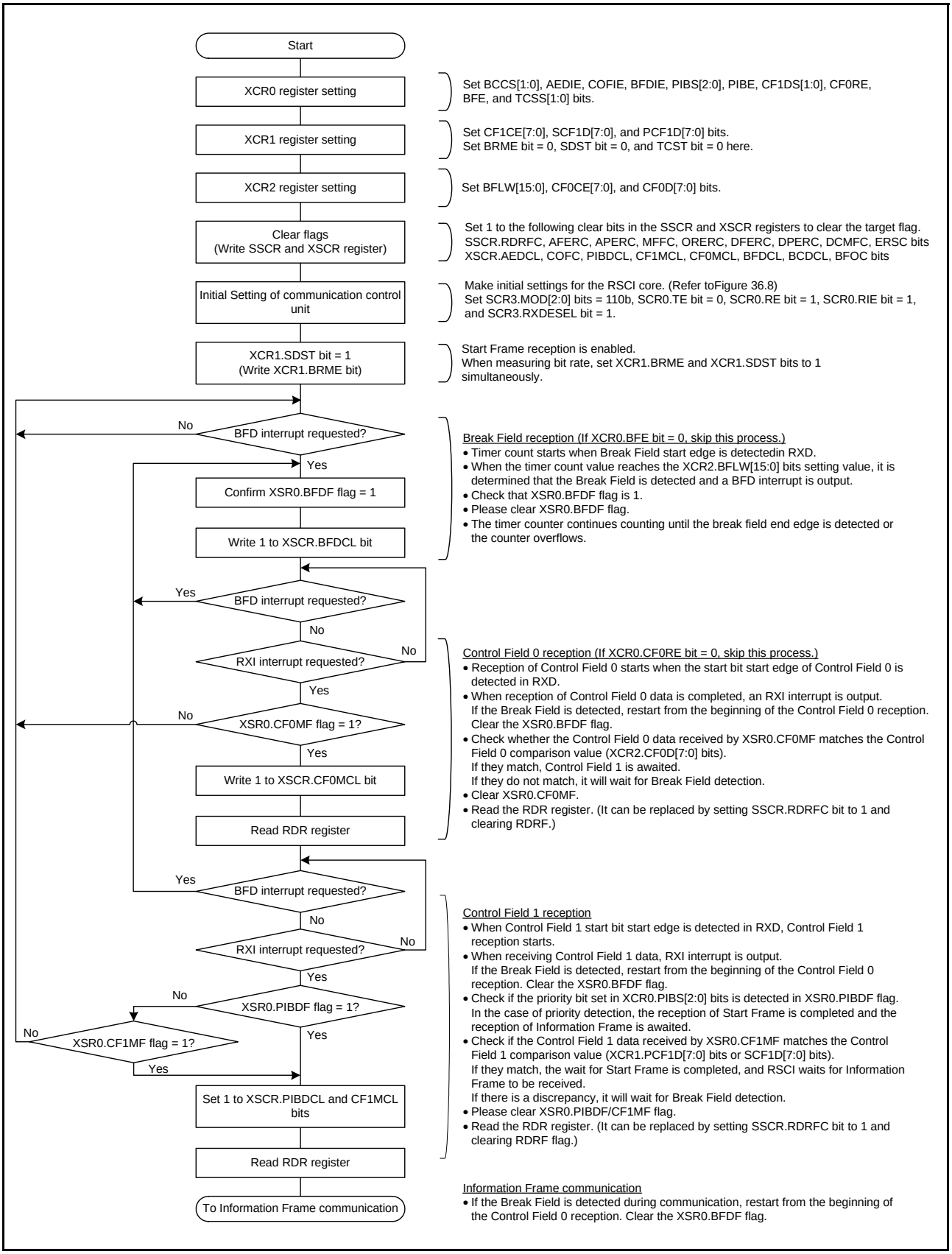


Figure 36.72 Sample Flowchart for Reception of a Start Frame

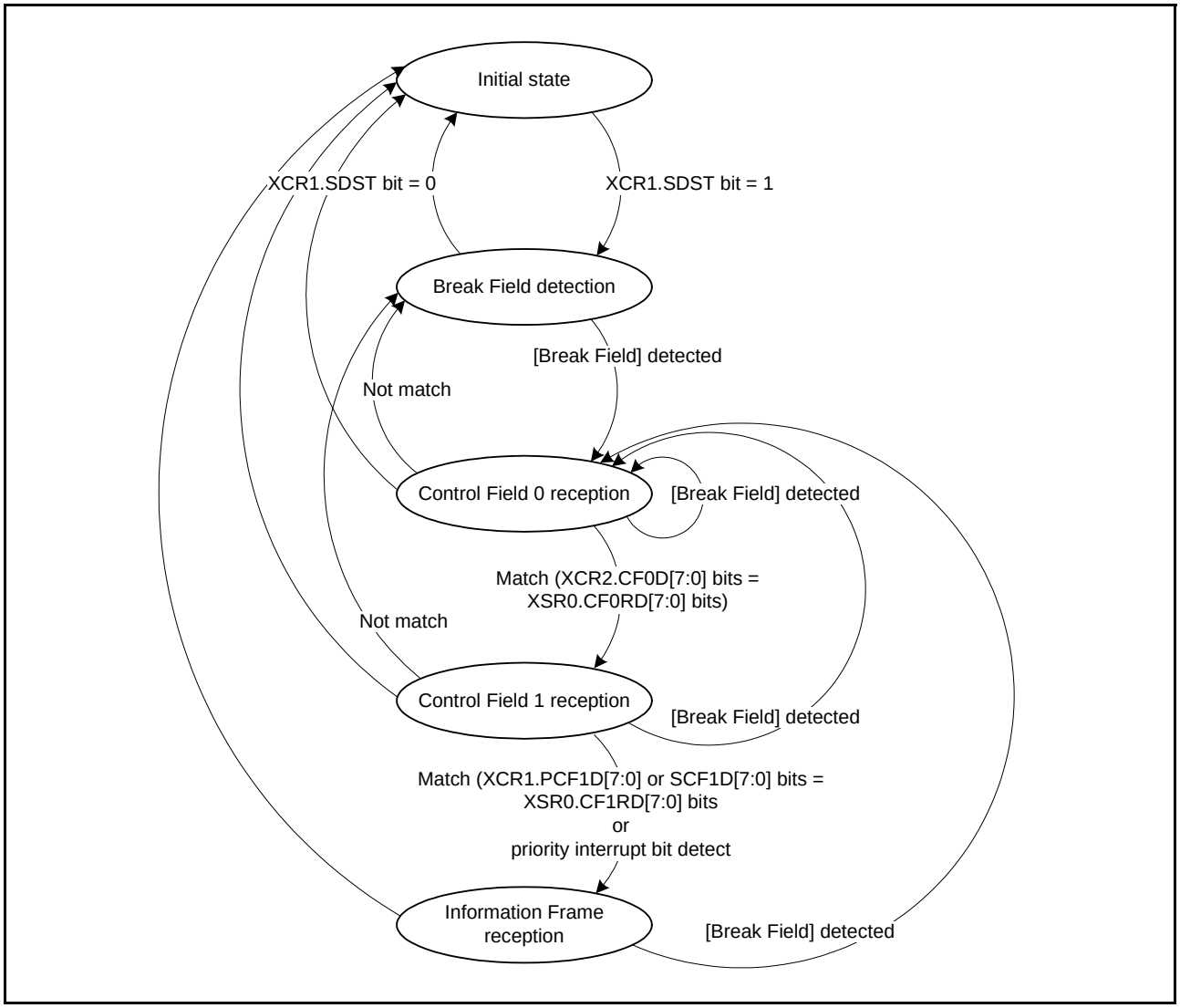


Figure 36.73 State Transition Diagram of Start Frame Reception

36.8.3.2 Priority Interrupt Bit

Figure 36.74 shows an example of Start Frame reception using the priority interrupt bit. The priority interrupt bit is enabled by setting XCR0.PIBE bit to 1.

The RSCI operates as follows during Start Frame reception using the priority interrupt bit.

Steps (1) to (7) are the same as in Figure 36.70, for Start Frame reception.

(8) When the value specified in the XCR0.PIBS[2:0] bits matches the set XCR1.PCF1D[7:0] bits value, XSR0.PIBDF flag is set to 1and the RSCI core performs communication of the Information Frame. If the data received in Control Field 1 matches neither the set XCR1.PCF1D[7:0] bits value nor the set XCR1.SCF1D[7:0] bits value and the priority interrupt bit is not detected, the RSCI transitions to the state before the Break Field is detected.

(9) The RSCI core performs Information Frame communication.

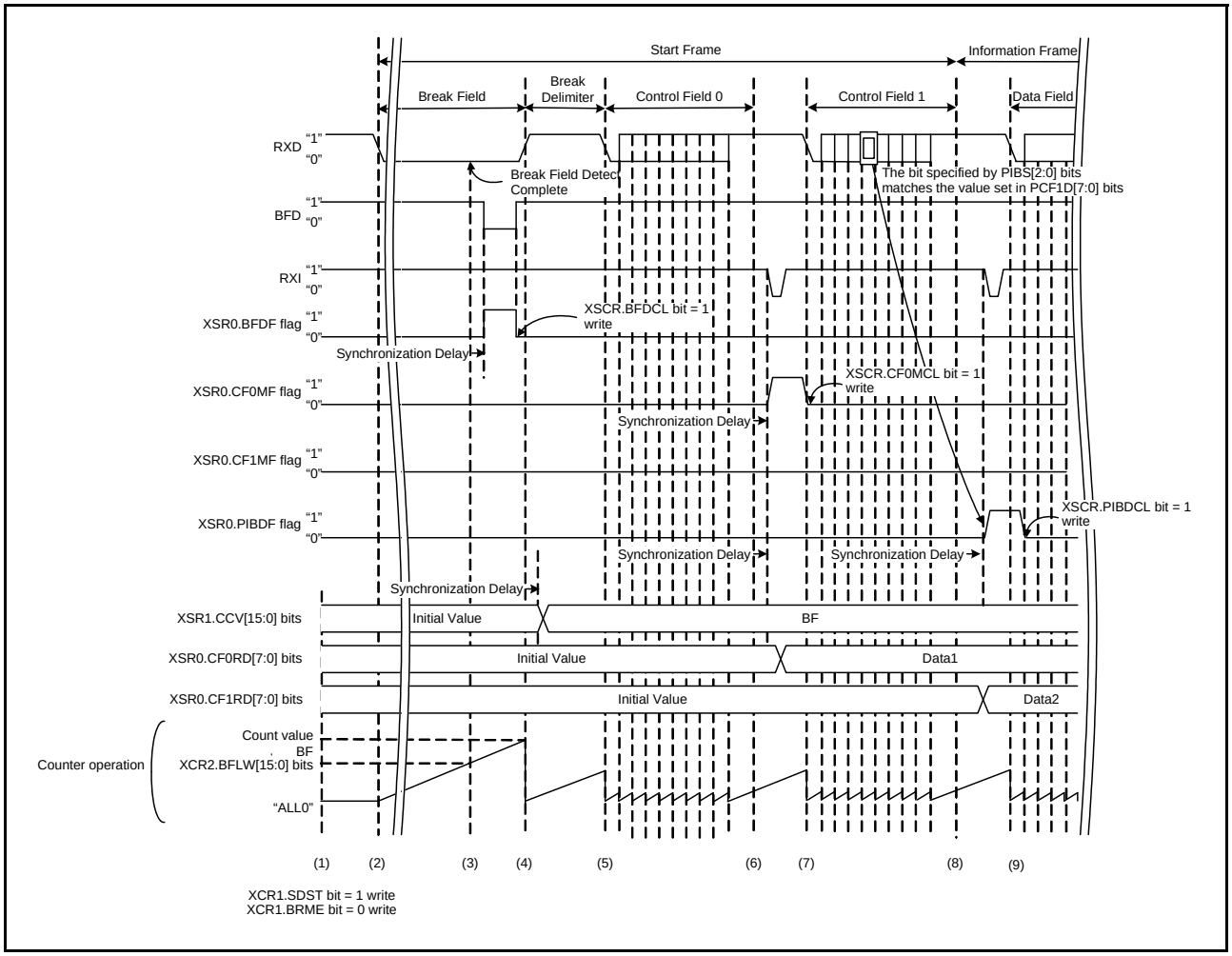


Figure 36.74 Start Frame Reception Example (Priority Interrupt Bit Used)

36.8.4 Detection of Bus Collisions

In extended serial mode (SCR3MOD[2:0] bits = 110b) when TE bit = 1, the bus conflict detection function works during Break Field transmission and during data transmission.

Figure 36.75 shows an operation example of the bus conflict detection function. The TXDn pin output and the RXDn pin input are sampled by the bus conflict detection clock set in XCR0.BCCS[1:0] bits. When a mismatch occurs three times in a row, XSR0.BCDF flag is set to 1, and if XCR0.BCDIE bit has been set to 1 at this time, an ERI interrupt is generated.

When an ERI interrupt is generated, stop transmission according to Figure 36.76. Check the bus state to decide whether to resume transmission.

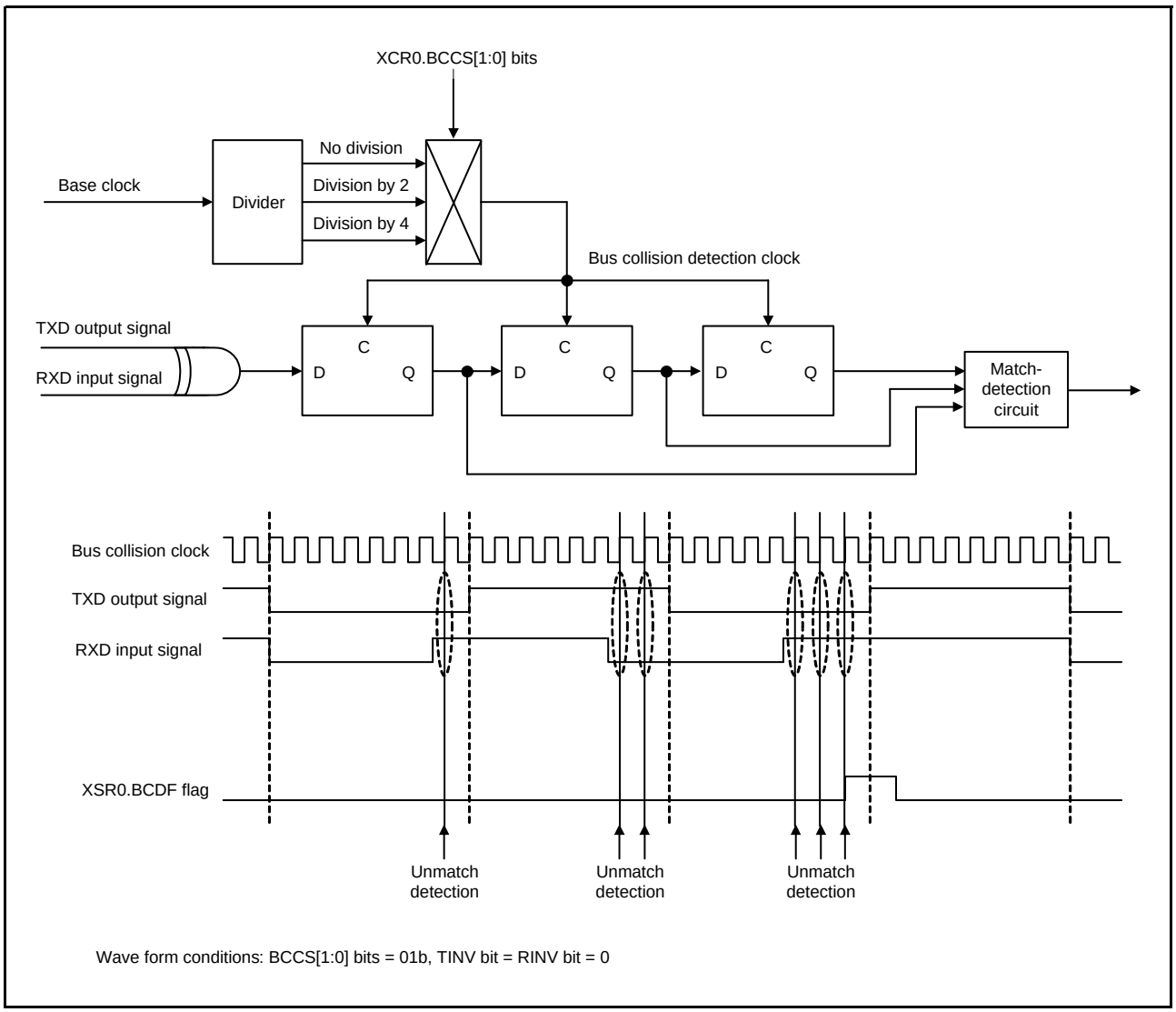


Figure 36.75 Example of Operations with Bus Collision Detection

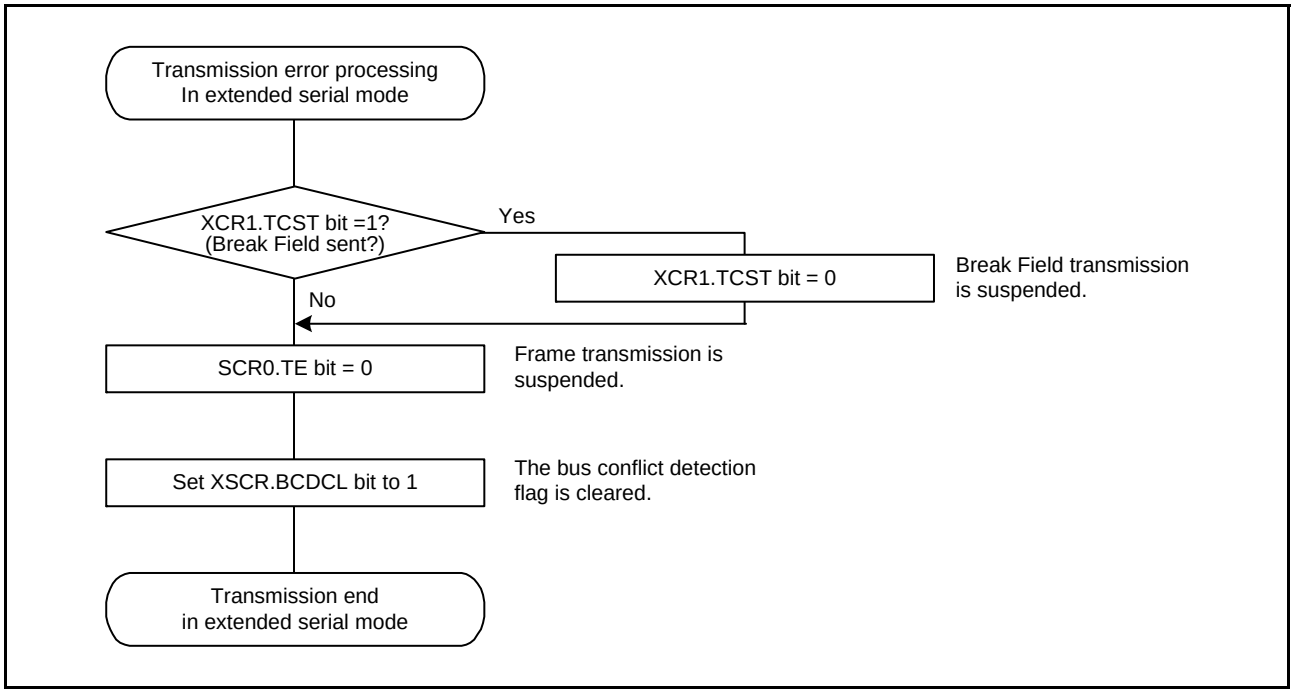


Figure 36.76 ERI Interrupt Handling Flow at Transmission in Extended Serial Mode

36.8.5 Bit Rate Measurement

This function measures a bit rate between the effective edges of the input signal from the RXDn pin. Figure 36.77 shows an operation example of the bit rate measurement function.

(1) Writing 1 to XCR1.SDST and XCR1.BRME bits enables bit rate measurement. When this bit is set to 1, the valid edge interval of Control Field 0 and Control Field 1 data is measured. However, bit rate is not measured between the Break Field and the Break Delimiter. Set XCR1.BRME and XCR1.SDST bits to 1 simultaneously, only when measuring bit rate.
(2) Because bit rate is not measured in the Break Field, the effective edge detection flag is not set to 1 at the rising edge at the end of the Break Field, and the counter capture value is not stored in XSR1.CCV[15:0] bits.
(3) The counter starts counting from the falling edge of the start bit in Control Field 0. The Break Delimiter count value is not captured in XSR1.CCV[15:0] bits.
(4) The rising edge of the start bit is detected as an effective edge, and then the XSR0.AEDF flag is set to 1. If XCR0.AEDIE bit has been set to 1 at this time, an AED interrupt is output. The start bit count value is stored in XSR1.CCV[15:0] bits. The XSR1.CCV[15:0] value is retained until the effective capture value is read.
(5) Even if an effective edge is input from the RXD input pin, the count value of this effective edge timing is not captured because the XRS1.CCV[15:0] bits value has not been read and retention has not been released. In this case, an AED interrupt is not output.
(6) The XSR1.CCV[15:0] value is read. Then the retention of XSR1.CCV[15:0] bits is released and the XSR0.AEDF flag is cleared by hardware.
(7) Because the retention of XSR1.CCV[15:0] bits has been released, the count value is captured at the effective edge and is retained. At the same time, the XSR0.AEDF flag is set to 1, and if XCR0.AEDIE bit has been set to 1, an AED interrupt is output. The bit rate can be adjusted by calculating it from the count value between effective edges by software and by changing the RSCI settings.
(8) To disable bit rate measurement, write 0 to XCR1.BRME bit.
(9) The XSR0.AEDF value and the XSR1.CCV[15:0] value remain unchanged at the effective edge timing because the bit rate measurement function is disabled.

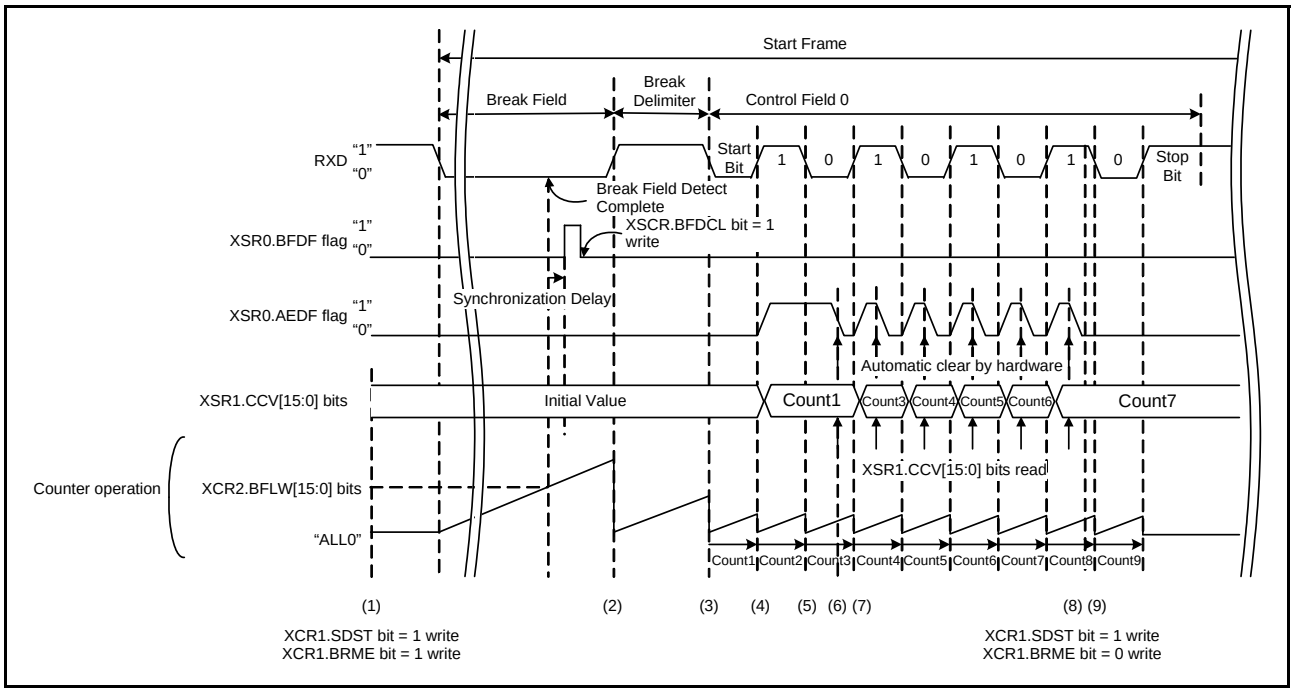


Figure 36.77 Operation Example of the Bit Rate Measurement Function

36.9 Operation in Simple I²C Mode

Simple I²C-bus format is composed of 8 data bits and an acknowledge bit. The frame following the start condition or restart condition is the slave-address frame, which is used by the master device to specify a slave device with which it is communicating. The currently specified slave device remains valid until a new slave device is specified or a stop condition is satisfied. The 8-bit data of each frame is transmitted from MSB.
Figure 36.78 shows I²C bus format, and Figure 36.79 shows the timing of I²C.

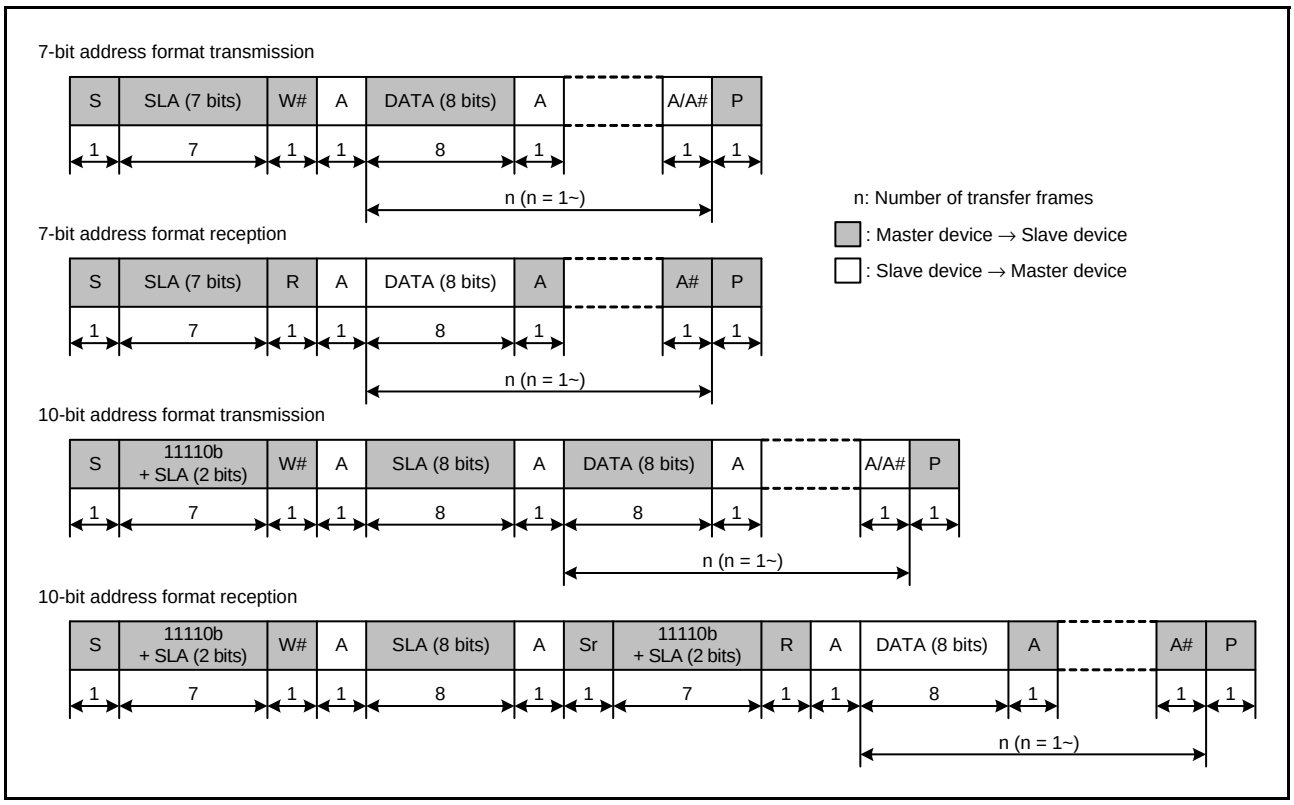


Figure 36.78 I²C-Bus Format

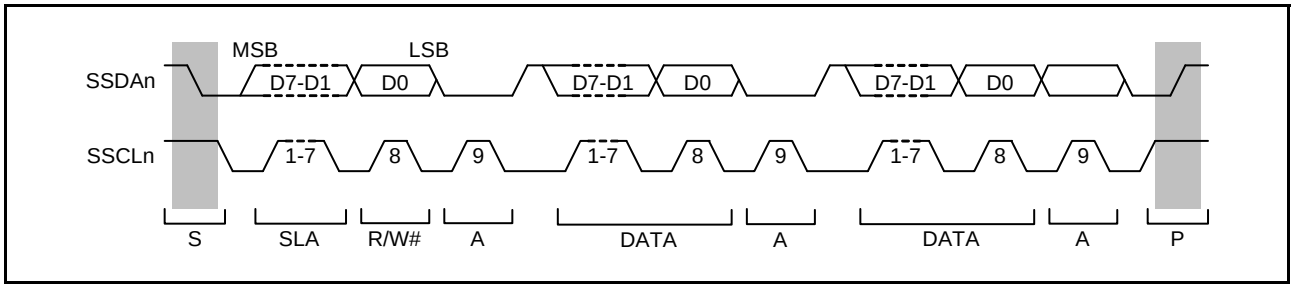


Figure 36.79 I²C-Bus Timing (When SLA is 7 Bits)

S: Indicates a start condition. The master device changing the level on the SSDAn line from the high to the low level when the SSCLn line is at the high level.
SLA: Indicates a slave address. The master device selects a slave device.
R/W#: Indicates the direction of transfer (reception or transmission). When R/W# is high, the transfer direction is from the slave device to the master device. When R/W# is low, the transfer direction is from the master device to the slave device.
A/A#: Indicates an acknowledge. This is returned by the slave device for master transmission and by the master device for master reception. The low level indicates ACK and the high level indicates NACK.
Sr: Indicates a restart condition. The master device changing the level on the SSDAn line from the high to the low level when the SSCLn line is at the high level.
DATA: Indicates the received or transmitted data.
P: Indicates a stop condition. The master device changing the level on the SSDAn line from the low to the high level when the SSCLn line is at the high level.

36.9.1 Generation of Start, Restart, and Stop Conditions

Writing 1 to SIMR.IICSTAREQ bit causes the generation of a start condition. The following operations are done at the generation of a start condition.

- 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 SCR2.BRR[7:0] bits.
- The level on the SSCLn line falls (from the high level to the low level), the SIMR.IICSTAREQ bit is set (to 0), and a start-condition generated interrupt is output.

Writing 1 to SIMR.IICRSTAREQ bit causes the generation of a restart condition. The following operations are done at the generation of a restart condition.

- 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 SCR2.BRR[7:0] bits.
- 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 SCR2.BRR[7:0] bits.
- 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 SCR2.BRR[7:0] bits.
- The level on the SSCLn line falls (from the high level to the low level), the SIMR.IICRSTAREQ bit is set (to 0), and a restart-condition generated interrupt is output.

Writing 1 to SIMR.IICSTPREQ bit causes the generation of a stop condition. The following operations are done at the generation of a stop condition.

- 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 SCR2.BRR[7:0] bits.
- 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 SCR2.BRR[7:0] bits.
- The SSDAn is released (transition from the low to the high level), the SIMR.IICSTPREQ bit is set (to 0), and a stop condition generated interrupt is output.

Figure 36.80 shows the timing of operations in the generation of start, restart and stop conditions.

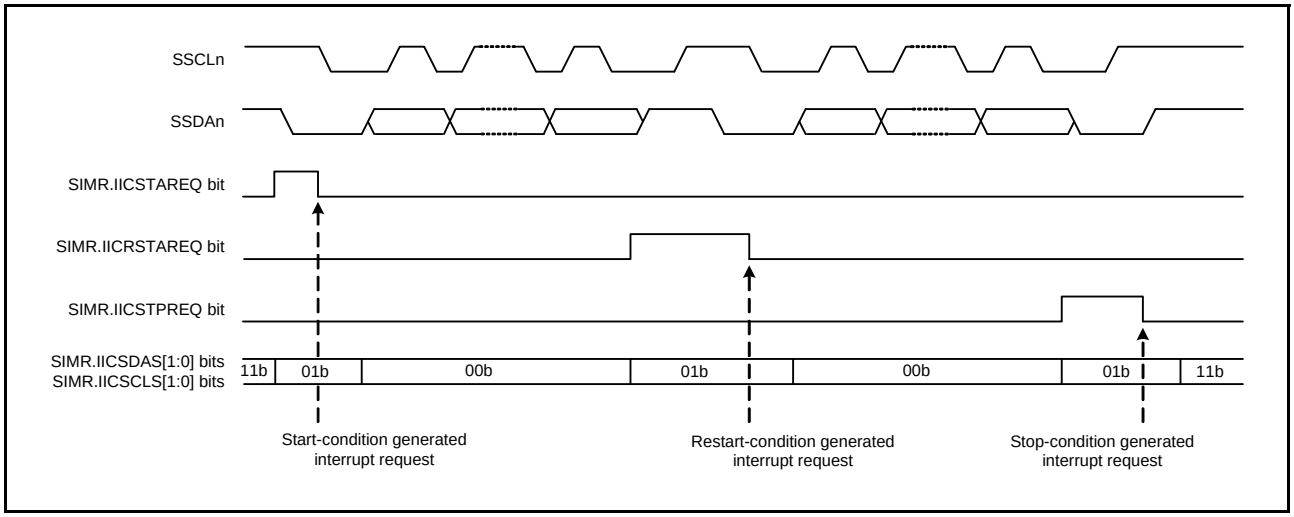


Figure 36.80 Timing of Operations in the Generation of Start, Restart, and Stop Conditions

36.9.2 Clock Synchronization

The slave device of the communication partner may make SSCLn line Low-level with a view to insert a wait. Setting the SIMR.IICCSC bit to 1, applies control to obtain synchronization when the levels of the internal SCL signal and the level being input on the SSCLn pin differ.

When the SIMR.IICCSC bit is set to 1, the level of the internal SCL 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 until counting to determine the period at high level starts on the transition of the SSCLn pin to the high level is the total time which contains the SSCLn input delay, the noise filtering delay of the SSCLn pin (2 or 3 cycles of the filtering clock), and the internal processing delay (1 or 2 cycles of PCLKA). The period at high level of the internal SCL signal is extended even if other devices are not placing the low level on the SSCLn line.

If the SIMR.IICCSC bit 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 SCL signal. If the SIMR.IICCSC bit is 0, synchronization with the internal SCL signal 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 SCL 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 SCL 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 36.81 shows an example of operations to synchronize the clocks.

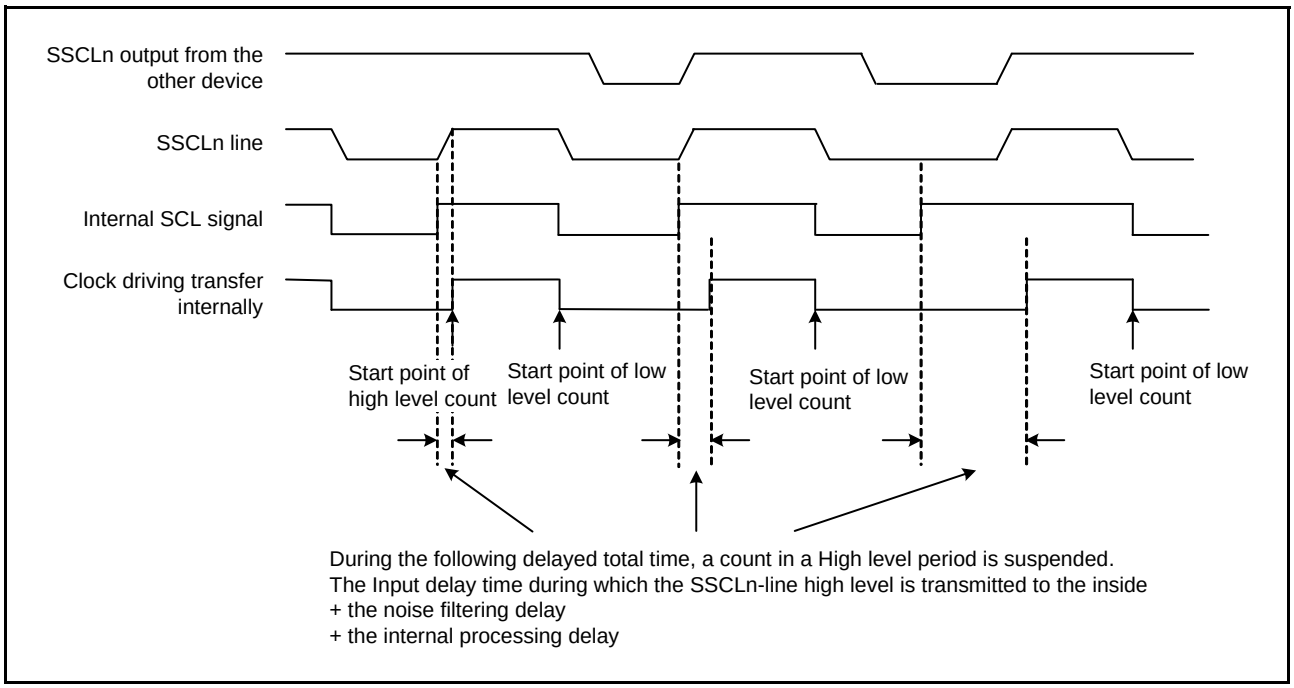


Figure 36.81 Example of Operations for Clock Synchronization

36.9.3 SDA Output Delay

The SIMR.IICDL[4:0] bits can be used to set a delay for output on the SSDAn pin relative to falling edges of output on the SSCLn pin. The delay-time settings are selectable from 0 to 31 cycles of the clock signal from the on-chip baud rate generator (The base is PCLKA and selected the divided clock by the SCR2.CKS[1:0] bits). About Start/Restart/Stop conditions, 8bit-transmission data and acknowledge, the SSDAn pin output can be delayed.

If the SDA output delay is shorter than the falling time of the SSCLn output pin, the change of the SSDAn output pin will start while the SSCLn output pin level is falling, then there is a possibility of erroneous operation for slave devices. Ensure setting the SDA output delay greater than the SSCLn maximum falling time. (300 ns for I^2C normal/fast mode.)

Figure 36.82 shows the timing of delays in SDA output.

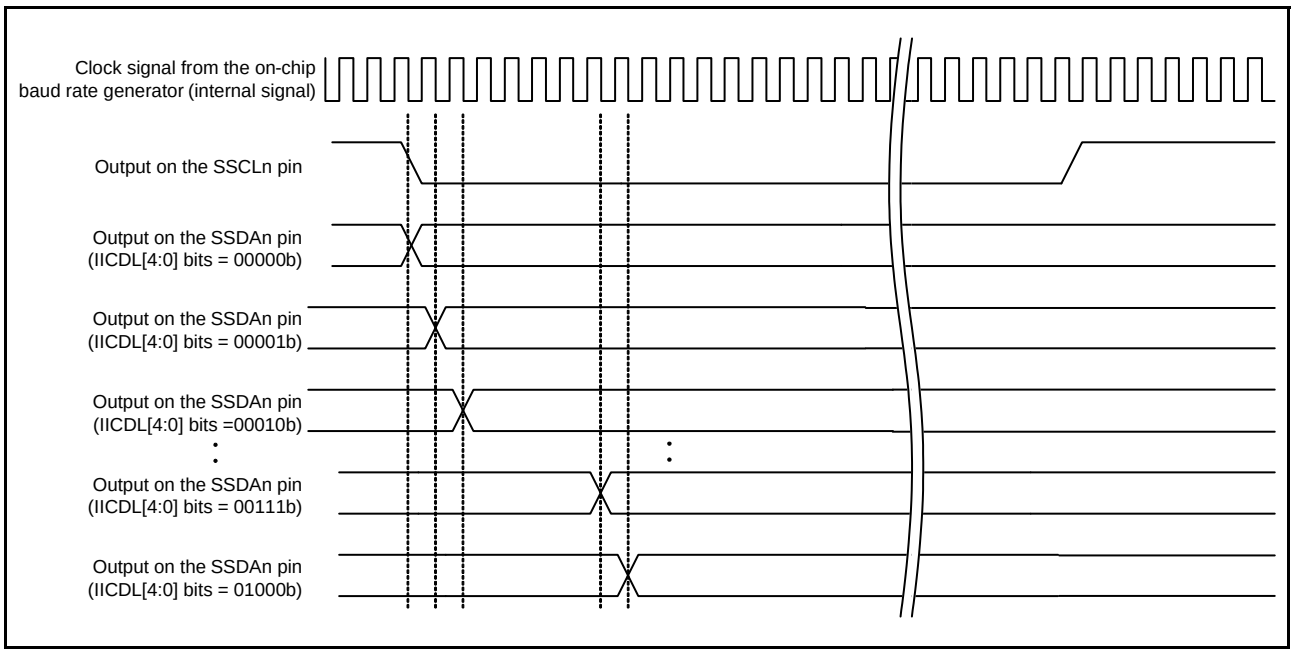
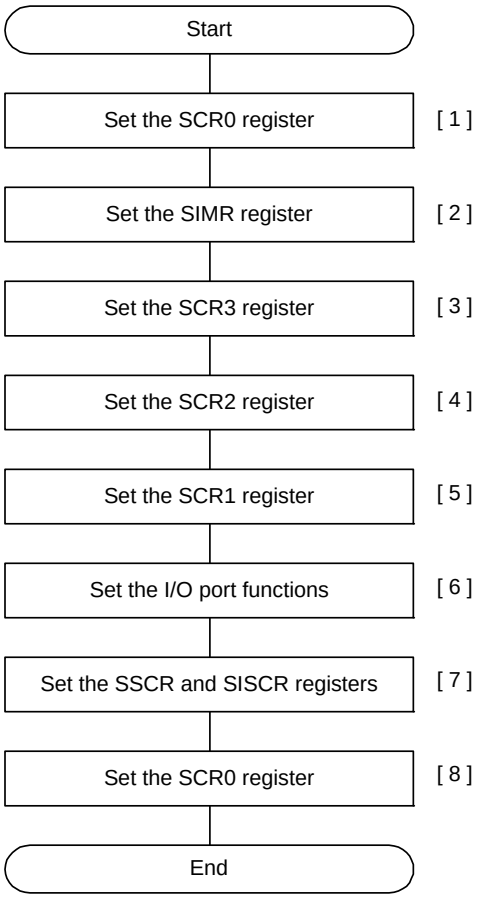


Figure 36.82 Timing of Delays in SDA Output

36.9.4 RSCI Initialization (Simple I2C Mode)

Write initial value (0000_0000h) to SCR0, then initialize RSCI according to Figure 36.83.
When changing the operating mode, transfer format, and so on, be sure to set 0 to SCR0.TE bit and SCR0.RE bit before proceeding with the changes. (Or write initial value to SCR0 again.) In simple I2C mode, the open-drain setting for the communication ports should be made on the port side.



[1] Set the following bits of SCR0 register to 0.
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set the IICSDAS[1:0] and IICSCLS[1:0] bits to 11b.
Set the IICDL[4:0] and IICINTM bits as required.
Set the IICACKT and IICCSC bits to 1.

[3] Set the transmission/reception format as the communication mode (MOD[2:0] bits = 100b) and CKE[1:0] bits = 00b.

[4] Set the bit rate modulation function*1, the clock selection, and the bit rate.

[5] Set noise filter, communication pin status, parity check, CTS/RTS function.

[6] Set I/O port settings that allow use (on NMOS opendrain output pins and Hi-Z) of the SSCLn and SSDAn pin functions.

[7] Write 1 to the following bits and clear the corresponding flag.
SSCR.RDRFC, AFERC, APERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
SISCR.IICSTIFC

[8] Set the TE and RE bits to 1.
To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time.
(for transmission and when the SIMR.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 1. If you do not use the bit rate modulation unction, you do not need to set it.

Note: Set the SCR0.TE and RE bits to 0 or 1 at the same time.

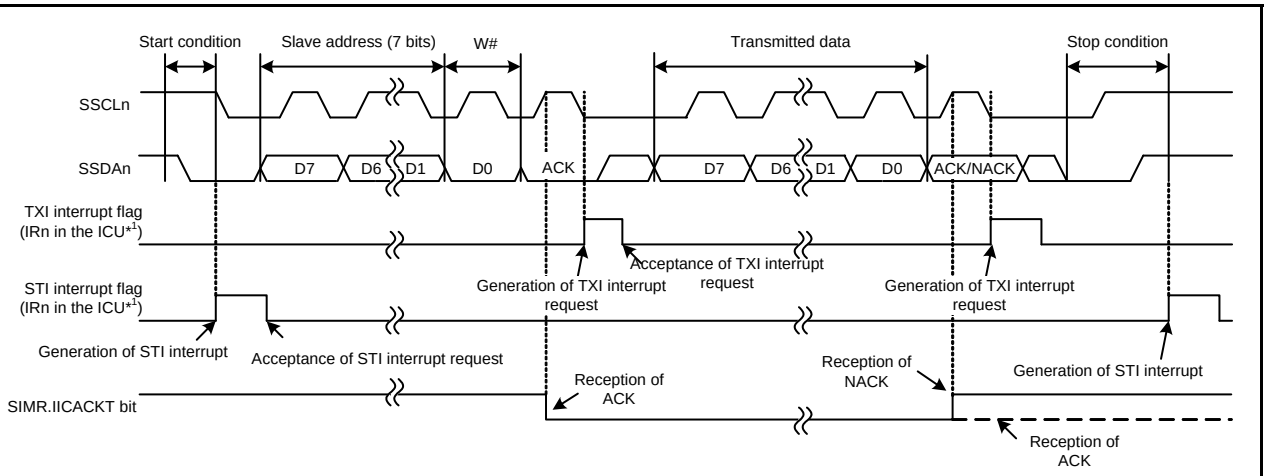
Figure 36.83 Example of the Flowchart of RSCI Initialization (for Simple I2C Mode)

36.9.5 Operation in Master Transmission (Simple I2C Mode)

Figure 36.84 and Figure 36.85 show examples of operations in master transmission, Figure 36.86 to Figure 36.88 show the example flowcharts. See Table 36.46 about the STI interrupt.

Figure 36.84 shows the operation example when SIMR.IICINTM bit is 1 (Reception/Transmission interrupt are in use). In this case, you can start DMAC or DTC by TXI interrupt. However, if use DMAC or DTC, ACK/NACK can not be confirmed. So, if you want to confirm ACK/NACK, prepare transmit data by CPU. In simple I2C mode, TXI interrupt is generated when communication of one frame is completed. And it isn't used reception interrupt in master transmission, so the SCR0.RIE bit set to 0.

Figure 36.86 shows a flow chart in the case of SIMR.IICINTM bit is 1 and address transmission by CPU and data transmission by DTC or DMAC. Figure 36.87 shows a flow chart of address and data transmission by CPU. When 10-bit slave addresses are in use, steps [3] and [4] are repeated twice.



Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

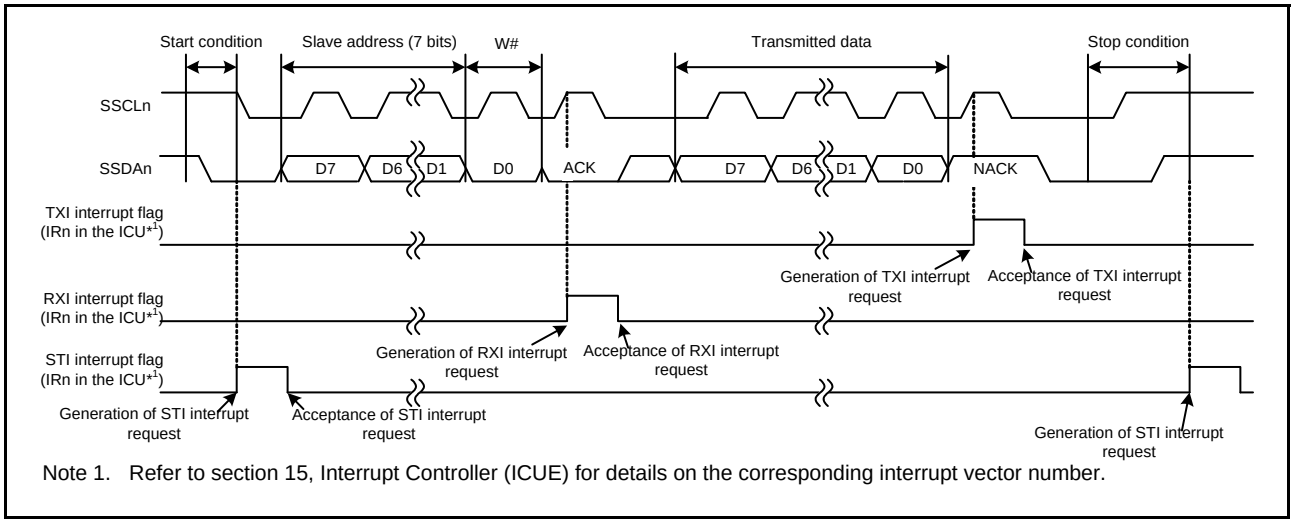
Figure 36.84 Example 1 of Operations for Master Transmission in Simple I2C Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use (SIMR.IICINTM Bit = 1))

Figure 36.85 shows an example of operations when SIMR.IICINTM bit is 0 (ACK and NACK interrupt in use). In this case, DTC or DMAC is activated by the ACK interrupt, and necessary number of data bytes are transmitted. When the NACK is received, error processing, such as transmission stop and retransmission, is performed by the NACK interrupt as the trigger.

Figure 36.88 shows a flow chart of SIMR.IICINTM bit is 0

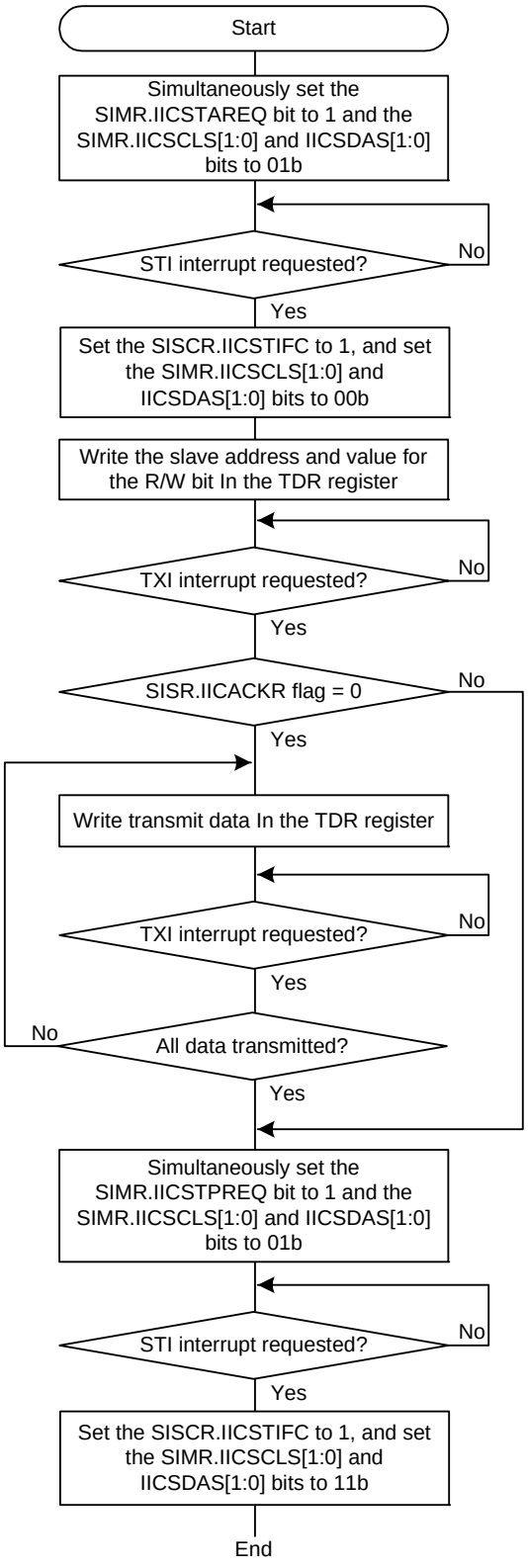
To resume communication after interrupting communication for some reason after writing transmit data to TDR, follow the procedure below.

1. Set the SCR0.TE and SCR0.RE bits to 0 to stop communication.
2. Set SIMR.IICSCLS[1:0] and SIMR.IICSDAS[1:0] bits to 11b, release the I2C bus, and clear various condition generation requests.
3. When the SSR.RDRF flag is 1, the RDR register is read by dummy and the RDRF bit is set to 0.
4. Set the SCR0.TE and SCR0.RE bits to 1 and restart communication.



Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 36.85 Example 2 of Operations for Master Transmission in Simple I²C Mode (with 7-Bit Slave Addresses, ACK Interrupts, and NACK Interrupts in Use (SIMR.IICINTM Bit = 0))



[0] [0] For transmission, set the SCR0.RIE bit to 0 at initialization. (RXI and ERI interrupts requests are disabled)

[1] [1] Generate a start condition.

[2] [2] Writing to TDR register:
Writing the slave address and value for the R/W bit to TDR register.

[3] [3] Confirming ACK response from the slave address:
Check the SISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

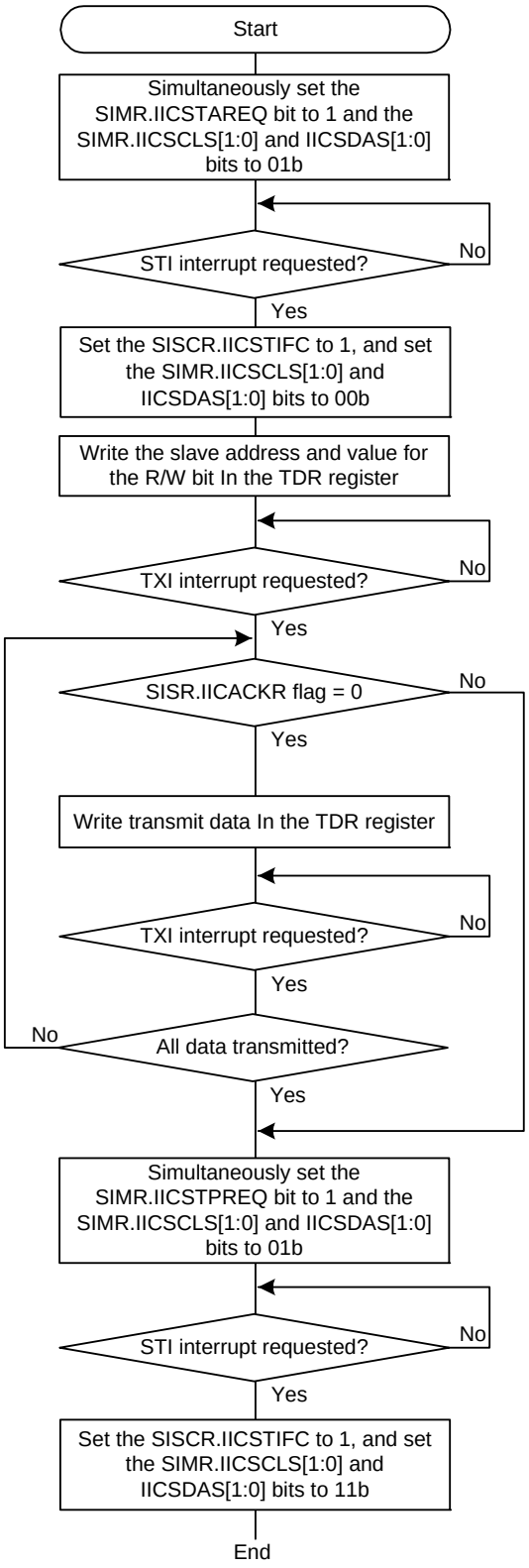
If 10-bit slave addresses are in use, processing of [2] and [3] is repeated twice.

[4] [4] Steps for continuing with serial transmission:
When transmission is to continue, write further transmit data to TDR register. Except for the first data to be transmitted, a TXI request can activate the DMAC or DTC to handle writing of data to TDR register.
But it cannot be confirmed status of ACK or NACK in this case.

[5] [5] Generation of a stop condition.

Note: In simple I^2C mode, the TXI interrupt request is generated when communication is completed, unlike the timing during clock synchronous transmission.

Figure 36.86 Example of the Procedure for Master Transmission Operations in Simple I^2C Mode (when SIMR.IICINTM Bit is 1, and when Confirming ACK/NACK by Address Transmission Only)



[0] [0] For transmission, set the SCR0.RIE bit to 0 at initialization. (RXI and ERI interrupts requests are disabled)

[1] [1] Generate a start condition.

[2] [2] Writing to TDR register:
Writing the slave address and value for the R/W bit to TDR register.

[3] Confirming ACK response from the slave address:
Check the SISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

If 10-bit slave addresses are in use, processing of [2] and [3] is repeated twice.

[4] Steps for continuing with serial transmission:
When transmission is to continue, write further transmit data to TDR. To checking whether ACK or NACK in every transmission, you cannot use DMA or DTC.

[5] Generation of a stop condition.

Note: In simple I^2C mode, the TXI interrupt request is generated when communication is completed, unlike the timing during clock synchronous transmission.

Figure 36.87 Example of the Procedure for Master Transmission Operations in Simple I^2C Mode (when SIMR.IICINTM Bit is 1, and when Confirming ACK/NACK in All Transmissions)

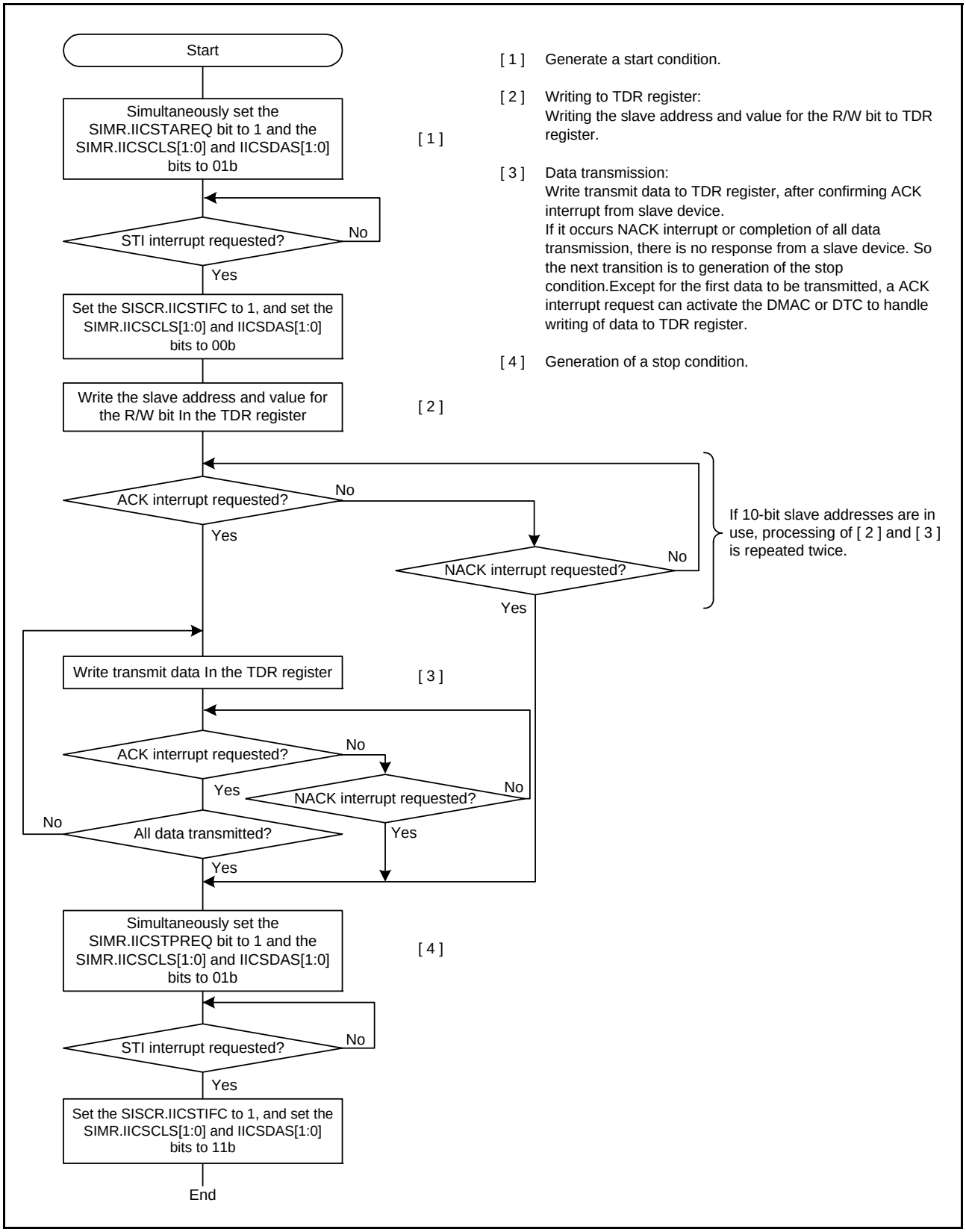
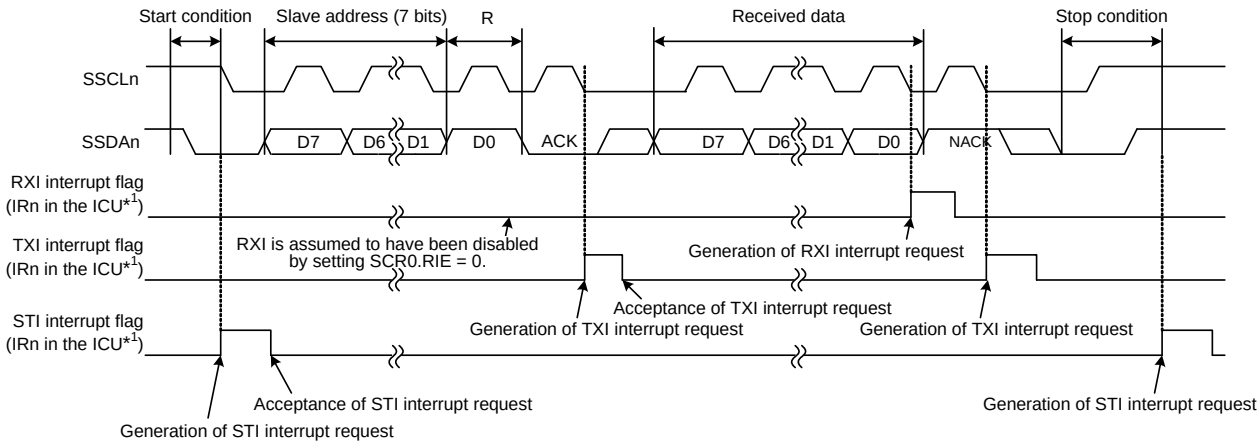


Figure 36.88 Example of the Procedure for Master Transmission Operations in Simple I²C Mode (when SIMR.IICINTM Bit is 0)

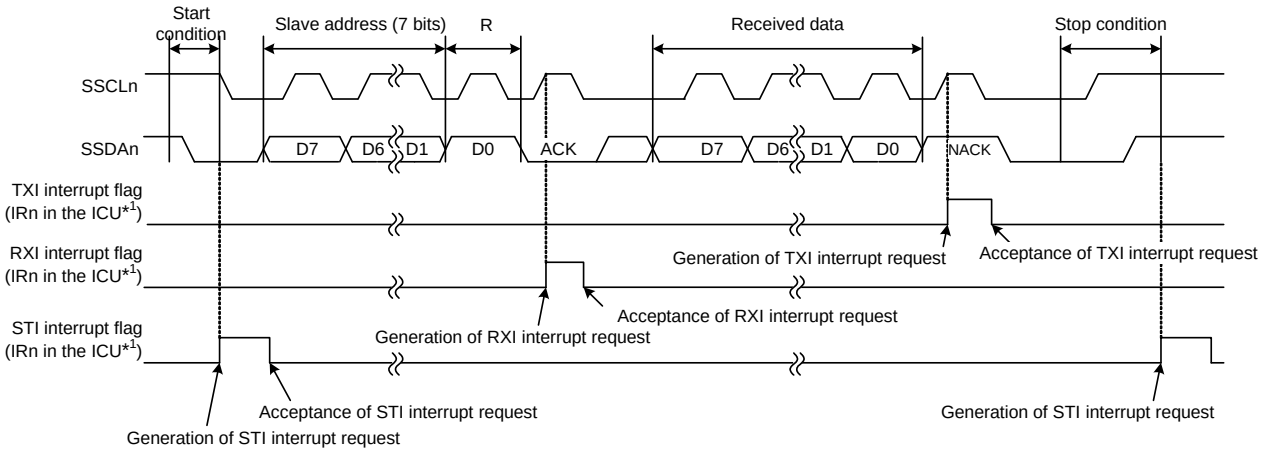
36.9.6 Master Reception (Simple I²C Mode)

Figure 36.89 and Figure 36.90 show example of operations in simple I²C mode master reception. Figure 36.91 and Figure 36.92 show flowchart of the master reception. The value of the SIMR.IICINTM bit is assumed to be 1 (use reception and transmission interrupts) and 0 (use ACK and NACK interrupts).
In simple I²C mode, the transmit end interrupt (TXI) is generated when communication of one frame is completed.



Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 36.89 Example of Operations for Master Reception in Simple I²C Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use (SIMR.IICINTM Bit = 1))



Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 36.90 Example of Operations for Master Reception in Simple I²C Mode (with 7-Bit Slave Addresses, ACK and NACK Interrupt in Use (SIMR.IICINTM Bit = 0))

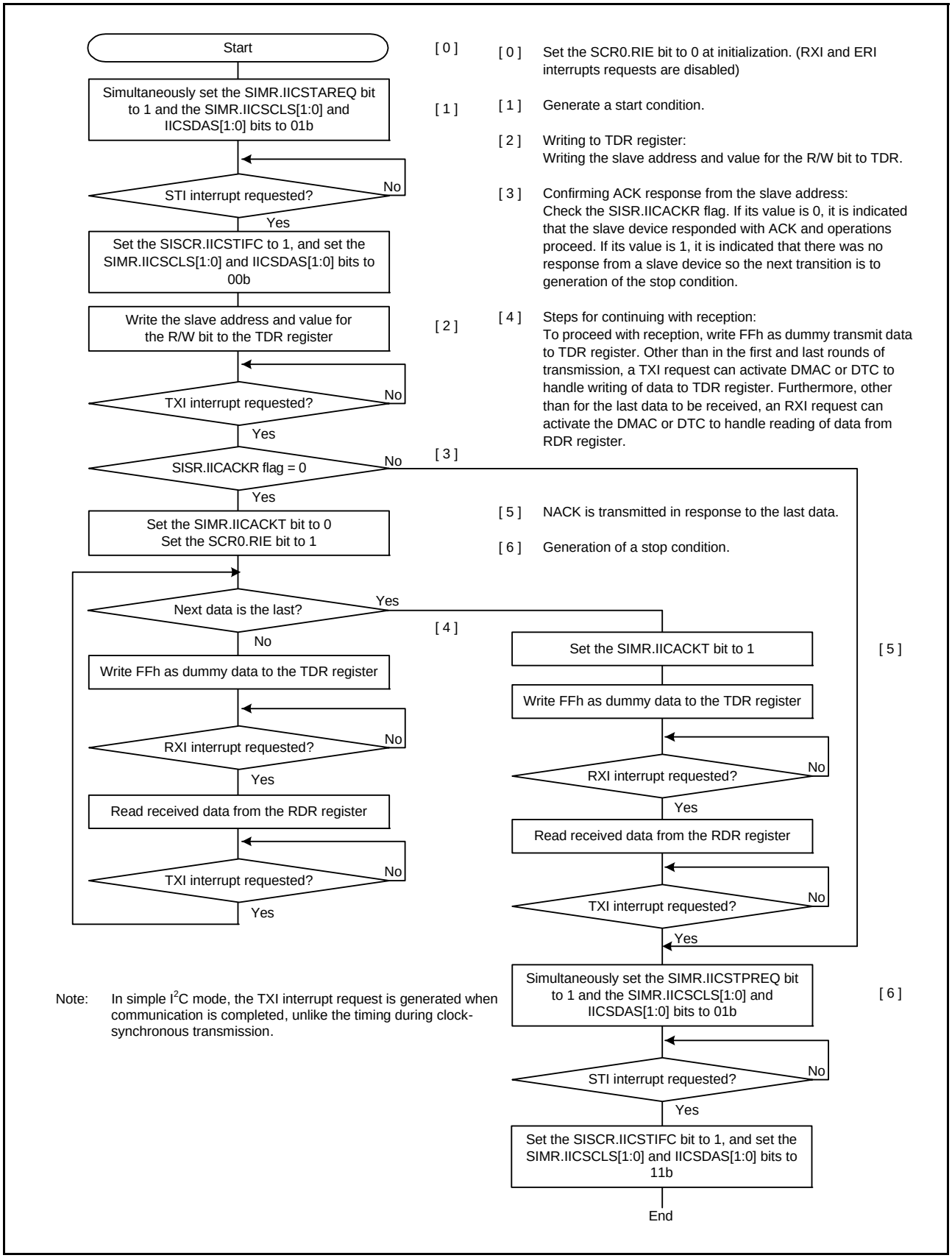


Figure 36.91 Example of the Procedure for Master Reception Operations in Simple I²C Mode (When SIMR.IICINTM Bit is 1, and Transmission Interrupts and Reception Interrupts in Use)

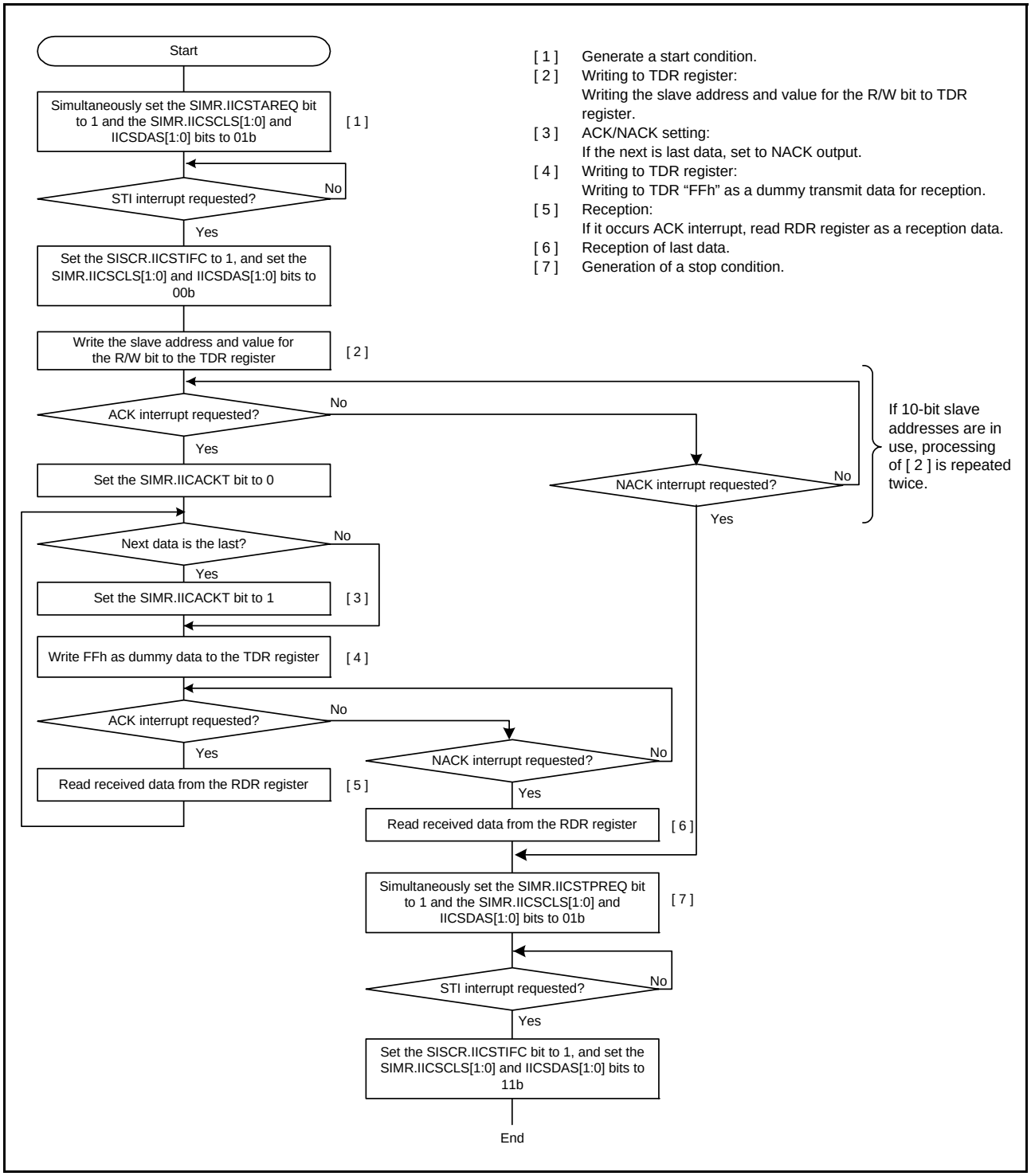


Figure 36.92 Example of the Procedure for Master Reception Operations in Simple I²C Mode (When SIMR.IICINTM Bit is 0, and ACK Interrupts and NACK Interrupts are in Use)

36.10 Operation in Clock Synchronous Mode

Figure 36.93 shows the communication data format of clock synchronous serial communication.
In clock synchronous mode, data is transmitted and received in synchronization with clock pulses. A communication data character consists of 8-bit data. In clock synchronous mode, no parity bit can be added.
In data transmission when CPHA bit = 1 and CPOL bit = 1, the RSCI outputs data from the falling edge of the sync clock until the next falling edge. In data reception, data is read at the rising edges of the sync clock. After 8-bit data is output, the communication line holds the final-bit output state. In slave communication when CPHA bit = 0, however, the communication line holds the first-bit output state.
Because the RSCI has an internal transmitter and a receiver independently, RSCI enable full-duplex communication by sharing a communication clock of the transmitter and the receiver. Furthermore, because both the transmitter and the receiver have a double- buffer structure, continuous transmission and reception are possible by writing the next transmit data during transmission and reading the previous receive data during reception.

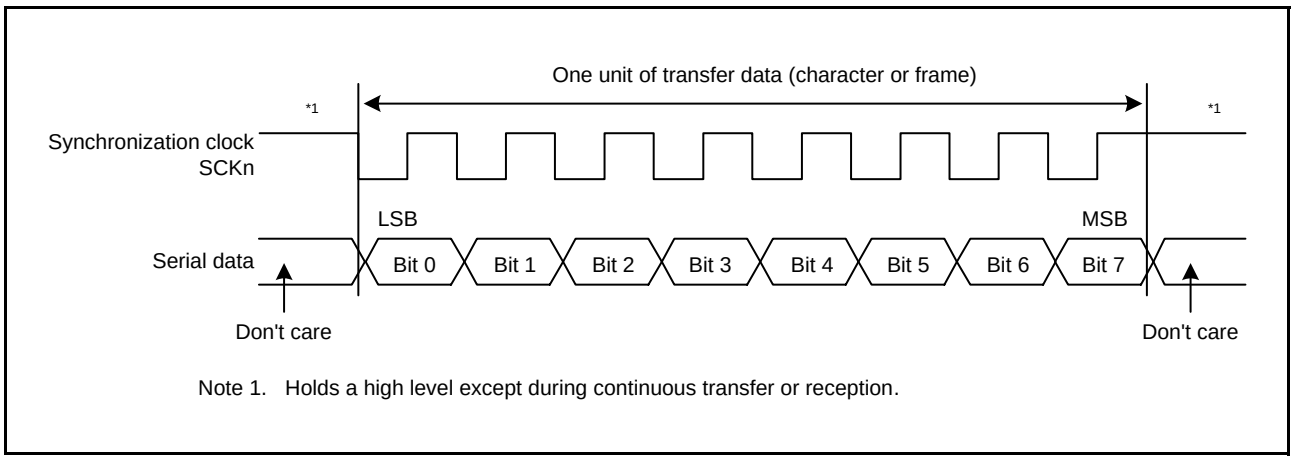


Figure 36.93 Data Format in Clock Synchronous Serial Communications (LSB First, CPHA Bit = 1, CPOL Bit = 1)

36.10.1 Clock

(1) When the Internal Clock is Selected

When the SCR3.CKE[1:0] bits are set to 00b or 01b (master mode), the internal clock generated by the on-chip baud rate generator can be selected and the sync clock is output from the SCKn pin. Eight pulses of the sync clock are output during single-character transmission/reception. The sync clock remains at a high level*1 while no transmission or reception is performed. In transmission-only or transmission/reception, the sync clock is not output unless transmit data is prepared.
When the internal clock is selected, the clock with a delay from the SCKn signal is used for the master reception sampling clock. This ensures the data setup time and hold time for high-speed communication.

Note 1. When SCR3.CPHA bit = 0 and SCR3.CPOL bit = 1 or when SCR3.CPHA bit = 1 and SCR3.CPOL bit = 1, the sync clock stops at a high level. When SCR3.CPHA bit = 0 and SCR3.CPOL bit = 0 or when SCR3.CPHA bit = 1 and SCR3.CPOL bit = 0, the sync clock stops at a low level.

(2) When the External Clock is Selected

When the SCR3.CKE[1:0] bits are set to 10b or 11b (slave mode), data is transmitted and received using the external clock that is input from the SCKn pin.

36.10.2 CTS and RTS Functions

The CTS function performs transmission/reception and controls transmission start using the CTSn# pin input when the internal clock is selected. Setting the SCR1.CTSE bit to 1 enables the CTS function. In clock synchronous communication, the CTS function can be used for the internal clock and the RTS function can be used for the external clock, so the CTS function and RTS function cannot be used at the same time.
When the CTS function is enabled, transmission/reception and transmission start only when the CTSn# pin input level is low.
When using the FIFO, if the CTSn# signal remains high before transmission, transmission will not start, but the number of data stored will be “number written to the TDR register – 1” (unlike using asynchronous FIFO). This is because data is transferred to the TSR register after writing to the TDR register, but if the CTSn# signal is set to low level, transmission starts from the TSR register, so there is no problem.
Even if the CTSn# pin input becomes high level during transmission/reception or transmission operation, frames that are being transmitted/received or being transmitted are not affected and transmission/reception or transmission operation continues.
The RTS function makes a serial communication start request using the RTSn# pin output when the external sync clock is selected. When serial communication is enabled, the RTSn# pin outputs a low level. A low level and a high level are output under the following conditions.

(a) When the SCR3.FM Bit is 0 (Non-FIFO Mode)

[Conditions for low-level output]
When the following conditions are all satisfied

- The SCR0.RE or SCR0.TE bit is 1
- No receive data are present before reading and reception is not in progress. (when SCR0.RE bit = 1)
- Data written in the TDR register is ready for transmission (when SCR0.TE bit = 1)
- The SSR.ORER flag is 0

[Condition for high-level output]
The conditions for low-level output have not been satisfied.
When SCR0.RE bit is set to 0 without reading the RDR register to terminate reception after reception is complete, the RTSn# pin output level remains high. At this time, write 0 to SCR0.RE bit.

(b) When the SCR3.FM Bit is 1 (FIFO Mode)

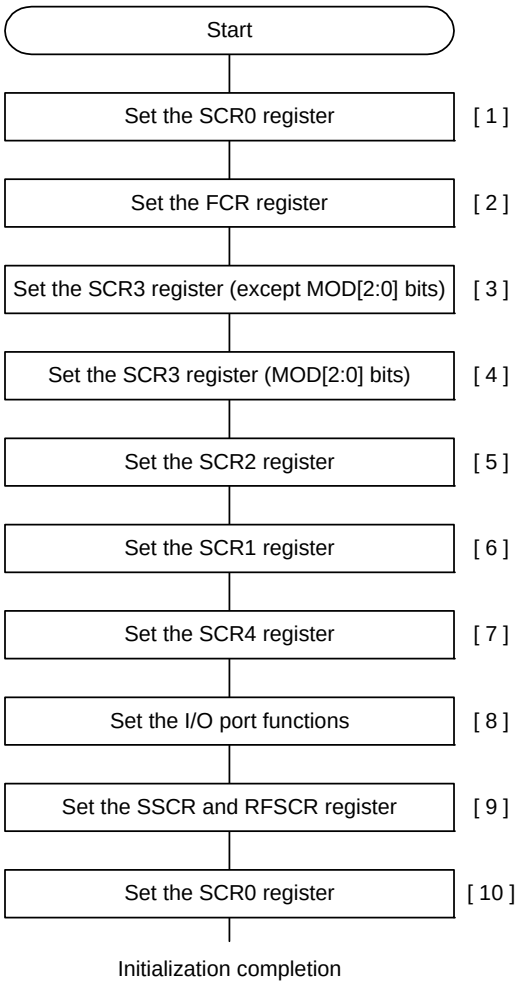
[Conditions for low-level output]
When the following conditions are all satisfied

- The SCR0.RE or SCR0.TE bit is 1
- The number of receive data stored in the receive FIFO (RDR register) is less than the value set in FCR.RSTRG[4:0] (when SCR0.RE bit = 1)
- Data written in the transmit FIFO (TDR register) is ready for transmission (when SCR0.TE bit = 1)
- The SSR.ORER flag is 0

[Condition for high-level output]
When the conditions for low-level output are not satisfied

36.10.3 RSCI Initialization (Clock Synchronous Mode)

Before starting data transmission/reception, write 0 to SCR0.TE bit and SCR0.RE bit (or write initial values to the SCR0 register) and initialize the RSCI according to the flowchart example in Figure 36.94.
Before changing operating mode or communication format, also be sure to write 0 to TE bit and RE bit.
Note that writing 0 to the RE bit does not initialize the ORER, AFER, APER, and RDRF flags in SSR register and the RDR register. Also note that writing 0 to the TE bit does not initialize the SSR.TEND flag in FIFO mode. Attention is also needed for changing operating mode.
When the SCR0.TIE bit = 1, note that setting the TE bit to 1 from 0 generates a TXI interrupt.



[1] Set the following bits of SCR0 register to 0[*1].
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set the TFRST and RFRST bits to 1 to empty FIFO.
Set the TTRG[4:0], RTRG[4:0], and RSTRG[4:0] bits.

[3] Set SCR3 except of communication mode.
FIFO use/no-use
Transmission/reception format
Clock setting
Leave unused bits (CHR[1:0], STOP, RXDESEL, MP, DEEN, GM, BLK) at their initial values.[*2]

[4] Set communication mode (MOD[2:0] bits = 010b)[*2].

[5] Select clock, set bit rate[*3]. Leave unused bits (SCR2.BCP[2:0], ABCS, ABCSE, BRME, and MDDR[7:0]) at their initial values.

[6] Set up the half-duplex communication, the loop-back function, communication pin status, and the CTS/RTS function.

[7] Set up the adjust sampling timing function. Leave unused bits (SCR4.CMPD[8:0] bits) at their initial values.

[8] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[9] Write 1 to the following bits and clear the corresponding flag.
SSCR.RDRFC, AFERC, APERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
RFSCR.DRC

[10] Set the TE or RE bit to 1.[*1, *4] To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time.
Setting the TE and RE bits allows TXDn and RXDn to be used.

Note 1. In simultaneous transmit and receive operations, the TE and RE bits in SCR0 should both be to 0 or set to 1 simultaneously.
Note 2. Set the clock (CPOL and CPHA bits) before setting the communication mode.
Note 3. If you use an external clock, you do not need to set it.
Note 4. When using the internal clock (master), the setting of reception only is prohibited.

Figure 36.94 Example of RSCI Initialization Flowchart (Clock Synchronous Mode)

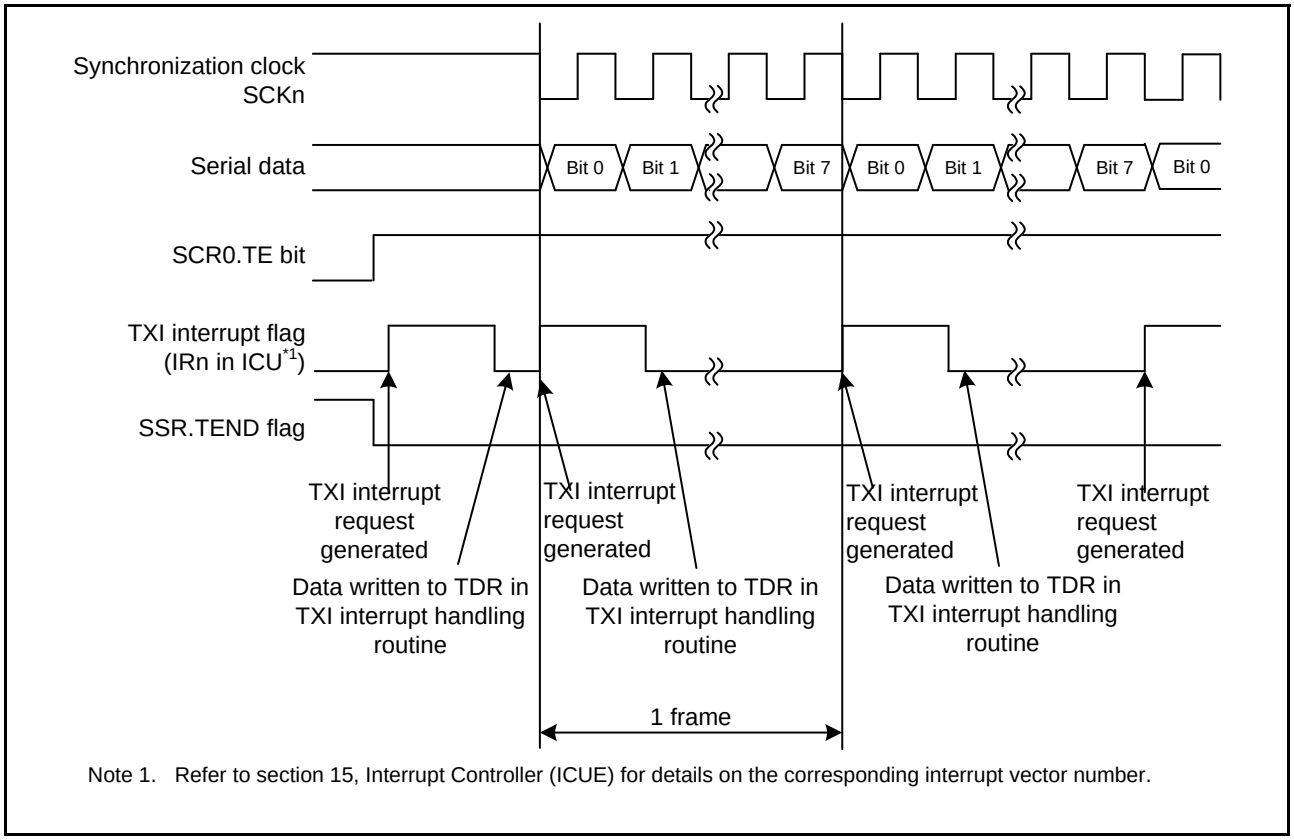
36.10.4 Serial Data Transmission (Clock Synchronous Mode)

(1) Non-FIFO Mode

Figure 36.95 to Figure 36.97 show an example of the operation for serial transmission in clock synchronous mode. In serial data transmission, the RSCI operates as described below.

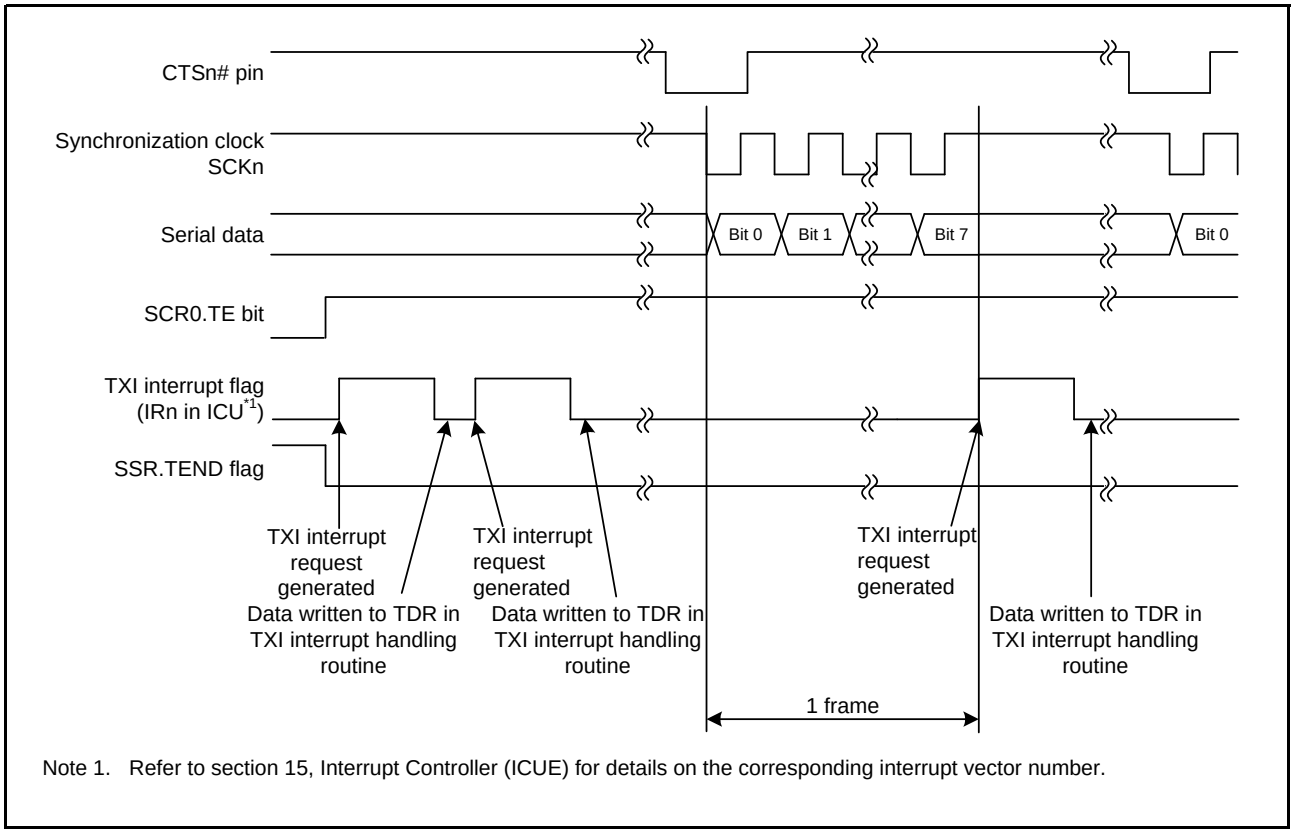
1. When data is written to the TDR register in the TXI interrupt routine, the RSCI transfers the data from the TDR register to the TSR register. When starting data transmission, set the SCR0.TIE bit and the SCR0.TE bit to 1 simultaneously by a single instruction. Then a TXI interrupt request is generated.
2. The written data is transferred from the TDR register to the TSR register, which starts transmission. When the SCR0.TIE bit is set to 1 at this time, a TXI interrupt request is generated. Writing the next transmit data to the TDR register before transmission of data transferred previously in the TXI interrupt routine is complete enables continuous transmission. When a TEI interrupt request is used, after the final transmit data is written to the TDR register in the TXI interrupt request processing routine and the final data's transmission is started, set 0 to the SCR0.TIE bit and set 1 to the TEIE bit.
3. The TXDn pin outputs 8-bit data in synchronization with the output clock (in clock output mode) or with the input clock (when external clock is selected). When the SCR1.CTSE bit = 1 (CTS function enabled), the output clock starts after the CTS signal input becomes low level.
4. Update (data write) of the TDR register is checked at the final-bit transmission timing.
5. When the TDR register has been updated, data is transferred from the TDR register to the TSR register to start sending the next frame.
6. If the TDR register has not been updated, the SSR.TEND flag is set to 1 and the final-bit output state is retained. When the SCR0.TEIE bit is set to 1 at this time, a TEI interrupt request is generated. The SCKn pin is held high.

Figure 36.98 shows a sample flowchart of serial data transmission.



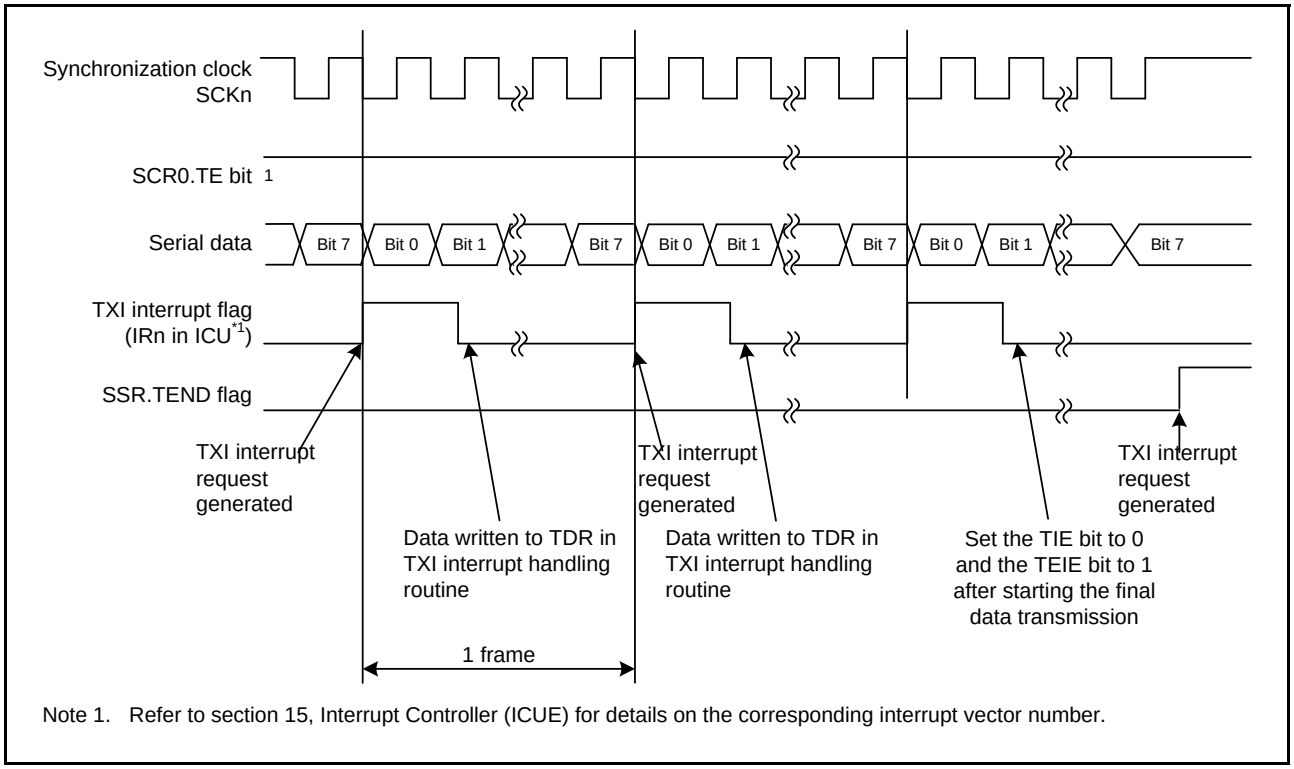
Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 36.95 Serial Transmission Example in Clock Synchronous Mode (1) (CTS Function Not Used/ Transmission Start/CPHA Bit = 1, CPOL Bit = 1)



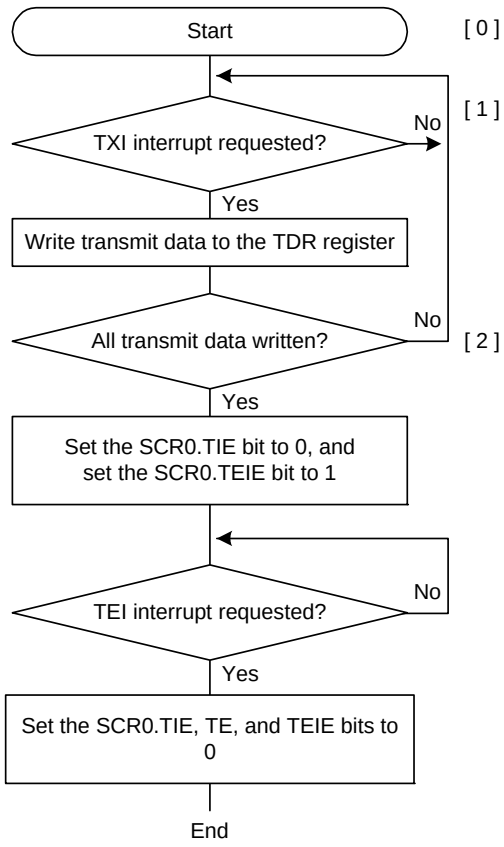
Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 36.96 Serial Transmission Example in Clock Synchronous Mode (2) (CTS Function Used/Transmission Start/CPHA Bit = 1, CPOL Bit = 1)



Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 36.97 Serial Transmission Example in Clock Synchronous Mode (3) (During Transmission to Transmission End/CPHA Bit = 1, CPOL Bit = 1)



[0] Set the SCR0.TE and TIE bits to 1 at initialization.

[1] Writing transmit data to TDR register by a TXI interrupt request:
The transmit data empty interrupt (TXI) is generated when transmission data forwards to TSR from TDR register. Write transmit data to TDR register once in the TXI interrupt handling routine.

[2] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to TDR register in the TXI interrupt handling routine. Transmit data can also be written by activating the DMAC or DTC.
If you use TEI interrupt request, set SCR0.TIE bit to 0 and TEIE bit to 1 after writing the last transmit data to TDR register.

Note: When the external clock is in use (the value of the SCR3.CKE[1:0] bits is 10b or 11b), the rising edge on the SCKn pin for the last bit sets the SSR.TEND flag to 1. Setting SCR0.TE bit to 0 immediately after this, then it may lead to insufficient received data hold time on the receiver side.

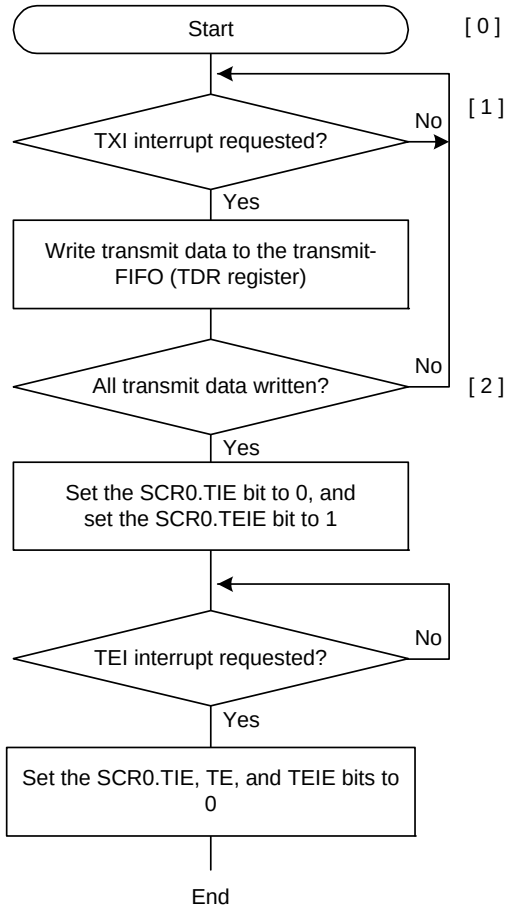
Figure 36.98 Example Flowchart of Serial Transmission in Clock Synchronous Mode (Non-FIFO Mode)

(2) FIFO Mode

Figure 36.99 shows an example of flowchart of serial transmission (a FIFO buffer selected) in clock synchronous mode with FIFO enabled.
The RSCI operates as follows when serial data transmission.

1. When data is written to the transmit FIFO (TDR register) in the TXI interrupt routine, the RSCI transfers the data from the transmit FIFO (TDR register) to the TSR register. The number of writable transmit data is [32 – number of unsent transmit data stored in the transmit FIFO (TDR register)]. If the SCR0.TIE and SCR0.TE bits are simultaneously set to 1 at the start of data transmission, a TXI interrupt request is generated.
2. Data is transferred from the transmit FIFO (TDR register) to the TSR register and transmission starts. When the number of data stored in the transmit FIFO (TDR register) is equal to or less than the threshold value of the transmit FIFO, the SSR.TDRE flag is set to 1. When the SCR0.TIE bit is set to 1 at this time, a TXI interrupt request is generated. Writing the next transmit data to the transmit FIFO (TDR register) in the TXI interrupt routine before transmission of data written to the transmit FIFO (TDR register) is complete enables continuous transmission. When a TEI interrupt request is used, after the final transmit data is written to the transmit FIFO (TDR register) in the TXI interrupt request processing routine, set 0 to the SCR0.TIE bit and set 1 to the TEIE bit.
3. The TXDn pin outputs 8-bit data in synchronization with the output clock (in clock output mode) or with the input clock (when external clock is selected). When the SCR1.CTSE bit = 1 (CTS function enabled), the output clock starts after the CTSn# pin input becomes low level.
4. The RSCI checks whether unsent transmit data is remaining in the transmit FIFO (TDR register)*1 at the final-bit transmission timing.
5. When data is remaining in the transmit FIFO (TDR register), the data is transferred from the transmit FIFO (TDR register) to the TSR register to start sending the next frame.
6. If no data is remaining in the transmit FIFO (TDR register), the SSR.TEND flag is set to 1 and the final-bit output state is retained. When the SCR0.TEIE bit is set to 1 at this time, a TEI interrupt request is generated. The SCKn pin is held high.

Note 1. The number of unsent transmit data stored in the TDR register (transmit FIFO) can be monitored by reading the TFSR.T[5:0] bits.



[0] Set the SCR0.TE and TIE bits to 1 at initialization.

[1] Writing transmit data to transmit FIFO (TDR) by a TXI interrupt request:
The transmit data empty interrupt (TXI) is generated when transmission data forwards to TSR from the transmit FIFO (TDR register) and the number of transmit-data written in the transmit FIFO is equal to or less than the transmit FIFO threshold.

[2] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to the transmit FIFO (TDR register) by using a TXI interrupt and clear the SSR.TDRE flag to 0.
The number of transmission data it's possible to write in is 32 minus (the number of stored data in transmit FIFO).
Transmit data can also be written to the transmit FIFO (TDR register) by activating the DMAC or DTC by TXI interrupt request. In this case, the TDRE flag is cleared automatically, so please don't operate the TDRE flag.
If you use TEI, set TIE bit to 0 and TEIE bit to 1 after writing the last transmit data to TDR register.

Note: When the external clock is in use (the value of the CKE[1:0] bits is 10b or 11b), the rising edge on the SCKn pin for the last bit sets the SSR.TEND flag to 1. Setting SCR0.TE bit to 0 immediately after this, then it may lead to insufficient received data hold time on the receiver side.

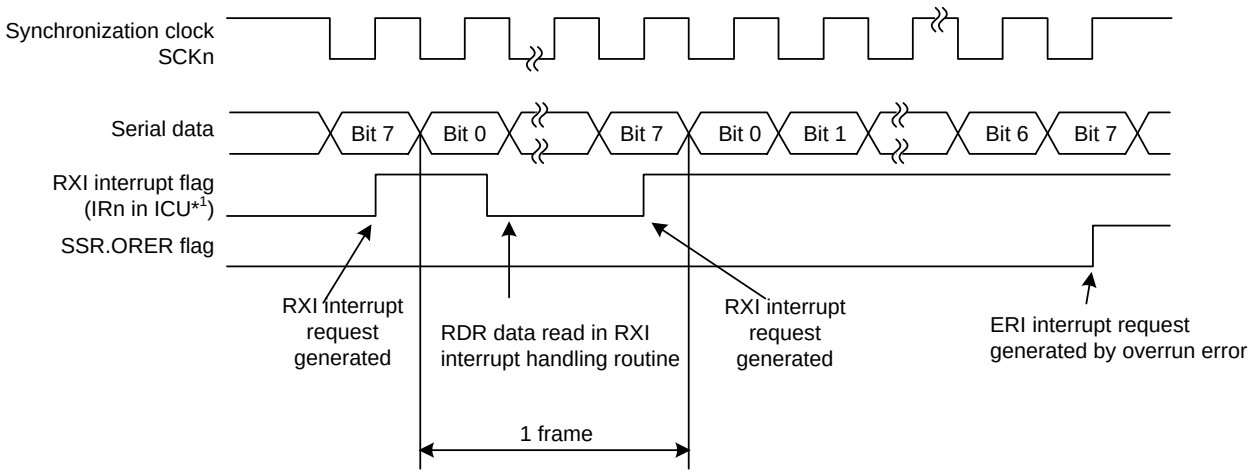
Figure 36.99 Example Flowchart of Serial Transmission in Clock Synchronous Mode (FIFO Mode)

36.10.5 Serial Data Reception (Clock Synchronous Mode)

(1) Non-FIFO Mode

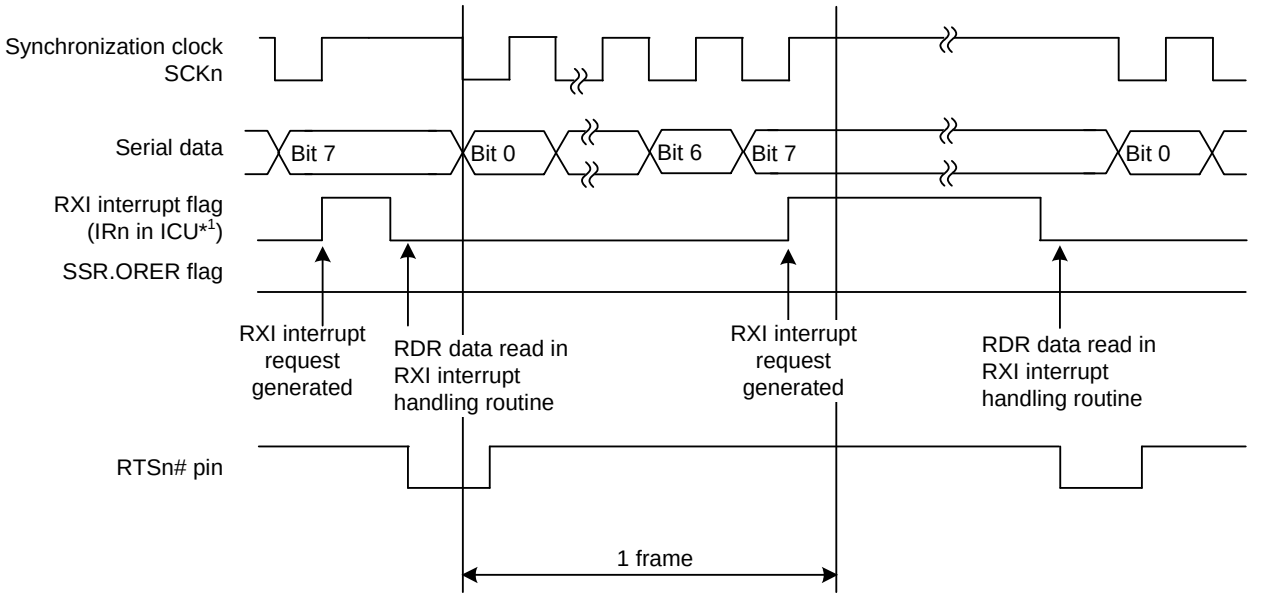
Figure 36.100 and Figure 36.101 show operation examples of serial data reception in clock synchronous mode.
The RSCI operates as follows during serial data reception. Reception-only operation is possible only in slave mode. (In master mode, reception-only operation is prohibited.)

1. When the SCR0.RE bit is set to 1, the RTSn# pin output becomes low (when the RTS function is used).
2. The RSCI starts data reception in synchronization with input or output of the sync clock, and transfers receive data to the RSR register.
3. When an overrun error occurs, the SSR.ORER flag is set to 1. When the SCR0.RIE bit = 1 at this time, an ERI interrupt request is generated and the received data is not transferred to the RDR register.
4. When data is normally received, the received data is transferred to the RDR register. When the RIE bit = 1 at this time, an RXI interrupt request is generated. Reading the received data transferred to the RDR register in the RXI interrupt handling routine before the next data is completely received enables continuous reception. When the received data transferred to the RDR register is read, the RTSn# pin output becomes low (when the RTS function is used).
   If you want to prevent the RTSn# pin output from turning low after the final data is received, clear the SCR0.RE bit to 0 and then read the RDR register.



Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 36.100 Example of Operation for Serial Reception in Clock Synchronous Mode (1)
(When RTS Function is Not Used/CPHA Bit = 1, CPOL Bit = 1)

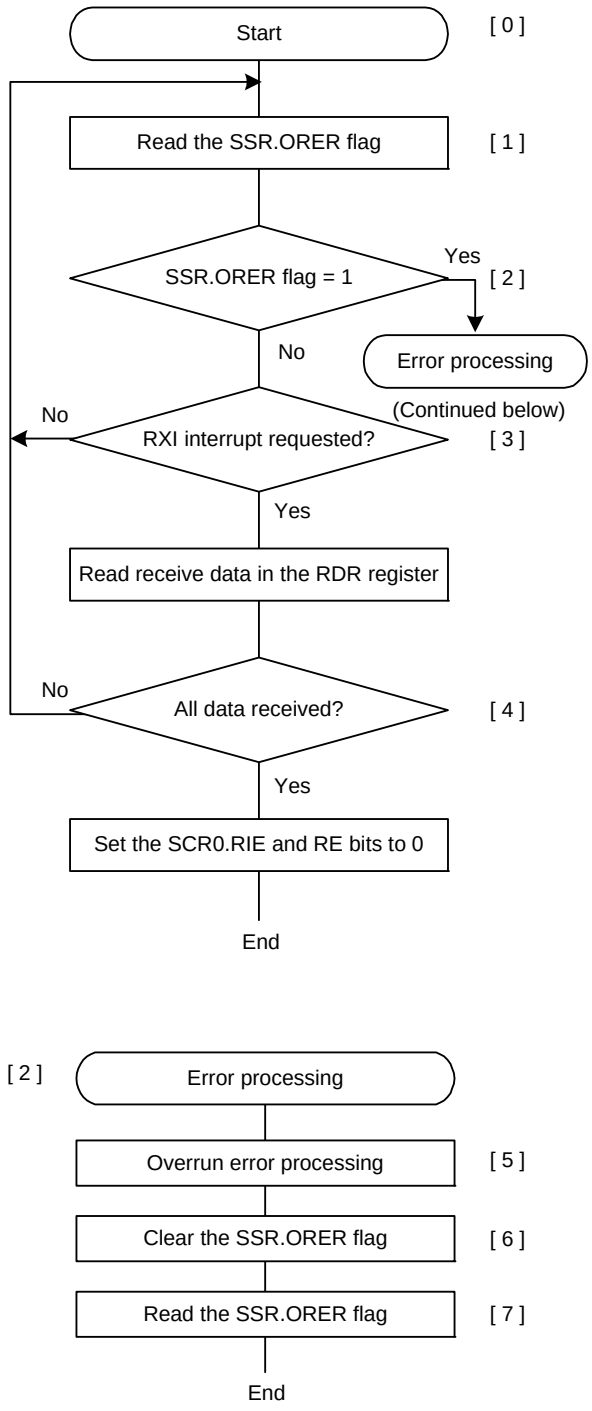


Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 36.101 Example of Operation for Serial Reception in Clock Synchronous Mode (2) (When RTS Function is Used/CPHA Bit = 1, CPOL Bit = 1)

While the reception error flag is set to 1, subsequent reception are disabled. Therefore, before continuing reception, be sure to clear the ORER, AFER, and APER flag in SSR to 0. Also be sure to read the RDR register in the overrun error processing. If the SCR0.RE bit is set to 0 during reception to forcibly terminate the reception operation, unread receive data may be remaining in the RDR register. In this case, read the RDR register.

Figure 36.102 shows a sample flowchart for serial data reception.



[0] Set the SCR0.RE and RIE bits to 1 at initialization.

[1][2]Receive error processing:
If a receive error occurs, after performing the appropriate error processing, be sure to ORER flag is 0. Reception cannot be resumed if ORER flag is 1.

[3] Read the receive data in the RDR register once in the receive data full interrupt (RXI) request handling routine.

[4] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, read data from RDR register in the RXI interrupt handling routine. The RDR data can also be read by activating the DMAC or DTC by the receive data full interrupt (RXI) request.

[5] Overrun error processing:
Read the RDR register.

[6] Clearing the error flag:
Set the flag clear bit corresponding to the reception error to 1. Done this step and step [5], the next frame can be received normally.

[7] Confirming that the error flag is cleared:
Confirm all error flags are cleared.

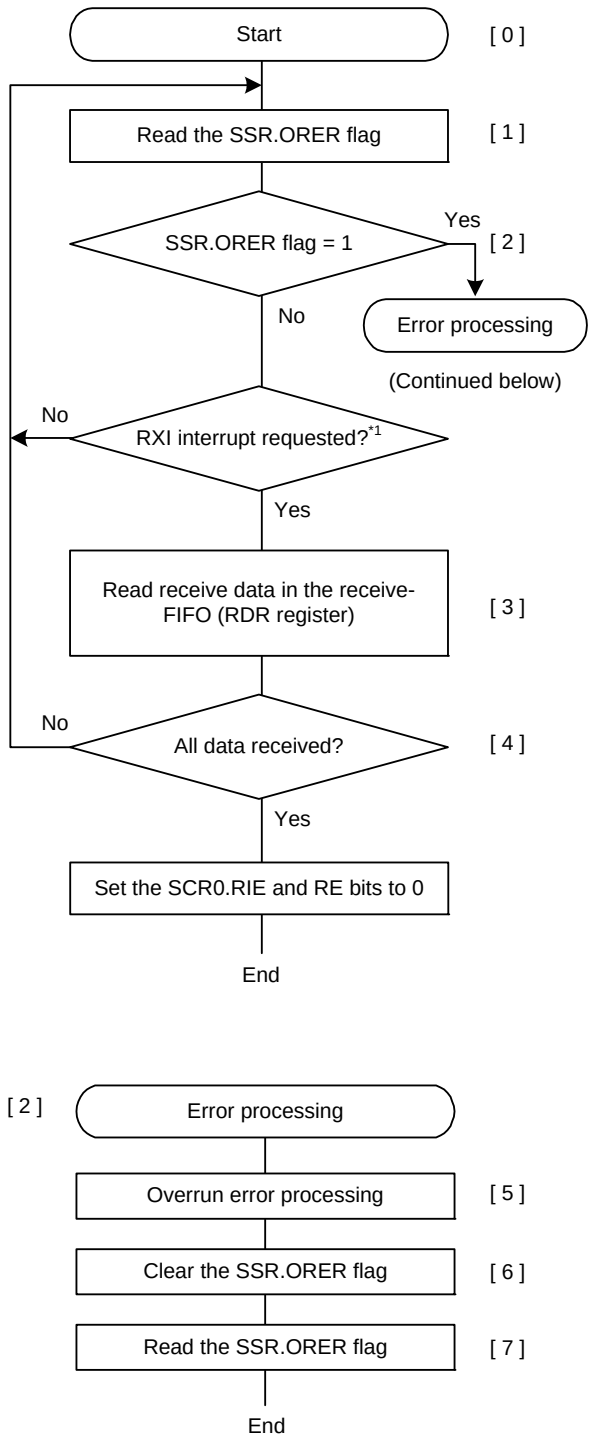
Figure 36.102 Example Flowchart of Serial Reception in Clock Synchronous Mode (Non-FIFO Mode)

(2) FIFO Mode

Figure 36.103 shows an example of serial data reception flowchart in clock synchronous mode with FIFO enabled. The RSCI operates as follows during serial data reception. Reception-only operation is possible only in slave mode. (In master mode, reception-only operation is prohibited.)

1. When the SCR0.RE bit is set to 1, the RTSn# pin output turns low (when the RTS function is used).
2. The RSCI starts receiving data in synchronization with input or output of the sync clock, and transfers the received data to the receive FIFO (RDR register).
3. When an overrun error occurs, the SSR.ORER flag is set to 1. When the SCR0.RIE bit = 1 at this time, an ERI interrupt request is generated and the received data is not transferred to the receive FIFO (RDR register)*1.
4. When data is normally received, the received data is transferred to the receive FIFO (RDR register)*1. When the number of receive data stored in the receive FIFO (RDR register) is equal to or more than the threshold value of the receive FIFO, the SSR.RDRF flag is set to 1. When the RIE bit = 1 at this time, an RXI interrupt request is generated. Reading the received data transferred to the receive FIFO (RDR register) in the RXI interrupt handling routine before an overrun error occurs enables continuous reception. When the received data transferred to the receive FIFO (RDR register) is read and the number of data becomes lower than the RTS# output threshold value, the RTSn# pin output becomes low (when the RTS function is used).

Note 1. The RDR.RDAT[8] bit is not used.



[0] Set the SCR0.RE and RIE bits to 1 at initialization.

[1][2]Receive error processing:
If a receive error occurs, after performing the appropriate error processing, be sure to ORER flag is 0. Reception cannot be resumed if ORER flag is 1.

[3] Read the receive data in the receive FIFO (RDR register) in the RXI interrupt handling routine. The reception data stored in the receive FIFO (RDR register) is read until the number of stored data will be below the FCR.RTRG value at least. When the RDR register is read, the SSR.RDRF flag is cleared. You can confirm the number of the reception data in the FIFO by RFSR.R[5:0] bits.

[4] Serial reception continuation procedure:
To continue serial reception, before the overrun error occurs, read data from RDR register. The RDR data can also be read by activating the DMAC or DTC by the receive data full interrupt (RXI) request.

Note 1. Set all received data to an integer multiple of the receive FIFO threshold. If it is not an integral multiple, an RXI interrupt is not generated and it will become an infinite loop, so set a timeout time separately using a timer etc.

[5] Overrun error processing:
Read the RDR register.

[6] Clearing the error flag:
Set the flag clear bit corresponding to the reception error to 1. Done this step and step [5], the next frame can be received normally.

[7] Confirming that the error flag is cleared:
Confirm all error flags are cleared.

Figure 36.103 Example Flowchart of Serial Reception in Clock Synchronous Mode (FIFO Mode)

36.10.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode)

In clock synchronous mode, transmission and reception are simultaneously performed, so the number of transmitted data and the number of received data are the same.

(1) Non-FIFO Mode

Figure 36.104 shows an example of serial data concurrent transmission/reception flowchart in clock synchronous mode.
After the RSCI is initialized, perform the following procedure for serial data concurrent transmission/reception.
When switching mode from transmission to concurrent transmission/reception, check that the SSR.TEND flag is set to 1 to ensure that the RSCI is in the transmission complete state. Then set SCR0.TE bit = 0 and RE bit = 0 and then set the TE, RE, TIE, and RIE bits in SCR0 register to 1 simultaneously by a single instruction.
When switching mode from reception to concurrent transmission/reception, check that the RSCI is in the reception complete state, and then set SCR0.TE bit = 0 and RE bit = 0. After that, check that the error flags (ORER, AFER, and APER flags) in SSR are cleared to 0, and then set the TE, RE, TIE, and RIE bits in SCR0 register to 1 simultaneously by a single instruction.
When the RTS function is used in the concurrent transmission/reception operation, if you want to prevent the RTSn# pin output from turning to low after the final data is received as in the reception operation, clear the RE and TE bits in SCR0 register to 0 simultaneously, and then read the RDR register.

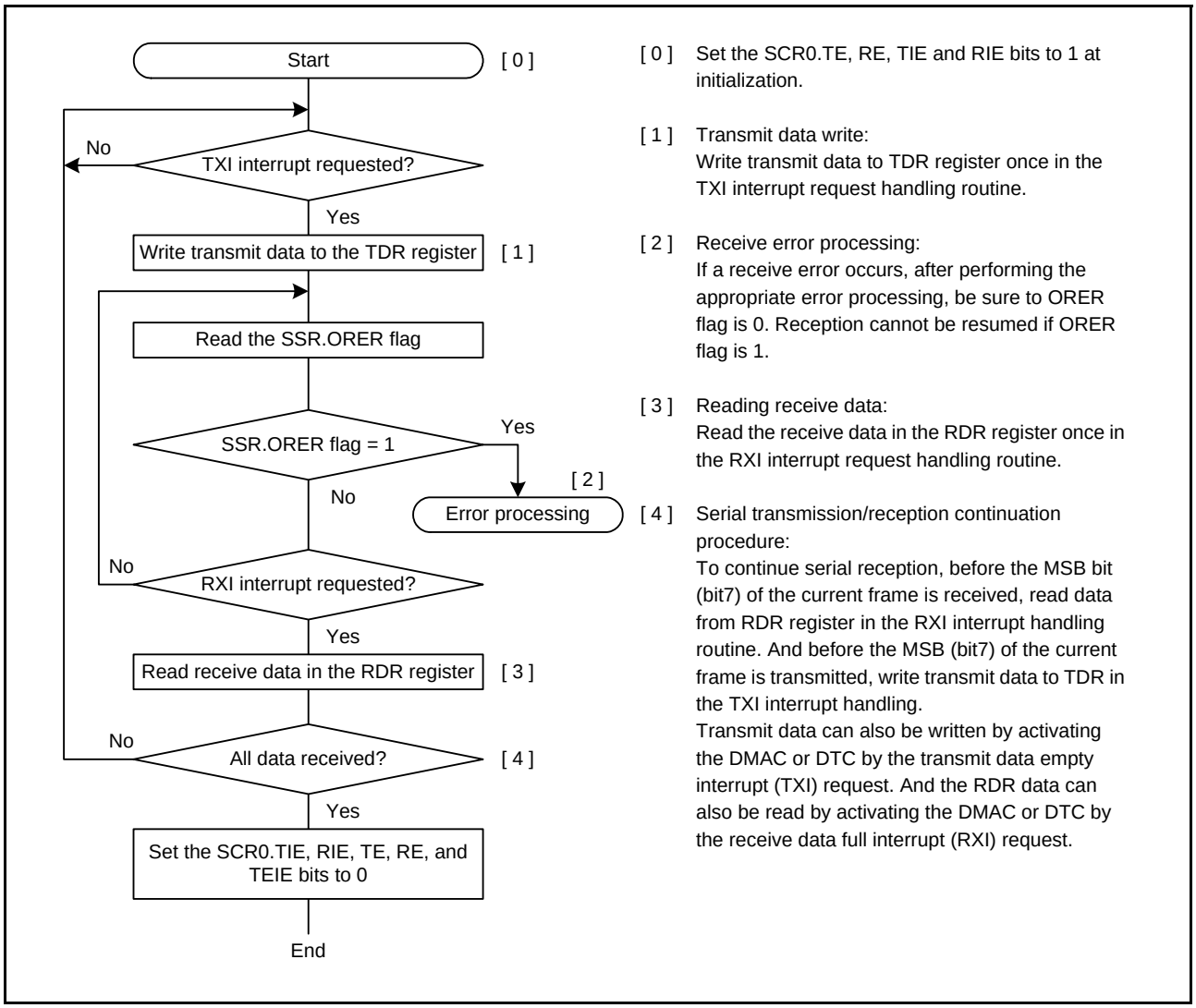


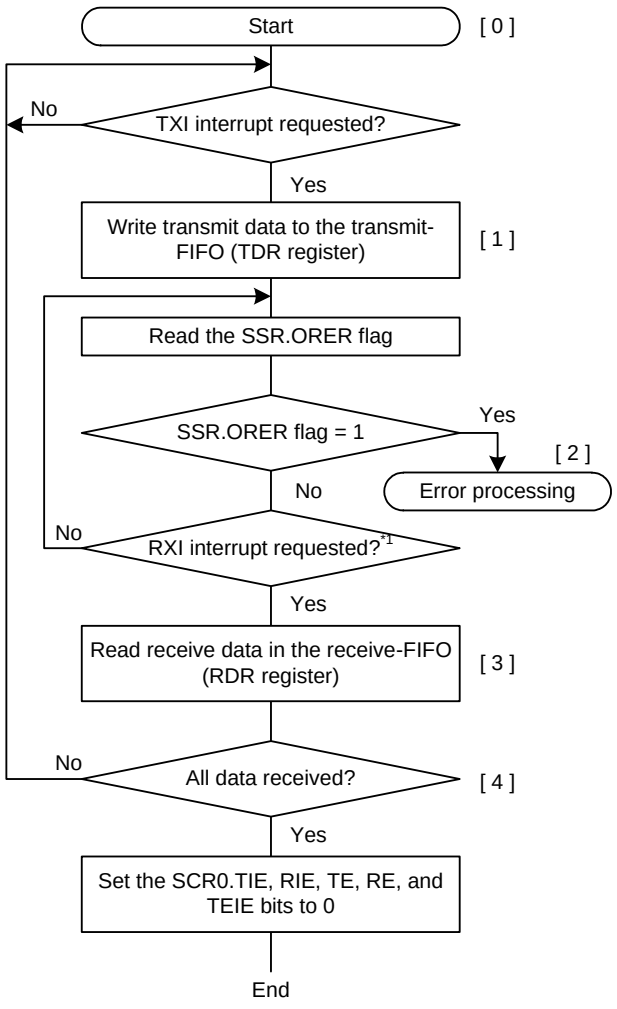
Figure 36.104 Example Flowchart of Simultaneous Serial Transmission and Reception in Clock Synchronous Mode (Non-FIFO Mode)

(2) FIFO Mode

Figure 36.105 shows an example of serial data concurrent transmission/reception flowchart in clock synchronous mode with FIFO enabled.

After the RSCI is initialized, perform the following procedure for serial data concurrent transmission/reception.When switching mode from transmission to concurrent transmission/reception, check that the SSR.TEND flag is set to 1 to ensure that the RSCI is in the transmission complete state. Then set SCR0.TE bit = 0 and RE bit = 0 and then set the TE, RE, TIE, and RIE bits in SCR0 register to 1 simultaneously by a single instruction.

When switching mode from reception to concurrent transmission/reception, check that the RSCI is in the reception complete state, and then set SCR0.TE bit = 0 and RE bit = 0. After that, check that the error flags (ORER, AFER, and APER flags) in SSR are cleared to 0, and then set the TE, RE, TIE, and RIE bits in SCR0 to 1 simultaneously by a single instruction.



Note 1. Set the number of all received data to an integer multiple of the receive FIFO threshold. If it is not an integral multiple, an RXI interrupt is not generated and it will become an infinite loop, so set a timeout time separately using a timer etc.

[0] Set the SCR0.TE, RE, TIE and RIE bits to 1 at initialization.

[1] Transmit data write:
Write transmit data to transmit FIFO (TDR register) in the TXI Interrupt request handling routine. The number of data to be written should match the number specified in the receive FIFO threshold.

[2] Receive error processing:
If a receive error occurs, read the SSR.ORER flag, perform the appropriate error processing such as stopping communication and reading all received data to clear the SSR.RDRF flag, and then set the ORER flag to 0. Reception cannot be resumed if ORER flag is 1.

[3] Reading receive data:
Read the same number of received data stored in the receive FIFO (RDR register) as the number specified in the receive FIFO threshold.

[4] Serial transmission/reception continuation procedure:
To continue serial reception, before the overrun error occurs, read data from the receive FIFO (RDR register) and clear SSR.RDRF flag. To continue serial transmission, write transmit data to the transmit FIFO (TDR) by using a TXI interrupt and clear TDRE flag to 0. Transmit data can also be written to the transmit FIFO (TDR register) by activating the DMAC or DTC by the transmit data empty interrupt (TXI) request. And The RDR data can also be read by activating the DMAC or DTC by the receive data full interrupt (RXI) request. In this case, the TDRE and RDRF flags are cleared automatically, so please don't operate the TDRE and RDRF flags.

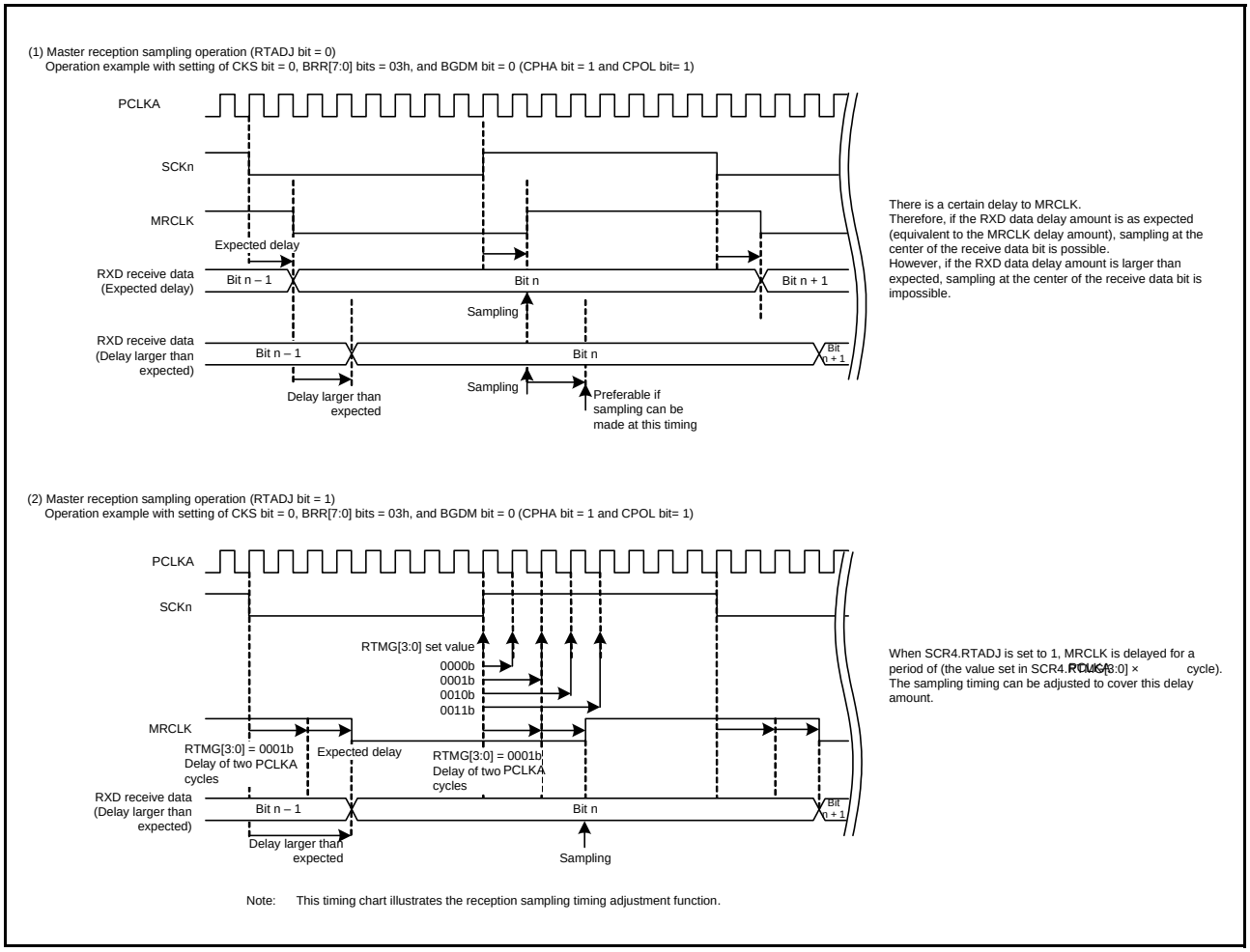
Figure 36.105 Example Flowchart of Simultaneous Serial Transmission and Reception in Clock Synchronous Mode (FIFO Mode)

36.10.7 Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with Internal Clock Used

When the clock synchronous mode with internal clock is used (master mode), MRCLK is used as a reception sampling clock.

This function adjusts the reception sampling timing by delaying MRCLK by 1 to 4 PCLKA cycles and adding a digital delay. MRCLK's analog delay cannot be adjusted by this function.

Setting the SCR4.RTADJ bit to 1 enables this function. The delay value is set in SCR4.RTMG[3:0] bits.



Note: This timing chart illustrates the reception sampling timing adjustment function.

Figure 36.106 Reception Sampling Timing Adjustment Operation in Clock Synchronous Mode (Master) and Simple SPI Mode (Master)

36.11 Operation in Simple SPI Mode

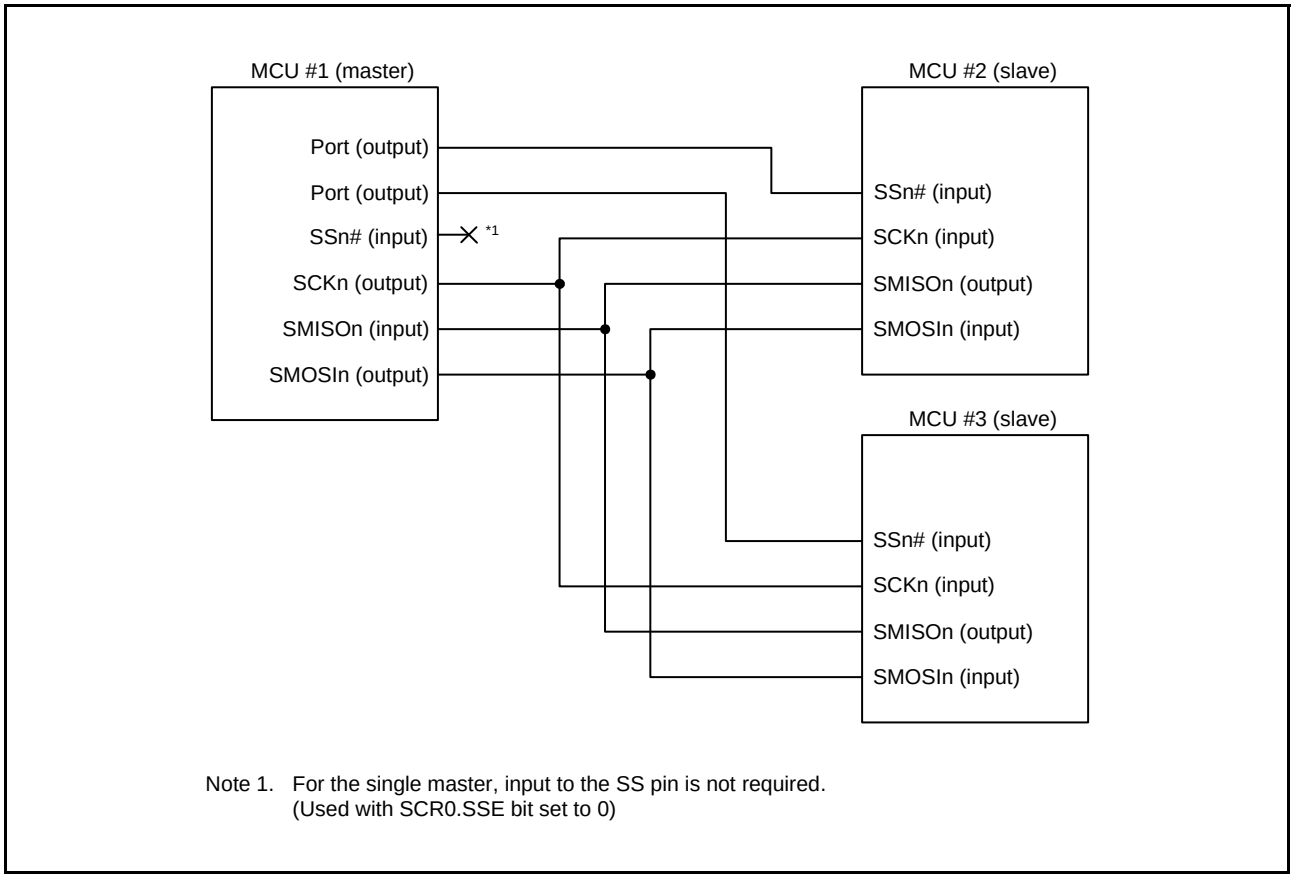
As an extended function, the RSCI supports a simple SPI mode that handles transfer among one or multiple master devices and multiple slave devices.

Making the settings for simple SPI mode (SCR3.MOD[2:0] bits = 011b) plus setting the SSE bit in the SCR0 register to 1 places the RSCI 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 SCR0 to 0 in such cases.

Figure 36.107 shows an example of connections for simple SPI mode.

In simple SPI mode, data are transferred in synchronization with clock pulses in the same way as in clock synchronous mode. One character of data for transfer consists of 8 bits of data, and parity bits cannot be appended to this.

Since the receiver and transmitter are independent of each other within the RSCI module, full-duplex communications are possible, with a common clock signal. Furthermore, since both the transmitter and receiver have a double-buffered structure, writing of further transmit data while transmission is in progress and reading of previously received data while reception is in progress are both possible. Continuous transfer is thus possible.



Note 1. For the single master, input to the SS pin is not required.
(Used with SCR0.SSE bit set to 0)

Figure 36.107 Example of Connections via a Simple SPI Mode

36.11.1 States of Pins in Master and Slave Modes

In simple SPI mode, input and output directions of each pin vary depending on master mode (SCR3.CKE[1:0] bits = 00b or 01b) and slave mode (SCR3.CKE[1:0] bits = 10b or 11b).
Table 36.40 lists the states of pins according to the mode and the level on the SSn# pin.

Table 36.40 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 cannot proceed)	Input for received data (but disabled)	High-impedance	Clock input (but disabled)
	Low level (transfer can proceed)	Input for received data	Output for data transmission*2	Clock input

Note 1. When there is only a single master (SCR0.SSE bit = 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 SMISOn pin is in the high-impedance state when serial transmission is disabled (SCR0.TE bit = 0).
Note 3. The SCKn pin is in the high-impedance state when serial transmission is disabled (SCR0.TE bit = 0 and RE bit = 0) in a multi-master configuration (SCR0.SSE bit = 1).

36.11.2 SS Function in Master Mode

Setting the SCR3.CKE[1:0] bits to 00b or 01b enables master mode.
In single master mode (SSE bit = 0), the SSn# pin is not used and data transmission and reception are enabled regardless of the SSn# pin input level. The SSn# pin is available for other purposes.
When in multi-master mode (SSE bit = 1) and the SSn # pin input is high, the master outputs a clock from the SCKn pin and performs transmission and reception operations. And outputting clock indicates “There are no other masters” or “Another master is not performing reception or transmission”. When the SSn# pin input level is low in multi-master mode (SSE bit = 1), this means that another master exists and it is performing data transmission/reception. At this time, the RSCI makes the TXDn pin output and the SCKn pin output high impedance and does not start data transmission/reception. In addition, the SSR.MFF flag is set to 1 as a mode fault error. In multi-master mode, read this flag bit to perform the error processing. If a mode fault error occurs during the transmission/reception operation, the SCKn pin and the TXDn pin output are made high impedance while the SSn# pin input level is low. In this case, any of TXI, RXI, and TEI interrupts occurs.
Control the SS signal output in master mode with a general-purpose port.

36.11.3 SS Function in Slave Mode

Setting the SCR3.CKE[1:0] bits to 10b or 11b enables slave mode.
When the SSn# pin input level is high, the RXDn pin output becomes high impedance and the clock input from the SCKn pin is ignored. When the SSn# pin input level is low, the clock input from the SCKn pin becomes effective, enabling data transmission and reception.
When the SSn# pin input changes from low to high level during the transmission/reception operation, the RXDn pin output is made high impedance and the transmission/reception operation is immediately suspended. If the transmission is in progress, the SSR.TEND flag will not be set, a transmit end interrupt will not be output, and an abnormal stop status will occur. So, do not negate the SSn # pin during slave transmission/reception. If an abnormal stop occurs, set SCR0.RE bit and SCR0.TE bit to 0 to stop transmission/reception. To resume transmission/reception, set SCR0.RE bit and SCR0.TE bit to 1 after at least 3 PCLKA cycles.

36.11.4 Relationship between Clock and Transmit/Receive Data

The clock to be used for data transmission/reception is selectable from four types using the SCR3.CPOL and CPHA bits. Figure 36.108 shows the relationship between clock and transmit data/receive data. The same relationship applies to master mode and slave mode.

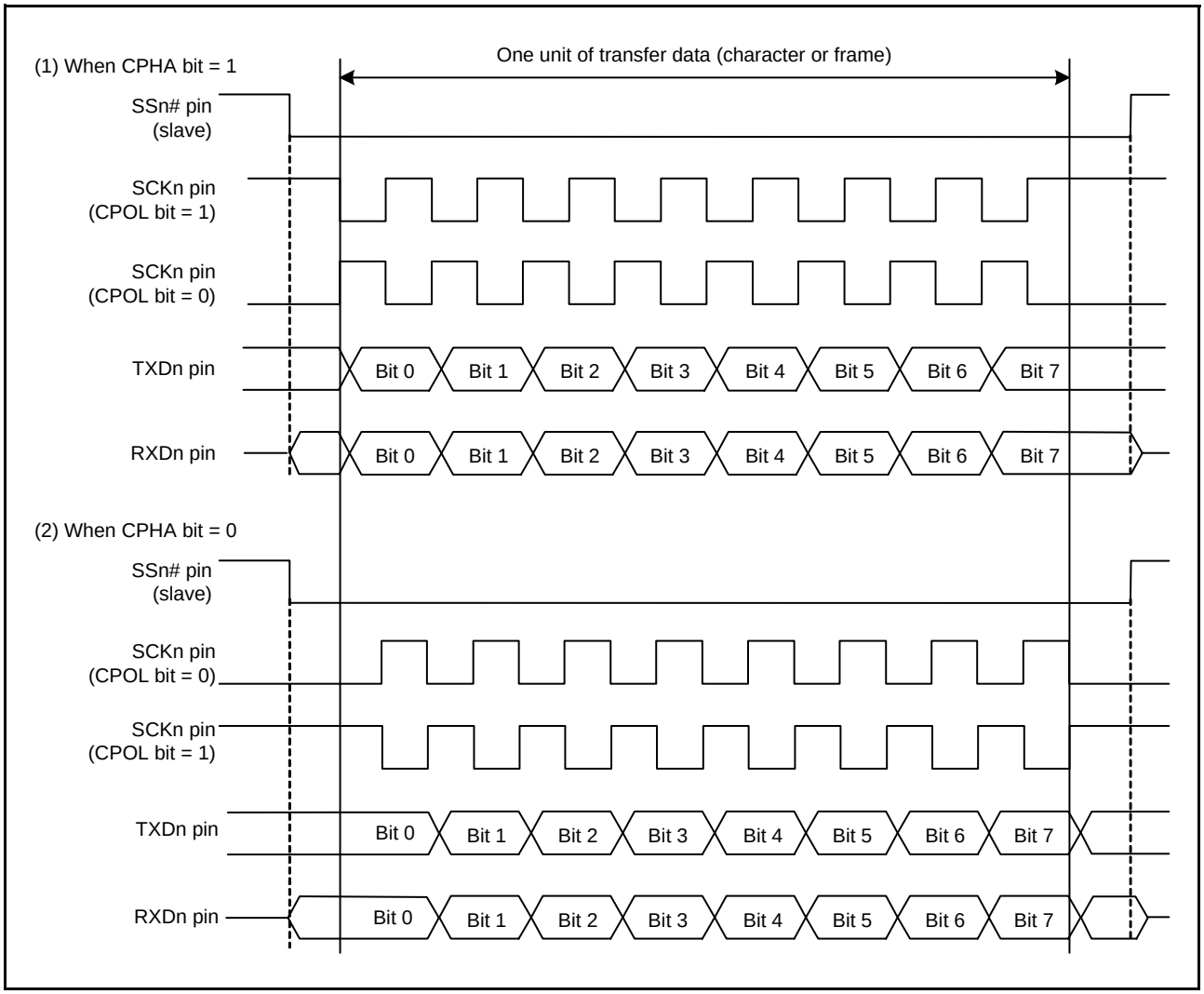


Figure 36.108 Relation between Clock Signal and Transmit/Receive Data in Simple SPI Mode

36.11.5 RSCI Initialization (Simple SPI Mode)

The RSCI can be initialized using the same initialization procedure as for clock synchronous mode (Figure 36.94, Example of RSCI Initialization Flowchart (Clock Synchronous Mode)). The master devices and slave devices use the same clock type selected by the SCR3.CPOL and CPHA bits.
Before performing initialization or changing operating mode or communication format, be sure to stop communication (SCR0.RE bit = 0 and SCR0.TE bit = 0).
Note that setting the RE bit to 0 does not initialize the ORER, AFER, and APER flags in SSR register and the RDR register.
Note that, when SCR0.TIE bit = 1, setting the TE bit to 1 from 0 generates a TXI interrupt.

36.11.6 Transmission and Reception of Serial Data (Simple SPI Mode)

In master mode, set the SSn# pin of the destination slave device to low level before starting data transmission/reception and set to high level after the end of data transmission/reception. In multiple master operation with SCR0.SSE bit = 1 even in master mode, a mode fault error will occur if the SSn# pin goes low. Therefore, make sure that no mode fault error has occurred before starting communication, and start communication, and make sure that no mode fault error has occurred even after communication ends. If a mode fault error has occurred, communication may be incomplete, so measures such as retransmission are required. The other procedures are the same as in clock synchronous mode.
In slave mode, it operates according to the SSn# pin input level. Other steps are the same as those of clock synchronous mode.

36.11.7 Reception Sampling Timing Adjustment Function in Simple-SPI Mode with Internal Clock Used

The reception sampling timing adjustment function in simple SPI mode is the same as the reception sampling timing adjustment function in clock synchronous mode. For the description of operation, see section 36.10.7, Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with Internal Clock Used.

36.12 Bit Rate Modulation Function

Using the bit rate modulation function, the bit rate can be corrected by evenly enabling the clocks of the number specified in the SCR2.MDDR[7:0] bits among 256 clock cycles of internal clocks which is selected by the CKS[1:0] bits in SCR2 register.
Figure 36.109 shows an example where the PCLKA is selected by the CKS[1:0] bits in SCR2 register and the BRR[7:0] bits and MDDR[7:0] bits are set to 0 and 160 respectively in asynchronous mode. In this example, the cycle of the base clock is evenly corrected (256/160) and the bit rate is also corrected (160/256). Note that there is an imbalance in enabling the internal clock, and expansion and contraction occur in the pulse width of the base clock.

Note: Do not use this function in clock synchronous mode, simple-SPI mode, smart card Interface mode, manchester mode and extended serial mode.

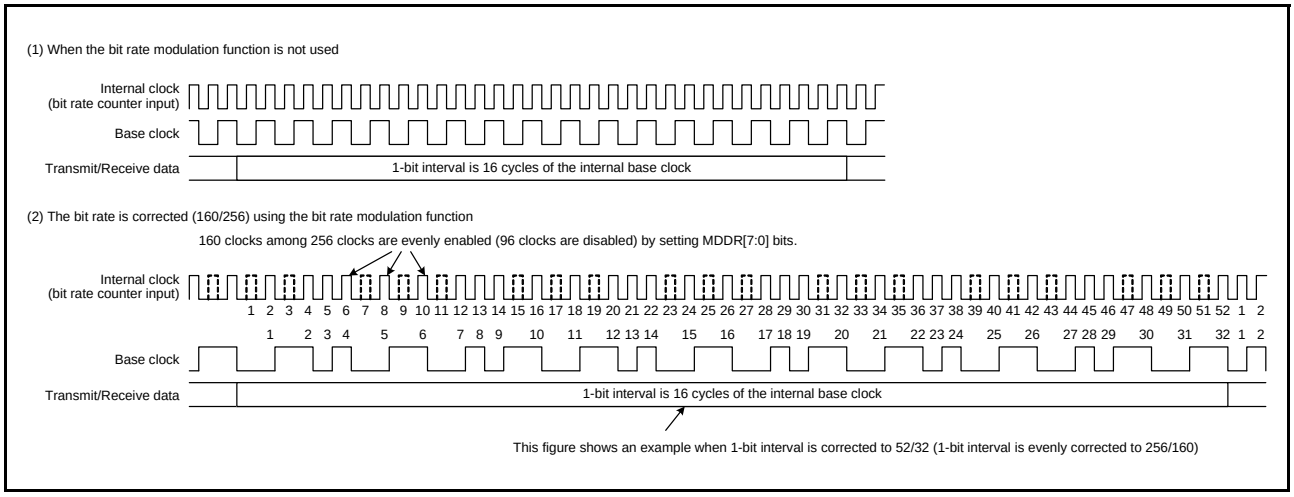


Figure 36.109 Example of Base Clock when Using the Bit Rate Modulation Function

36.13 Noise Cancellation Function

Figure 36.110 shows the configuration of the noise filter used for noise cancellation. The noise filter consists of a two-stage flip-flop circuits and a match detection circuit. When the input signals of the noise filter and the output signals of the two-stage flip-flop circuits completely match, the matched level is conveyed as an internal signal. Unless otherwise matched, the previous value is retained. (When the same level is retained for three cycles or longer on the sampling clock of the noise filter, it is considered as a valid receive signal. A change in pulse for three cycles or shorter is considered as a noise, not as a receive signal).

In asynchronous mode, manchester mode and extended serial mode, the noise cancellation function can be applied to the receive signal input to the RXDn pin. The sampling period of the noise filter can be selected from the base clock period and the divided clock of the baud rate generator clock source by SCR1.NFCS[2:0] bits.

(When SCR1.NFCS[2:0] bits = 000b, SCR2.ABCS bit = 0 and SCR2.ABCSE bit = 0, the cycle is 1/16 of a period 1 transfer bit.

When SCR1.NFCS[2:0] bits = 000b, SCR2.ABCS bit = 1 and SCR2.ABCSE bit = 0, the cycle is 1/8 of a period 1 transfer bit.

When SCR1.NFCS[2:0] bits = 000b, SCR2.ABCSE bit = 1, the cycle is 1/6 of a period 1 transfer bit.)

In simple I^2C mode, the noise elimination function can be used for the input pins of TXDn/SSDAn and RXDn/SSCLn. The sampling period of the noise filter can be selected from the divided clock of the baud rate generator clock source by SCR1.NFCS[2:0] bits.

If the base clock is stopped once with the noise filter enabled and then the base clock input is restarted again, the noise filter operation resumes from the state where the clock was stopped. When SCR0.TE and SCR0.RE bits 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 as an 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 sampling cycles.

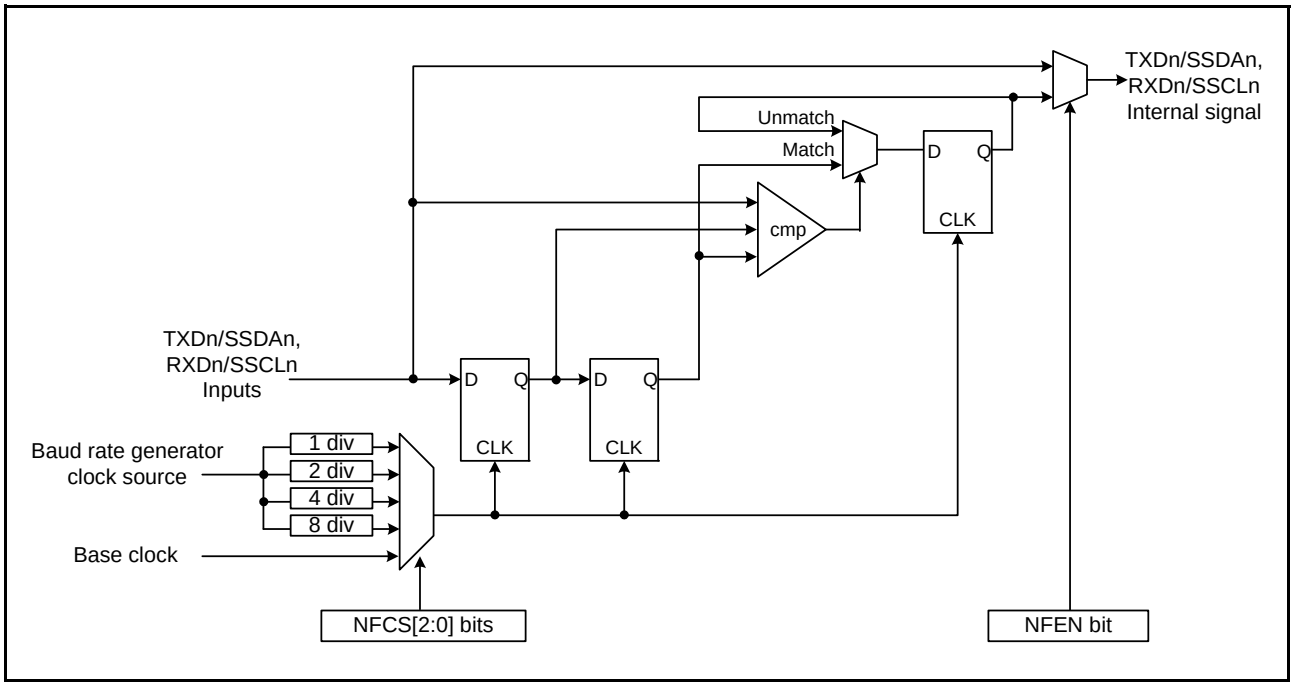


Figure 36.110 Block Diagram of Digital Noise Filter

36.14 RS-485 Driver Control Function

Setting the DEEN bit in the RSCI control register 3 (SCR3) to 1 enables the RS-485 driver control function and generates a DE (Driver Enable) signal that enables the external transceiver transmission mode. The DE signal outputs a valid level for the period with setup time and hold time added before and after data transmission. The DE signal valid level is set by the DELVL bit in the DE signal control register (DECR).

The setup time is the time from when the DE signal is valid until the start bit starts. Set by DESU[4:0] bits of DE signal control register (DECR).

The hold time is the time from the end of the last stop bit of the transmitted message to the invalidation of the DE signal. Set with DEHLD[4:0] bits of the DE signal control register (DECR).

DESU[4:0] and DEHLD[4:0] bits are expressed in base clock units (1/8 or 1/16 bit time). For details, refer to section 36.2.13, DE Signal Control Register (DECR).

When this function is used (DEEN bit = 1), the TEND set timing and TEI interrupt output timing are at the end of the DE signal hold time.

When transmission is completed and the next transmission data is not written before the DE signal is negated, the DE signal is negated once. If the timing for writing the next transmit data is not in time, the DE signal is negated and asserted again, the setup time is inserted, and the next data is transmitted. If you want to perform the next transmission with the DE signal asserted, write the next transmission data to the TDR quickly enough.

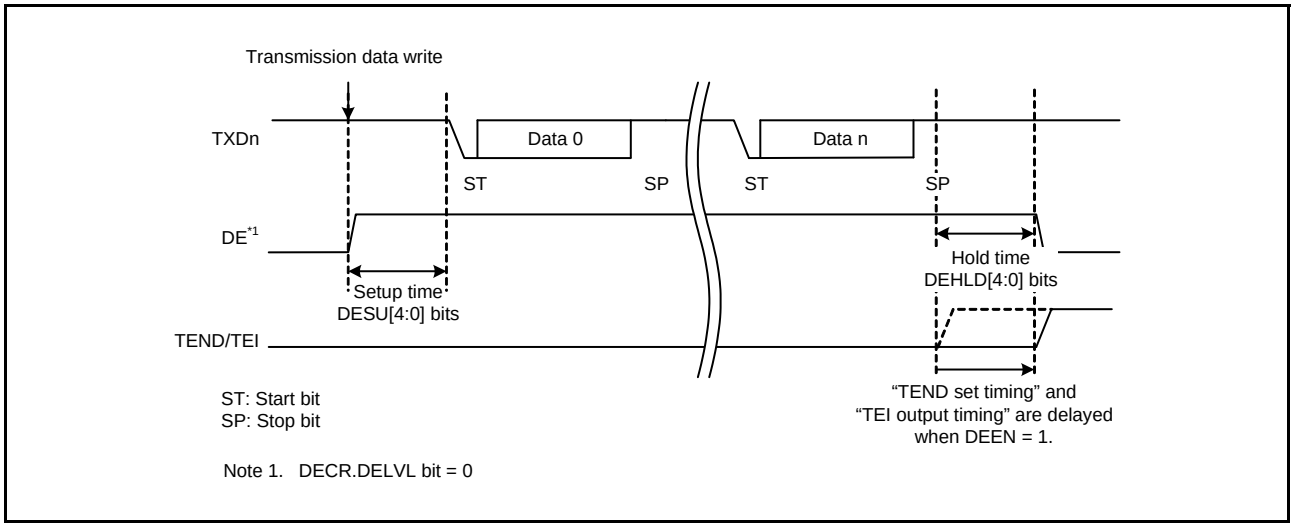


Figure 36.111 Image Waveform for RS-485 Driver Control DE Signal Output

36.15 Loopback Function

The loopback function can be used in Asynchronous mode with the internal clock, and manchester mode with the internal clock, and Clock synchronous mode with the internal clock.
When 1 is written to the LOOP bit in the SCR1 register, RSCI blocks the external input (RXD) path and connects the output path of the transmit data register and the input path of the receive data register.
When this function is used with TINV bit = 1, inversion of transmission data becomes reception data. However, this function can be used with TINV bit = 1 only when operating in clock synchronous mode internal clock.
Table 36.41 shows the relationship between the TINV and LOOP bit settings and the received data.

Table 36.41 TINV and LOOP Bit Settings and Received Data

			Communication Mode		
TINV	**LOOP**	**Receive Data**	**Async Internal Clock**	**Manchester Internal Clock**	**Clock Sync Internal Clock**
0 or 1	0	Receive data from RXDn pin	Possible	Possible	Possible
0	1	Transmit data	Possible	Possible	Possible
1	1	Inverted transmit data	Impossible	Impossible	Possible

Figure 36.112 shows the configuration of the shift register input/output path in loopback mode.

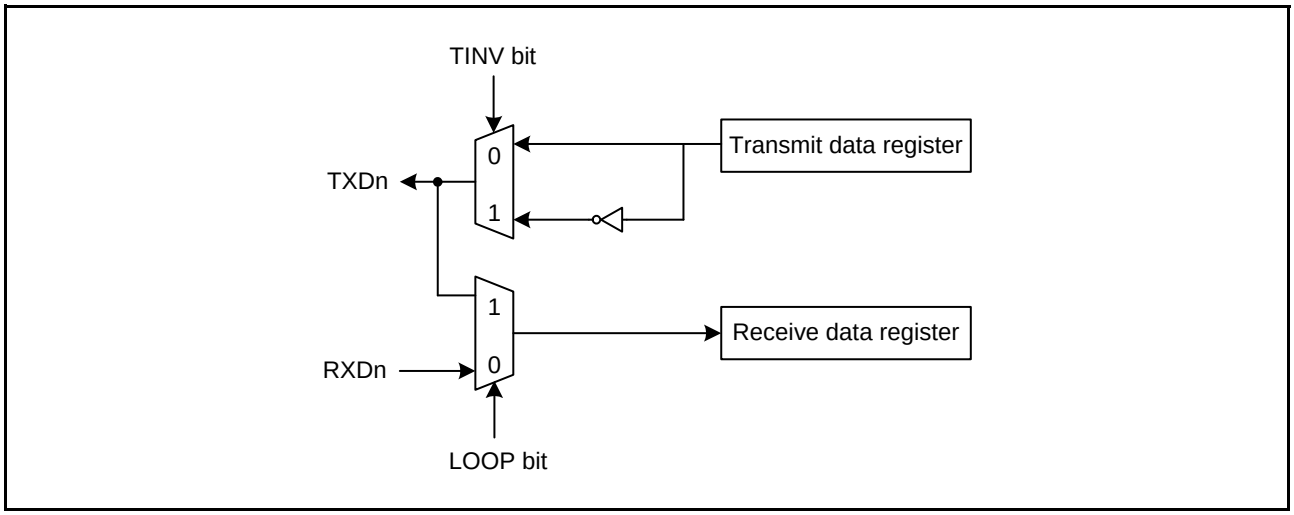


Figure 36.112 Shift Register Input/Output Configuration Image in Loopback Mode

36.16 Half-Duplex Communication Function

Do not use the half-duplex communication function in simple I^2C mode, simple-SPI mode and smart card interface mode.

In other communication modes, if the SCR1.HDSEL bit is set to 1, half-duplex communication using the TXDn pin is possible. When half-duplex communication is used, transmission and reception must be performed exclusively. Transmission and reception settings (SCR0.TE bit = 1 and SCR0.RE bit = 1) is prohibited.

However, if half-duplex communication is performed as the master reception in clock synchronous mode, perform transmission/reception settings (SCR0.TE bit = 1 and SCR0.RE bit = 1) and perform dummy transmission. By dummy transmission (arbitrary transmission data is written to TDR), SCKn is output and reception is enabled. The dummy transmission data is discarded inside the RSCI and is not actually transmitted.

During half-duplex communication, the only communication port terminal used is the TXDn pin. Output when SCR0.TE bit = 1, input when SCR0.TE bit = 0.

36.17 Interrupt Signal

Table 36.42 lists RSCI interrupt signals.

The interrupt explanation corresponding to each operation mode is described in sections 36.17.2 to 36.17.5. Also, TXI and RXI have an interrupt buffer function. Refer to section 36.17.1, Buffer Operations for TXI and RXI Interrupts. When performing transmission and reception using DTC or DMAC, be sure to set DTC or DMAC first, and then enable RSCI before setting. Refer to section 20, Data Transfer Controller (DTCb) and section 18, DMA Controller (DMACAb) for how to set DTC or DMAC.

Table 36.42 RSCI Interrupt List

Interrupt Symbol	Interrupt Function	Pulse/ Level	Pulse Width	Active Level	SYNC Clock	Note
ERI	Error interrupt Bus collision detection interrupt	Level	—	Low	PCLKA	
RXI	Simple I²C: Reception end interrupt Other mode: Receive data full interrupt	Pulse	1 cycle	Low	PCLKA	
TXI	Simple I²C and smart card interface: Transmit end interrupt Other mode: Transmit data empty interrupt, Break Field transmission completion	Pulse	1 cycle	Low	PCLKA	
TEI	Simple I²C: Completion of generation of a start, restart, or stop condition (STI) Other mode: Transmit end interrupt	Level	—	Low	PCLKA	
AED	Active edge detection interrupt	Pulse	1 cycle	Low	PCLKA	
BFD	Break Field detection interrupt	Level	—	Low	PCLKA	Only when extended serial mode

36.17.1 Buffer Operations for TXI and RXI Interrupts

The TXI and RXI interrupts have an interrupt buffer function. When the first interrupt request is generated during interrupt handling and the next interrupt request is generated (when the status flag of the interrupt controller (ICU) is 1), the RSCI does not output the interrupt request, and holds it internally. The interrupt that can be held is up to one.

36.17.2 Interrupt in Asynchronous Mode, Manchester Mode, Clock Synchronous Mode, and Simple SPI Mode

A different interrupt vector is assigned to each interrupt source, and individual interrupt sources can be enabled or disabled with the enable bits in SCR0 register.

(1) Non-FIFO Mode

Table 36.43 lists interrupt sources in asynchronous mode, manchester mode, clock synchronous mode, and simple SPI mode with FIFO disabled.

If the SCR0.TIE bit is 1, a TXI interrupt request is generated when transmit data is transferred from the TDR register to the TSR register. At the start of transmission, a TXI interrupt request can also be generated by using a single instruction to set the SCR0.TE and SCR0.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 SCR0.TIE bit to 1 while the setting of the SCR0.TE bit is 1.*1

When the SCR0.TEIE bit is 1, the SSR.TEND flag becomes 1 and the TEI interrupt request is generated if the next data is not written to the TDR register by the timing to transmit the last bit of transmission data. In addition, the TEND flag holds 1 during the period from setting the SCR0.TE bit to 1 until writing transmit data to the TDR register, and if the TEIE bit is set to 1, a TEI interrupt request is generated.

Writing data to the TDR register clears the TEND flag and cancels the TEI interrupt request, but it takes time to cancel it.

If the SCR0.RIE bit is 1, an RXI interrupt request is generated when received data is stored in the RDR register. An RXI interrupt request can activate the DTC or DMAC to handle data transfer.

Setting of any from among the ORER, AFER, and APER flags in the SSR register or the MCER, SYER (if SYERIE = 1)*2, PFER (if PFERIE = 1)*2, and SBER (if SBERIE = 1)*2 flags in MMSR register to 1 while the SCR0.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 flags (ORER, AFER, APER, MCER, SYER (if SYERIE = 1)*2, PFER (if PFERIE = 1)*2, and SBER (if SBERIE = 1)*2) 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 transmit end interrupt, control prohibiting and permitting of the interrupt by using the interrupt request enable bit in the interrupt controller rather than using the TIE bit. This can prevent the suppression of TXI interrupt requests in the transfer of new data.

Note 2. In manchester mode only, MMSR.SYER (if SYERIE bit = 1), PFER (if PFERIE bit = 1), SBER (if SBERIE bit = 1) flags are added to the ERI interrupt sources.

(2) FIFO Mode

Table 36.44 lists interrupt sources in asynchronous mode, manchester mode, clock synchronous mode, and simple SPI mode with FIFO enabled.

If the SCR0.TIE bit is 1, a TXI interrupt request is generated when the stored number of data in transmit FIFO becomes equal to or less than the transmit FIFO threshold. A TXI interrupt request can also be generated by using a single instruction to set the SCR0.TE and SCR0.TIE bit to 1 at the same time. A TXI interrupt request is not generated by setting the SCR0.TE bit to 1 while the setting of the SCR0.TIE bit is 0 or by setting the SCR0.TIE bit to 1 while the setting of the SCR0.TE bit is 1.

If SCR0.TEIE bit is 1, when the next data isn't being written in transmit FIFO by the timing to which the last bit of the transmission data is sent, SSR.TEND flag will be 1 and TEI interrupt request is generated.

If the SCR0.RIE bit is 1, RXI interrupt request is generated when the stored number of data in receive FIFO exceeds the threshold. When the threshold value is set to 0, RXI interrupt request is occurred if the quantity of data in receive FIFO is greater than or equal to 1.

If the SCR0.RIE bit is 1, when SSR.ORER flag is set to 1 or the data a framing error or a parity error generated is stored in receive FIFO, ERI interrupt request is generated.

When the number of data stored in a receive FIFO at this time is a threshold value or above, RXI interrupt request is also

generated. The ERI interrupt request can be canceled by clearing all flags (SSR.ORER, AFER, APER).

Table 36.43 RSCI Interrupt Sources with FIFO Disabled

Name	Interrupt Source	Interrupt Flag	Interrupt Enable	DTC/DMAC Activation
ERI	Receive error	ORER, AFER, APER, DFER, DPER, MCER, SYER (SYERIE = 1), PFER (PFERIE = 1), SBER (SBERIE = 1)	RIE	Not possible
RXI	Receive data full	RDRF	RIE	Possible
	Receive data match	DCMF		
TXI	Transmit data empty	TDRE	TIE	Possible
	TE = 0 → 1 detection			
TEI	Transmit end	TEND	TEIE	Not possible

Table 36.44 RSCI Interrupt Sources with FIFO Enabled

Name	Interrupt Source	Interrupt Flag	Interrupt Enable	DTC/DMAC Activation
ERI	Receive error	ORER, AFER, APER, DFER, DPER, DR (FCR.DRES bit = 1)	RIE	Not possible
RXI	Receive FIFO data full	RDRF	RIE	Possible
	Receive data ready	DR (FCR.DRES bit = 0)		
	Receive data match	DCMF		
TXI	Transmit FIFO data empty	TDRE	TIE	Possible
	TE = 0 → 1 detection			
TEI	Transmit end	TEND	TEIE	Not possible

36.17.3 Interrupt in Smart Card Interface Mode

Table 36.45 lists interrupt sources in smart card interface mode. A transmit end interrupt (TEI) request cannot be used in this mode.

Table 36.45 RSCI Interrupt Sources in Smart Card Interface Mode

Name	Interrupt Source	Interrupt Flag	Interrupt Enable	DTC/DMAC Activation
ERI	Receive error or error signal detection	ORER, APER, ERS	RIE	Not possible
RXI	Receive data full	RDRF	RIE	Possible
TXI	Transmit end	TNED	TIE	Possible
	TE = 0 → 1 detection			

Data transmission/reception using the DTC or DMAC is also possible in smart card interface mode. In transmission operation, when the SSR.TEND flag is set to 1, a TXI interrupt request is generated. This TXI interrupt request activates the DTC or DMAC allowing transfer of transmit data if the TXI request is specified beforehand as a source of DTC or DMAC activation. The TEND flag is automatically set to 0 when the DTC or DMAC transfers the data.
If an error occurs, the RSCI automatically retransmits the same data. During the retransmission, the TEND flag is kept to 0 and the DTC or DMAC is not activated. Therefore, the RSCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, the SSR.ERS flag is not automatically cleared to 0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the SCR0.RIE bit to 1 to enable an ERI interrupt request to be generated at error occurrence.
In reception operation, an RXI interrupt request is generated when receive data is set to RDR register. 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.
When transmitting/receiving data using the DTC or DMAC, be sure to make settings to enable the DTC or DMAC before making RSCI settings. For DTC or DMAC settings, refer to section 20, Data Transfer Controller (DTCb) and section 18, DMA Controller (DMACAb).

36.17.4 Interrupts in Simple I2C Mode

Table 36.46 lists RSCI interrupts in Simple I2C mode.

The STI interrupt is allocated to the transmit end interrupt (TEI) request. The receive error interrupt (ERI) request cannot be used.

The DTC or DMAC can also be used to handle transfer in simple I2C mode.

When the value of the SIMR.IICINTM bit is 1, a RXI request will be generated on the falling edge of the SSCLn signal for the eighth bit. If the RXI has been set up as an activating request for the DTC or DMAC beforehand, the RXI request will activate the DTC or DMAC to handle transfer of the received data. Furthermore, a TXI request is generated on the falling edge of the SSCLn signal for the ninth bit (acknowledge bit). If the TXI has been set up as an activating request for the DTC or DMAC beforehand, the TXI request will activate the DTC or DMAC to handle transfer of the transmit data. (In this case, ACK/NACK judging are impossible.)

When the value of the SIMR.IICINTM bit is 0, RSCI moves as follows. RXI request (ACK detection) is generated if the input on the SSDAn pin is at the low level on the rising edge of the SSCLn signal for the ninth bit (acknowledge bit). TXI request (NACK detection) is generated if the input on the SSDAn pin is at the high level 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 RSCI.

When the SIMR.IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits are used to generate a start condition, restart condition, or stop condition, the STI request is issued when generation is complete.

Table 36.46 RSCI Interrupt Sources in Simple I2C Mode

Name	Interrupt Source		Interrupt Flag	Interrupt Enable	DTC/DMAC Activation
	IICINTM bit = 1	IICINTM bit = 0			
RXI	Reception end	—	—	RIE	Possible*1
	—	ACK detection	—		Possible
TXI	Transmit end	—	—	TIE	Possible*1
	—	NACK detection	—		Possible
STI	Completion of generation of a start, restart, or stop condition		IICSTIF	TEIE	Not possible

Note 1. If the DMAC or DTC are being used, you can not confirm whether ACK or NACK.

36.17.5 Interrupts in Extended Serial Mode

Table 36.47 lists interrupt sources in extended serial mode.

Table 36.47 RSCI Interrupt Sources in Extended Serial Mode

Name	Interrupt Source	Interrupt Flag	Flag the needs to be confirmed	Interrupt Enable	DTC/DMAC Activation
ERI	Error	ORER, AFER, APER	—	RIE	Not possible
		BCDF		BCDIE	
		COF		RIE, COFIE	
RXI	Reception data full	RDRF	CF0MF, CF1MF, PIBDF	RIE	XSR0.SFSF flag = 0: Possible XSR0.SFSF flag = 1: Not possible
AED	Active edge detection	AEDF	—	AEDIE	Possible
TXI	Transmit data empty	TDRE	—	TIE	Possible
	TE = 0 → 1 detection				
	Break Field transmission completion	BFOF	—	TIE, BFOIE	
TEI	Transmit end	TEND	—	TEIE	Not possible
BFD	Break Field detection	BFDF	—	BFDIE	Not Possible (Unnecessary)

In extended serial mode, in addition to reception errors (overrun, framing, and parity errors), an ERI interrupt request is output when a bus conflict is detected during transmission, or when a counter overflow of the extended serial module occurs. At this time, a RXI interrupt request is not output. The ERI interrupt request can be canceled by clearing all the flags.

When transmitting Start Frame, if SCR0.TIE bit = 1 and XCR0.BFOIE bit = 1, a TXI interrupt request is output when Break Field transmission is completed. When Control Field 0 data is written to the TDR register, data transmission starts. Therefore, transmission using DTC or DMAC is possible.

Set SCR0.TEIE bit = 1 after writing the last transmit data to the TDR register and transmission starts.

During Start Frame reception (XSR0.SFSF flag = 1), reception using DTC or DMAC by RXI interrupt is not possible. Check the SSR register and XSR0 register, check the reception status (See Figure 36.72), and then clear the flag.When data is received, read the RDR register so that an overrun error does not occur (if you do not need to check the received data value, clear the RDRF flag without reading the RDR register). When reception of Control Field 1 is completed (XSR0.CF1MF flag = 1), Start Frame detection is disabled (XSR0.SFSF flag = 0) and reception using DTC or DMAC is possible. Be sure to read the RDR register.

When Start Frame/Break Field detection is enabled (XCR1.SDST bit = 1), if a Break Field longer than the period set in XCR2.BFLW[15: 0] bits is received, the BFDF flag is set and a BFD interrupt request is output. Then RSCI becomes the Start Frame reception state. Clear the BFDF flag.

When Start Frame/Break Field detection is enabled (XCR1.SDST bit = 1) and the bit rate measurement function is enabled (XCR1.BRME bit = 1), an AED interrupt factor is output when an active edge is detected. Read the timer count capture value (XSR1.CCV[15:0] bits).

36.18 Event Linking

By employing interrupt request signals as event signals, RSCI can provide linked operation through the event link controller (ELC) for modules selected in advance.
Event signals can be output regardless of the values of the corresponding interrupt request enable bits. And even when the next interrupt source occurs while the corresponding interrupt status flag is 1, the event signal can be output.
All event output is a pulsed output and becomes negated behind 1 PCLKA of asserting.
Table 36.48 lists RSCI event link signals.

Table 36.48 RSCI Event Link Signal List

Function	Pulse/Level	Pulse Width	Active Level	Synchronize Clock
Error event	Pulse	1 cycle	High	PCLKA
Receive data full event	Pulse	1 cycle	High	PCLKA
Receive data match event	Pulse	1 cycle	High	PCLKA
Transmit data empty event	Pulse	1 cycle	High	PCLKA
Transmit end event	Pulse	1 cycle	High	PCLKA
Receive data unmatch event	Pulse	1 cycle	High	PCLKA
Active edge detect event	Pulse	1 cycle	High	PCLKA

(1) Error Event Output (Receive Error, Error Signal Detection)

- Indicates abnormal termination due to a parity error during reception.
- Indicates abnormal termination due to a framing error during reception.
- Indicates abnormal termination due to an overrun error during reception.
- Indicates abnormal termination due to a manchester code error during reception (only in manchester mode).
- Indicates that a preface error occurred upon reception and abnormal termination occurred (only in manchester mode and MMCR.PFERIE bit = 1).
- Indicates that a start bit error occurred during reception and abnormal termination occurred (only in manchester mode and MMCR.SBERIE bit = 1).
- Indicates that a reception sync error occurred during reception and abnormal termination occurred (only in manchester mode and only when MMCR.SYERIE bit = 1).
- Indicates detection of the error signal during transmission in smart card interface mode.
- SSR.AFER and APER flags are 0, and reception data less than receive FIFO data trigger number is set in a receive FIFO buffer, and it indicates that time of 15etu has passed when FIFO enabled and FCR.DRES bit is 1.
- In extended serial mode, indicates that the 16-bit counter in the extended serial module has overflowed.
- In extended serial mode, a bus collision is detected during transmission (SCR0.TE bit = 1).

(2) Receive Data Full Event Output

- In non-FIFO mode, indicates that received data have been set in the receive data register (RDR).
- In FIFO mode, indicates that receive data has been transferred to the receive FIFO (RDR register), and the quantity of data in the register has exceeded the specified receive triggering number. If the receive FIFO threshold is set to 0, no event output occurs unless at least one data is received. However, in FIFO mode, it is prohibited because it causes inconvenience when processing events.
- When FIFO is enabled and FCR.DRES bit is 0, indicates followings.
  - SSR.AFER flag = 0 and SSR.APER flag = 0
  - Time of 15 etu has passed after the reception data less than receive FIFO data trigger number is set in receive FIFO
- Indicates that ACK has been detected if the SIMR.IICINTM bit is 0 in simple I^2C mode.

- Indicates that the 8th-bit SSCLn falling edge has been detected if the SIMR.IICINTM bit is 1 in simple I2C mode.
- When the SIMR.IICINTM bit is 1 during master transmission in simple I2C mode, set the event link controller (ELC) so that receive data full events are not used.

(3) Transmit Data Empty Event Output

- Indicates that the SCR0.TE bit has been changed from 0 to 1.
- In non-FIFO mode, indicates that transmit data have been transferred from the transmit data register (TDR) to the transmit shift register (TSR).
- In FIFO mode, indicates that the quantity of data in transmit FIFO (TDR register) has fallen below the specified transmit triggering number. However, in FIFO mode, it is prohibited because it causes inconvenience when processing events.
- Indicates that transmission has been completed in smart card interface mode.
- Indicates that NACK has been detected if the SIMR.IICINTM bit is 0 in simple I2C mode.
- Indicates that the ninth-bit SSCLn falling edge has been detected if the SIMR.IICINTM bit is 1 in simple I2C mode.
- In extended serial mode, indicates that Break Field transmission is complete.

(4) Transmit End Event Output

- Indicates the completion of transmission. In FIFO mode, it is prohibited because it causes inconvenience when processing events.
- Indicates that the starting condition, resumption condition, or termination condition has been generated in simple I2C mode.
- In smart card interface mode, the transmit end event is not output.

(5) Receive Data Match Event Output

- Indicates the match the comparison data (SCR4.CMPD[8:0] bits) with receive data that is one frame of data has been received, when SCR0.DCME bit is set to 1 in asynchronous mode (including multi-processor mode).

(6) Receive Data Unmatch Event Output

- Indicates the unmatch the comparison data (SCR4.CMPD[8:0] bits) with receive data that is one frame of data has been received, when SCR0.DCME bit is set to 1 in asynchronous mode (including multi-processor mode).

(7) Active Edge Detection Event Output

- In extended serial mode, when XCR1.BRME bit is 1, it indicates that a valid edge has been detected in the RXD input signal.

36.19 Usage Notes

36.19.1 Setting the Module Stop Function

Module stop control register D (MSTPCRD) is used to stop and start RSCI operations. With the value after a reset, RSCI 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.

36.19.2 RSCI Operations during Low Power Consumption State

(1) Transmission

Before using the power consumption reduction function to reduce RSCI's power consumption, please do the following to confirming transmission end (SSR.TEND flag = 1):

- Set the output terminal state after transmission operation is stopped by SCR1.SPB2DT and SPB2IO bits.
- Stop the transmission (SCR0.TIE bit = 0, TE bit = 0, TEIE bit = 0)

When transitions to these states are made during transmission, the data being transmitted become indeterminate.
To transmit data in the same operating mode after cancellation of the low power consumption state, set the TE bit to 1, read SSR register, and write data to TDR sequentially to start data transmission. To transmit data with a different operating mode, initialize the RSCI first.
To start transmission using the DMAC/DTC after cancellation from software standby mode, set the TE and TIE bit to 1 simultaneously. The TXI interrupt is generated and transmission starts using the DMAC/DTC.
Figure 36.113 shows a sample flowchart for transition to software standby mode during transmission. Figure 36.114 and Figure 36.115 show the port pin states during transition to software standby mode.

(2) Reception

Before specifying the module stop state or making a transition to software standby mode, stop the receive operations (SCR0.RE bit = 0). If transition is made during data reception, the data being received will be invalid.
Figure 36.116 shows a sample flowchart for reception to software standby mode during reception.

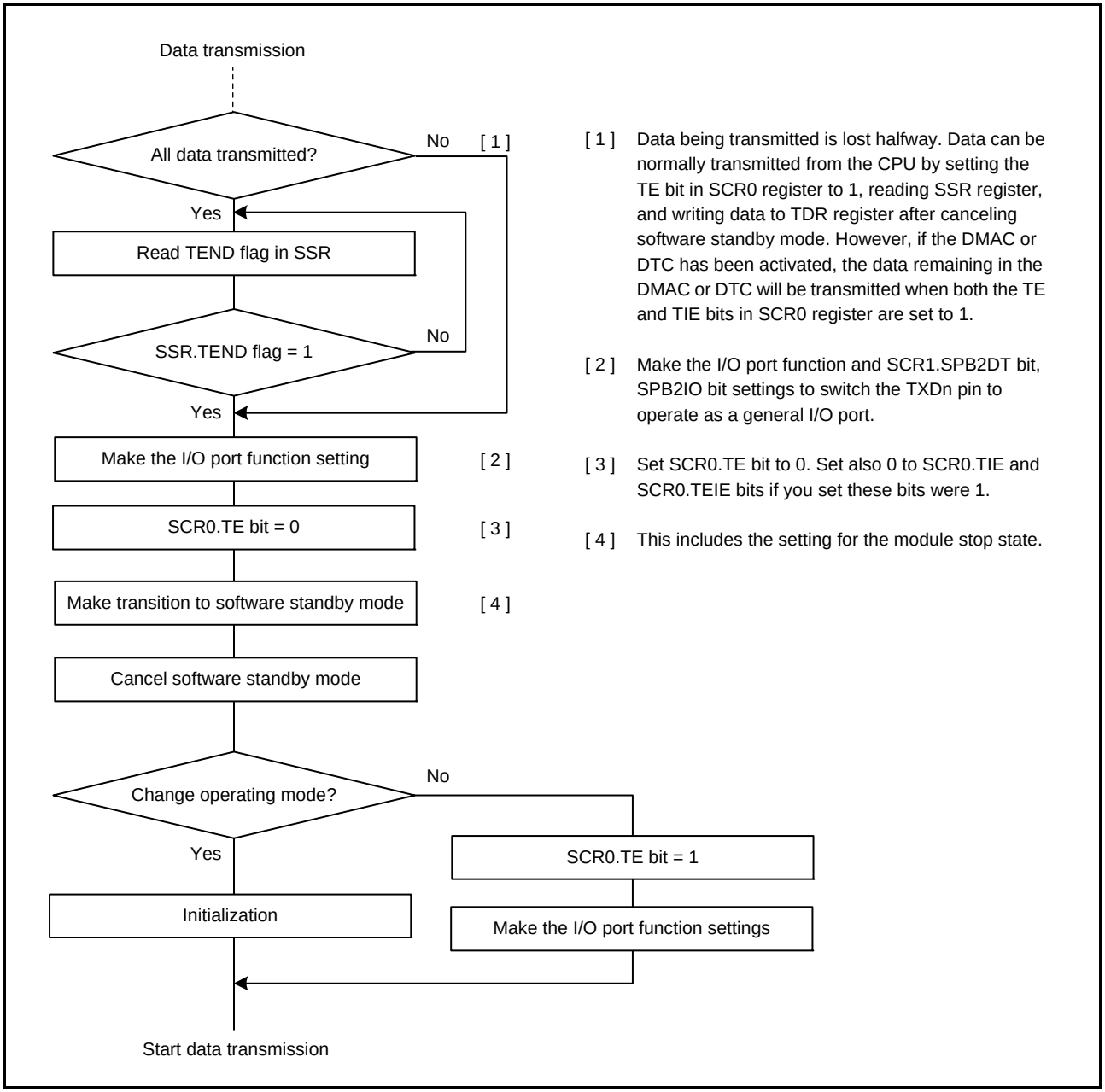


Figure 36.113 Example of Flowchart for Transition to Software Standby Mode during Transmission

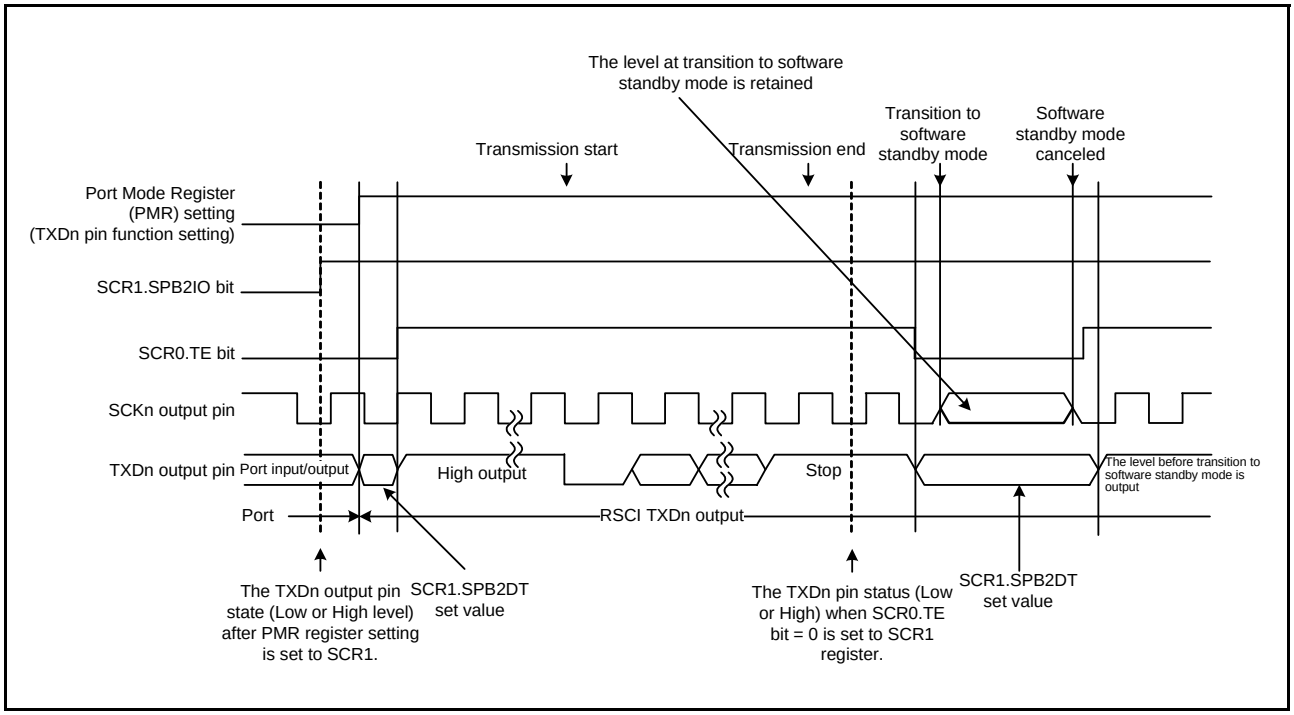


Figure 36.114 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission)

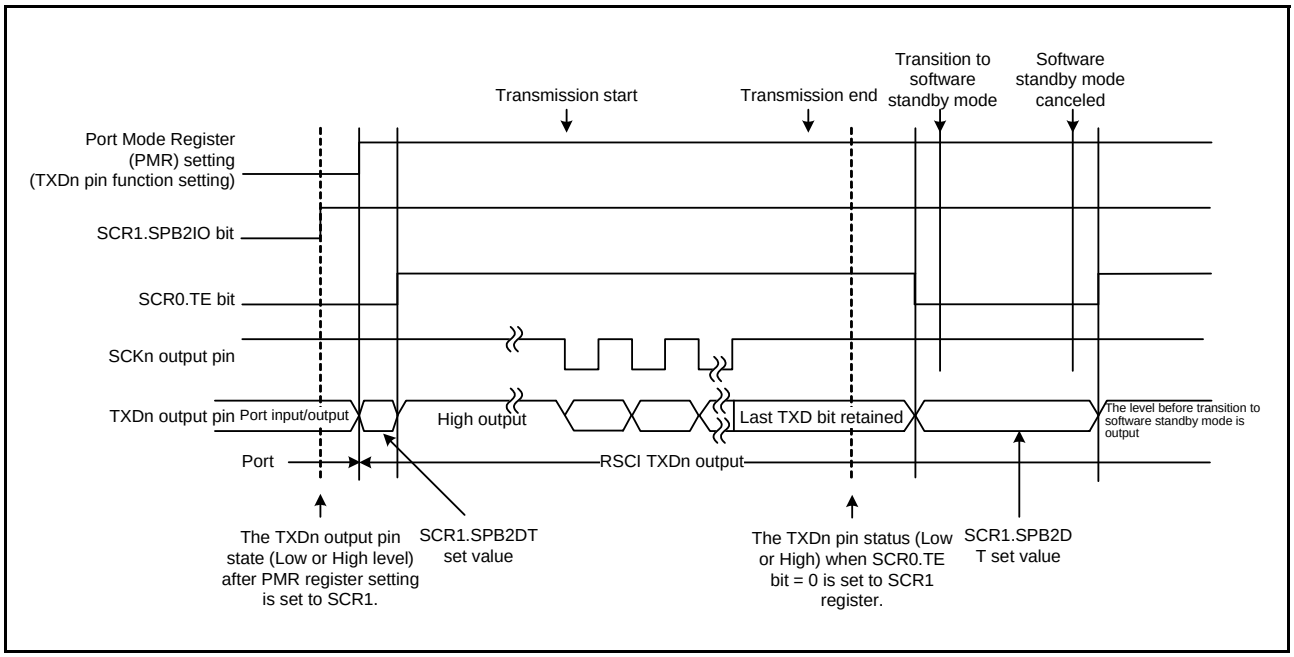


Figure 36.115 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission)

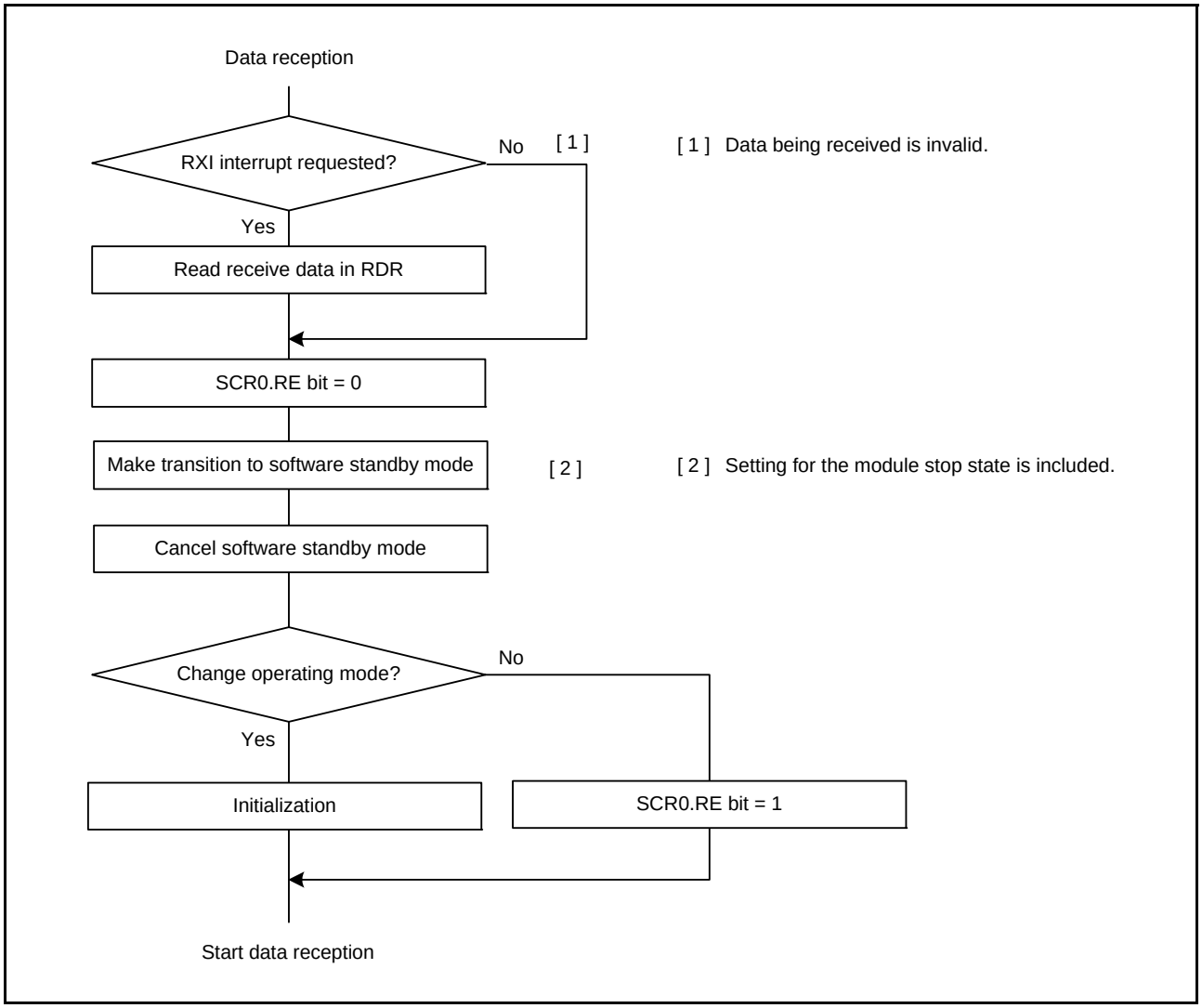


Figure 36.116 Example of Flowchart for Transition to Software Standby Mode during Reception

36.19.3 Break Detection and Processing

(1) Non-FIFO Mode

When a framing error is detected, a break can be detected by reading SSR.RXDMON flag value. In a break, the input from the RXDn pin becomes all low, and so the SSR.AFER flag is set to 1 (framing error has occurred), and the SSR.APER flag may also be set to 1 (parity error has occurred). When the SCR3.RXDESEL bit is 0, the RSCI continues the receive operation even after a break is received. Therefore, note that even if the AFER flag is set to 0 (no framing error occurred), it will be set to 1 again. When the SCR3.RXDESEL bit is 1, the RSCI sets the SSR.AFER flag to 1 and stops receiving operation until a start bit of the next data frame is detected. If the SSR.AFER flag is set to 0 at this time, the SSR.AFER flag retains 0 during the break. When the RXDn pin becomes high and the break ends, detecting the start bit at the first falling edge of the RXDn pin allows the RSCI to start the receiving operation.

(2) FIFO Mode

After a framing error is detected, when RSCI detects that continuous receive data is low for 1 frame, the data stored into the receive FIFO (RDR register) and reception stops. When a framing error is detected, a break can be detected by reading SSR.RXDMON flag value. After the RXD signal is in the mark state and it has finished the break, a stock of reception data to the receive FIFO (RDR register) resumes.

36.19.4 Mark State and Sending Breaks

When the SCR0.TE bit is 0 (serial transmission is disabled), the state of the TXDn pin can be set by SCR1.SPB2IO bit and SCR1.SPB2DT bit. Using this, it's possible to do a TXDn pin in the mark state and send a break out. When you want to make a communication-line is in the mark state (the state of 1) until the SCR0.TE bit is set to 1 (serial transmission is enabled). First, set it as High-level output by setting SPB2IO bit and SPB2DT. Next, it's changed to a TXDn pin by I/O port function. On the other hand, if you want to send a break when sending data, set the SCR0.TE bit to 0 after setting the SPB2IO and SPB2DT bits in the SCR1 register to low level output. Setting the TE bit to 0 initializes the transmitter regardless of the current transmission status.

36.19.5 Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode)

Transmission can be start by writing transmit-data to TDR register even if SSR.ORER flag is 1. However, reception can not be started. Note also that the receive error flags cannot be set to 0 even if the SCR0.RE bit is set to 0 (serial reception is disabled).

36.19.6 Writing Data to the TDR Register

(1) Non-FIFO Mode

Data can be written to TDR register anytime when TE bit is 1. However, if new data is written to TDR register when transmit data is remaining in TDR register, the previous data in TDR register is lost because it has not been transferred to TSR register yet. If you use DTC or DMAC, be sure to write transmit data to TDR register in the TXI interrupt request handling routine.

(2) FIFO Mode

Data can be written to transmit FIFO (TDR register) when TE bit is 1. Check the number of writable data with the TFSR.T[5:0] bits.

36.19.7 Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode and Simple SPI Mode)

When the external clock source is used as a synchronization clock, the following restrictions apply.

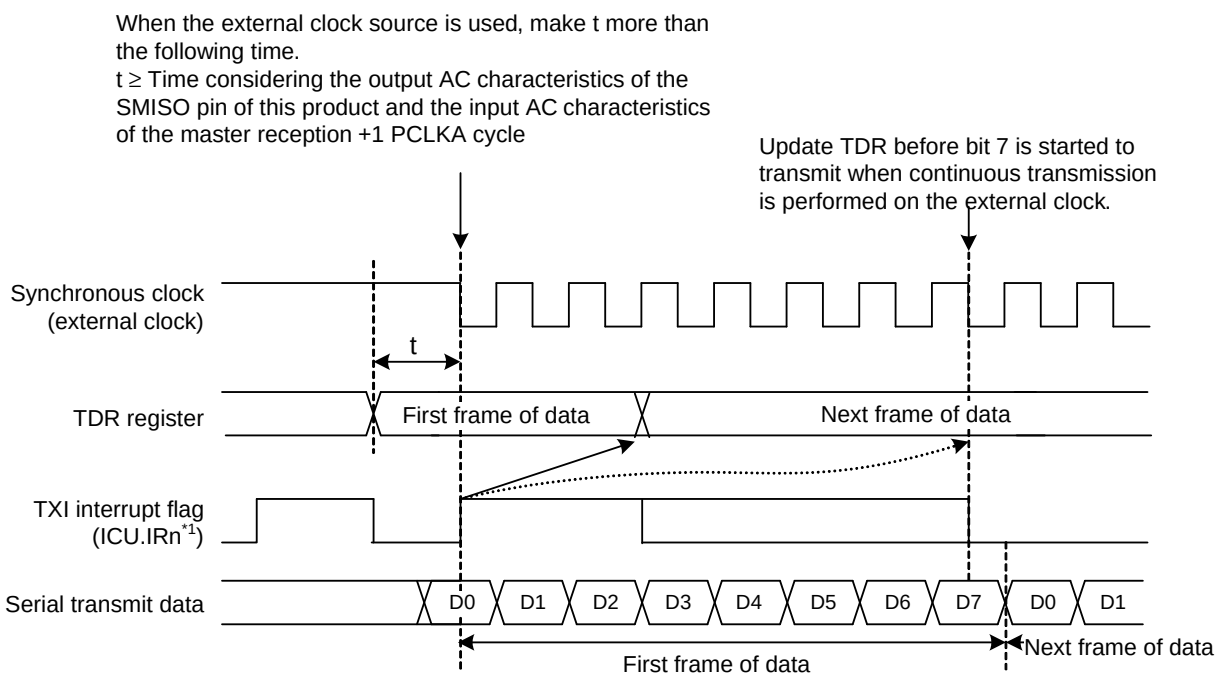
(1) Start of Transmission

Update TDR register by the CPU, DMAC, or DTC and wait at least the following time until the start of the external clock input: (See Figure 36.117)
Time considering the output AC characteristics of the SMISO pin of this product and the input AC characteristics of the master reception + 1 PCLKA cycle.

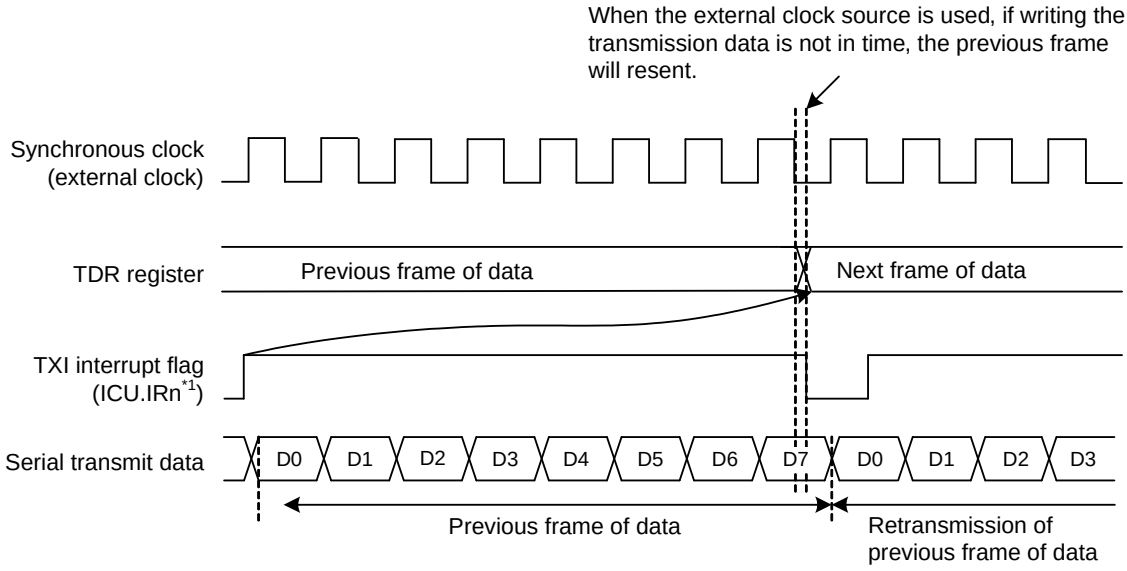
(2) Continuous Transmission

The next transmit data must be transferred to the TSR register before the falling edge*1 of the transmit clock for bit 7. Please write the transmit data to TDR register with this in mind. If the transmit data can not be written in time, the previous frame data is resent. (See Figure 36.117)

Note 1. When SCR3.CPOL bit = 1 and SCR3.CPHA bit = 0, or SCR3.CPOL bit = 0 and SCR3.CPHA bit = 1. In the case of SCR3.CPOL bit = 0 and SCR3.CPHA bit = 0, or SCR3.CPOL bit = 1 and SCR3.CPHA bit = 1, it's the rising edge.



(1) Start of transmission and (2) Continuous transmission (When transmission data is in time)



(2) Continuous transmission (When transmission data is not in time)

Note 1. Refer to section 15, Interrupt Controller (ICUE) for details on the corresponding interrupt vector number.

Figure 36.117 Restrictions on Use of External Clock in Clock Synchronous Transmission

36.19.8 Restrictions on Using DMAC or DTC

When using the DMAC or DTC to read RDR register, be sure to set the receive data full interrupt (RXI) as the activation source of the relevant RSCI.
During the operation in transmission/reception using the DMAC or DTC, it should not set transfer information of DMAC/DTC.

36.19.9 Notes on Starting Transfer

At the point where transfer starts when the interrupt status flag (IRn.IR flag) in the interrupt controller is 1, follow the procedure below to clear interrupt requests before permitting operations (by setting the SCR0.TE or SCR0.RE bit to 1). For details on the interrupt status flag, refer to section 15, Interrupt Controller (ICUE).

- Confirm that transfer has stopped (the setting of the SCR0.TE or SCR0.RE bits is 0).
- Set the corresponding interrupt enable bit (SCR0.TIE or SCR0.RIE) to 0.
- Read the corresponding interrupt enable bit (SCR0.TIE or SCR0.RIE bit) to check that it has become 0.
- Set the interrupt status flag (IRn.IR flag) in the interrupt controller to 0.

36.19.10 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 SCR3.CPHA and CPOL bits when the SCR0.SSE bit is 1. This prevents the clock line from being placed in the high-impedance state when the SCR0.TE bit is set to 0 or unexpected edges from being generated on the clock line when the SCR0.TE bit is changed from 0 to 1.
  In a single-master configuration, pull up or pull down the clock line is not necessary because the clock line does not become high-impedance state when SCR0.SSE bit = 0 and SCR0.TE bit = 0.
- In the case of the setting for clock delay (SCR3.CPHA bit is 0), the receive data full interrupt (RXI) is generated before the final clock edge on the SCKn pin as indicated in Figure 36.118. If the TE and RE bits become 0 at this time before the final edge of the clock signal on the SCKn pin, the SCKn pin is placed in the high-impedance state, so the width of the last clock pulse of the transfer clock is shortened. Furthermore, an RXI interrupt may lead to the input signal on the SSn# pin of a connected slave going to the high level before the final edge of the clock signal on the SCKn pin, leading to incorrect operation of the slave.
- In a multi-master configuration, take care because the SCKn pin output becomes high-impedance while the input on the SSn# pin is at the low level if a mode fault error occurs as the current character is being transferred. And 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.

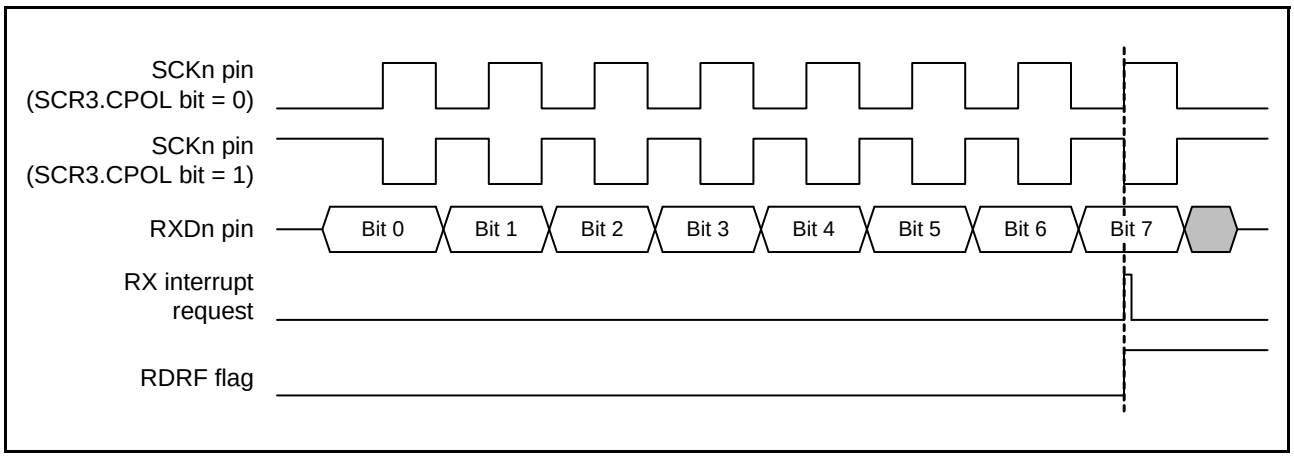


Figure 36.118 Timing of the RXI Interrupt in Simple SPI Mode (with Clock Delay)

(2) Slave Mode

- It takes “1 PCLKA + data output delay time (AC characteristics)” from writing the transmit data to the TDR register until the data is output to the RXDn pin. Take these into account when starting external clock input.
- Provide an external clock signal to the master the same as the data length for transfer.
- Secure the SS input setup time (AC characteristics) from the SSn# low-level input to the start of external clock input.
- 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, the transmission and reception is stopped immediately. Set the TE and RE bits in the SCR0 register to 0 and, after remaking the settings, restart transfer of the first byte.

36.19.11 Note on Transmit Enable Bit (TE Bit)

In the initial register value, when setting the SCR.TE bit to 0 (serial transmission is disabled) while the pin function is "TXDn", output of the TXDn pin becomes high-impedance.
Prevent the TXDn line from becoming high-impedance by any of the following ways:

(a) Connect a pull-up or pull-down resistor to the TXDn line.
(b) Set the SCR1 register to determine the level of the TXDn pin when the TE bit is 0.

36.19.12 Notes on Extended Serial Mode

In extended serial mode (SCR3.MOD[2:0] bits = 110b), the following functions cannot be used.

- CTS and RTS functions
- Multi-processor communication function
- Bit Rate Modulation function
- Loopback function
- FIFO buffer

36.19.13 Notes on RS-485 Driver Control Function

- RS-485 Driver control function is valid only in Asynchronous mode.
- When RS-485 Driver control function is active (SCR3.DEEN bit = 1), the TEND set timing/TEI output timing changes as follows. Wait for the TEI interrupt and set the TE bit to 0.
  When RS-485 Driver control function is inactive: When STOP bit output is completed.
  When RS-485 Driver control function is active: At the end of DE signal hold time.

36.19.14 Notes on Loopback Function

The Loopback function is valid in Asynchronous mode with internal clock, in manchester mode with internal clock and Clock synchronous mode with internal clock.
It can also operate in the asynchronous HBS support mode, and when the HBSCR.AOE bit = 1, it loops back the signal with the logical AND of the TXDAn and TXDBn pin outputs (Use with TINV bit = RINV bit = 0).

36.19.15 Notes on Aborting Operation

If 0 is written to SCR0.RE bit during data reception and the receive operation is aborted, the status may become invalid depending on the timing. Therefore, do not use the received data (value stored in the RDR register) or the flag value of each status register. To abort the receive operation, disable the interrupt and event link related to reception, and then write 0 to the SCR0.RE bit.

37. I2C-bus Interface (RIICa)

This MCU has a three-channel I2C-bus interface (RIIC0, RIIC1, RIIC2).
The RIIC module conforms with the NXP I2C-bus (Inter-IC bus) interface and provides a subset of its functions.
In this section, “PCLK” is used to refer to PCLKB.

37.1 Overview

Table 37.1 lists the specifications of the RIIC, Figure 37.1 shows a block diagram of the RIIC. Table 37.2 lists the I/O pins of the RIIC.

Table 37.1 RIIC Specifications (1/2)

Item	Description
Communications format	• I2C-bus format or SMBus format • Master mode or slave mode selectable • Automatic securing of the various setup times, hold times, and bus-free times for the transfer rate
Transfer rate	Fast-mode Plus is supported (up to 1 Mbps)
Serial clock (SCL)	For master operation, the duty cycle of the SCL is selectable in the range from 4 to 96%.
Generating and detecting conditions	Start, restart, and stop conditions are automatically generated. Start conditions (including restart conditions) and stop conditions are detectable.
Slave address	• Up to three different slave addresses can be set. • 7-bit and 10-bit address formats are supported (along with the use of both at once). • General call addresses, device ID addresses, and SMBus host addresses are detectable.
Acknowledgment	• For transmission, the acknowledgment bit is automatically loaded. Transfer of the next data for transmission can be automatically suspended on detection of a not-acknowledge signal. • For reception, the acknowledgment bit is automatically transmitted. If a wait between the eighth and ninth clock pulses has been selected, software control of the acknowledgment in response to the received data is possible.
Wait function	• During reception, cycles of waiting by holding the SCL line low can be inserted at the following two types of timing: Waiting between the eighth and ninth clock pulses Waiting between the ninth clock pulse and the first clock pulse of the next byte
SDA output delay function	Changes in the timing of the output of data bits for transmission, and of the acknowledgment bit, can be delayed relative to the falling edge of SCL.
Arbitration	• For multi-master operation Clock synchronization for the SCL line in cases of conflict with the SCL signal from another master is possible. When generating the start condition would create conflict on the bus, loss in arbitration is detected by testing for non-matching between the internal data level and the actual level on the SDA line. During master operation, loss in arbitration is detected by testing for non-matching between the actual level on the SDA line and the internal data level. • Loss in arbitration due to detection of the start condition while the bus is busy is detectable (to prevent the generating of double start conditions). • Loss in arbitration in transfer of a not-acknowledge signal due to the internal signal (NACK) and the actual level on the SDA line not matching is detectable. • Loss in arbitration due to non-matching of internal data level and the actual level on the SDA line is detectable in slave transmission.
Timeout function	The internal timeout function is capable of detecting long-interval stop of the SCL.
Noise cancellation	The interface incorporates digital noise filters for both the SCL and SDA signals, and the width for noise cancellation by the filters is adjustable by software.
Interrupt sources	Four sources: • Communication error/communication event Detection of arbitration-lost, NACK, timeout, a start condition including a restart condition, or a stop condition • Receive data full (including matching with a slave address) • Transmit data empty (including matching with a slave address) • Transmission end

Table 37.1 RIIC Specifications (2/2)

Item	Description
Low power consumption function	Module stop state can be set.
RIIC operating modes	• Four Master transmit mode, master receive mode, slave transmit mode, and slave receive mode
Event link function (output)	Four sources (RIIC0): • Communication error/communication event Detection of arbitration-lost, NACK, timeout, a start condition including a restart condition, or a stop condition • Receive data full (including matching with a slave address) • Transmit data empty (including matching with a slave address) • Transmission end

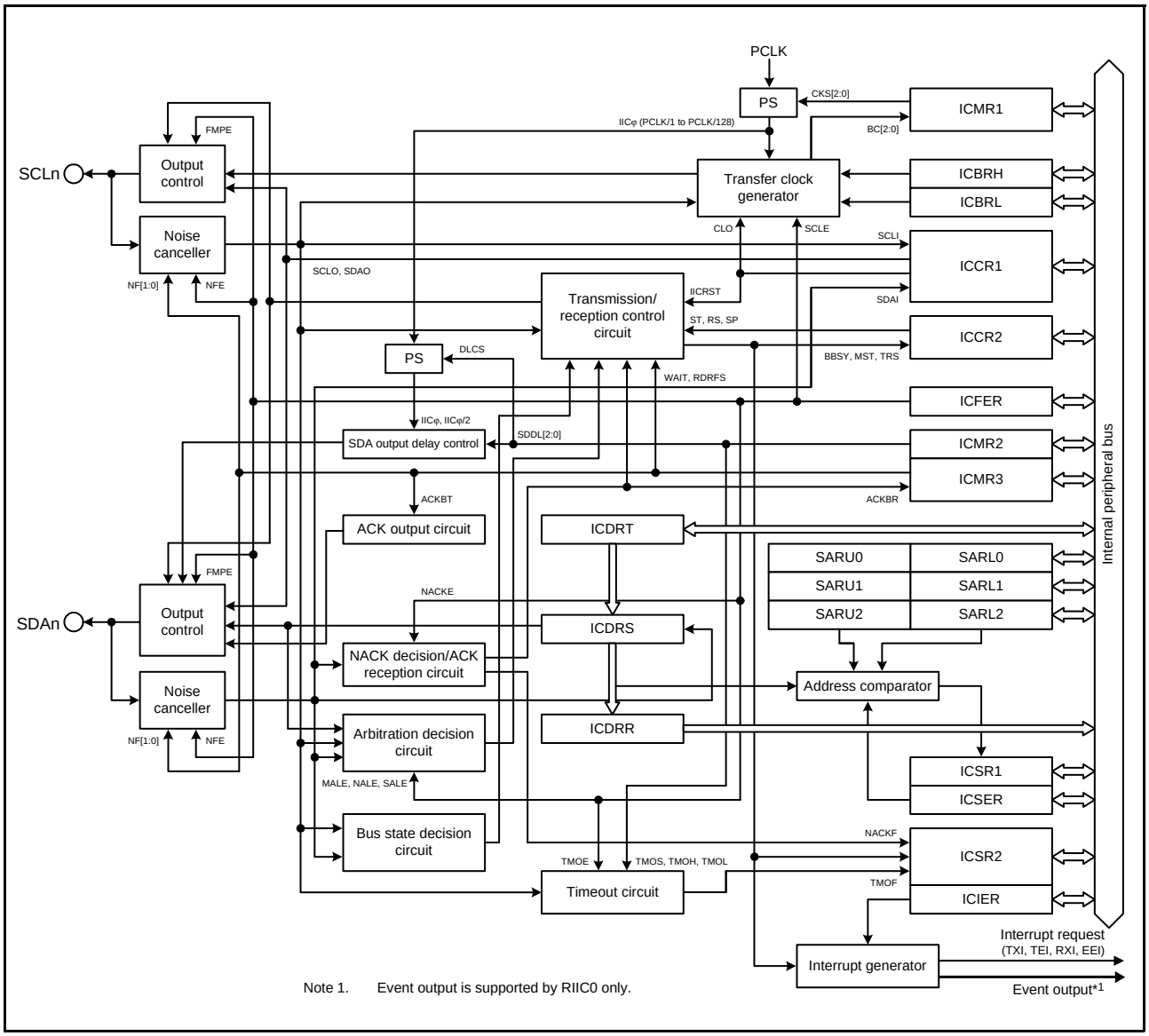


Figure 37.1 RIIC Block Diagram (n = 0 to 2)

The logic levels of the input signals for RIIC are CMOS when the I2C-bus is selected (ICMR3.SMBS bit is 0), or TTL when the SMBus is selected (ICMR3.SMBS bit is 1).

Table 37.2 RIIC Pin Configuration

Channel	Pin Name	I/O	Function
RIIC0	SCL0	I/O	RIIC0 serial clock I/O pin
	SDA0	I/O	RIIC0 serial data I/O pin
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

37.2 Register Descriptions

37.2.1 I2C-bus Control Register 1 (ICCR1)

Address(es): RIIC0.ICCR1 0008 8300h, RIIC1.ICCR1 0008 8320h, RIIC2.ICCR1 0008 8340h

	b7	b6	b5	b4	b3	b2	b1	b0
	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	SDAI	SDA Line Monitor	0: SDAn line is low. 1: SDAn line is high.	R
b1	SCLI	SCL Line Monitor	0: SCLn line is low. 1: SCLn line is high.	R
b2	SDAO	SDA Output Control/Monitor	• Read: 0: The RIIC has driven the SDAn pin low. 1: The RIIC has released the SDAn pin. • Write: 0: The RIIC drives the SDAn pin low. 1: The RIIC releases the SDAn pin. (High level output is achieved through an external pull-up resistor.)	R/W
b3	SCLO	SCL Output Control/Monitor	• Read: 0: The RIIC has driven the SCLn pin low. 1: The RIIC has released the SCLn pin. • Write: 0: The RIIC drives the SCLn pin low. 1: The RIIC releases the SCLn pin. (High level output is achieved through an external pull-up resistor.)	R/W
b4	SOWP	SCLO/SDAO Write Protect	0: SCLO and SDAO bits can be written. 1: SCLO and SDAO bits are protected. (This bit is read as 1.)	R/W
b5	CLO	Additional SCL Output	0: Does not output an additional SCL (default). 1: Outputs an additional SCL. (The CLO bit is cleared automatically after one clock pulse is output.)	R/W
b6	IICRST	I2C-bus Interface Internal Reset	0: Releases the RIIC reset or internal reset. 1: Initiates the RIIC reset or internal reset. (Clears the bit counter and the SCLn/SDAn output latch)	R/W
b7	ICE	I2C-bus Interface Enable	0: Disable (SCLn and SDAn pins in inactive state) 1: Enable (SCLn and SDAn pins in active state) (Combined with the IICRST bit to select either RIIC or internal reset.)	R/W

SDAO Bit (SDA Output Control/Monitor) and SCLO Bit (SCL Output Control/Monitor)

These bits are used to directly control the SDAn and SCLn signals output from the RIIC.

When writing to these bits, also write 0 to the SOWP bit.

The result of setting these bits is input to the RIIC via the input buffer. When slave mode is selected, a start condition may be detected and the bus may be released depending on the bit settings.

Do not rewrite these bits during a start condition, stop condition, restart condition, or during transmission or reception. Operation after rewriting under the above conditions is not guaranteed.

When reading these bits, the state of signals output from the RIIC can be read.

CLO Bit (Additional SCL Output)

This bit is used to output an additional SCL for debugging or error processing.
Normally, set the bit to 0. Setting the bit to 1 in a normal communication state causes a communication error.
For details on this function, refer to section 37.11.2, Additional SCL 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 37.3 lists the resets of the RIIC.
The RIIC reset initializes all registers and internal states of the RIIC, and the internal reset initializes the bit counter (ICMR1.BC[2:0] bits), the I2C-bus shift register (ICDRS), and the I2C-bus status registers (ICSR1 and ICSR2) as well as the internal states of the RIIC. For the reset conditions for each register, refer to section 37.14, Initialization of Registers and Functions When a Reset is Applied or a Condition is Detected.
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 it is necessary to perform an internal reset in slave mode, perform it during bus free state. If an internal reset is necessary because the RIIC has hung with the SCLn line in a low level output state in slave mode, initiate an internal reset and then generate a restart condition from the master device or resume communications from the start condition after having generated a stop condition. If communication is restarted by initiating a reset solely in the slave device without generating a start condition or restart condition from the master device, synchronization will be lost because the master and slave devices operate asynchronously.

Table 37.3 RIIC Resets

IICRST	ICE	State	Specifications
1	0	RIIC reset	Resets all registers and internal states of the RIIC.
	1	Internal reset	Resets the ICMR1.BC[2:0] bits, registers ICSR1, ICSR2, and ICDRS, and the internal states of the RIIC.

ICE Bit (I2C-bus Interface Enable)

This bit selects the active or inactive state of the SCLn and SDAn pins. It can also be combined with the IICRST bit to initiate two types of resets. See Table 37.3, RIIC Resets, for the types of resets.
Set the ICE bit to 1 when using the RIIC. The SCLn and SDAn pins are placed in the active state when the ICE bit is set to 1.
Set the ICE bit to 0 when the RIIC is not to be used. The SCLn and SDAn pins are placed in the inactive state when the ICE bit is set to 0. Do not assign the SCLn or SDAn pin to the RIIC when setting up the multi-function pin controller (MPC). Note that the slave address comparison operation is carried out if the pins are assigned to the RIIC.

37.2.2 I2C-bus Control Register 2 (ICCR2)

Address(es): RIIC0.ICCR2 0008 8301h, RIIC1.ICCR2 0008 8321h, RIIC2.ICCR2 0008 8341h

	b7	b6	b5	b4	b3	b2	b1	b0
	BBSY	MST	TRS	—	SP	RS	ST	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	ST	Start Condition Generation Request	0: Does not request to generate a start condition. 1: Requests to generate a start condition.	R/W
b2	RS	Restart Condition Generation Request	0: Does not request to generate a restart condition. 1: Requests to generate a restart condition.	R/W
b3	SP	Stop Condition Generation Request	0: Does not request to generate a stop condition. 1: Requests to generate a stop condition.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	TRS	Transmit/Receive Mode	0: Receive mode 1: Transmit mode	R/W*1
b6	MST	Master/Slave Mode	0: Slave mode 1: Master mode	R/W*1
b7	BBSY	Bus Busy Detection Flag	0: The I2C-bus is released (bus free state). 1: The I2C-bus is occupied (bus busy state).	R

Note 1. When the ICMR1.MTWP bit is set to 1, bits MST and TRS can be written to.

ST Bit (Start Condition Generation Request)

This bit is used to request transition to master mode and generation of a start condition.

When this bit is set to 1 to request to generate a start condition, a start condition is generated when the BBSY flag is set to 0 (bus free state).

For details on the start condition generation, refer to section 37.10, Start Condition/Restart Condition/Stop Condition Generating 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 generated (a start condition is detected)
- When the ICSR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: Set the ST bit to 1 (requests to generate a start condition) when the BBSY flag is set to 0 (bus free state). Note that arbitration may be lost due to a start condition generation error if the ST bit is set to 1 (requests to generate a start condition) when the BBSY flag is set to 1 (bus busy state).

RS Bit (Restart Condition Generation Request)

This bit is used to request that a restart condition be generated in master mode.
When this bit is set to 1 to request to generate a restart condition, a restart condition is generated when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode).
For details on the restart condition generation, refer to section 37.10, Start Condition/Restart Condition/Stop Condition Generating Function.
[Setting condition]

- When 1 is written to the RS bit with the ICCR2.BBSY flag set to 1

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition has been generated (a start condition is detected)
- When the ICSR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: Do not set the RS bit to 1 while generating a stop condition.
Note: If 1 (requests to generate a restart condition) is written to the RS bit in slave mode, the restart condition is not generated but the RS bit remains set to 1. If the operating mode changes to master mode with the bit not being cleared, note that the restart condition may be generated.

SP Bit (Stop Condition Generation Request)

This bit is used to request that a stop condition be generated in master mode.
When this bit is set to 1 to request to generate a stop condition, a stop condition is generated when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode).
For details on the stop condition generation, refer to section 37.10, Start Condition/Restart Condition/Stop Condition Generating Function.
[Setting condition]

- When 1 is written to the SP bit with both the BBSY flag and the ICCR2.MST bit set to 1

[Clearing conditions]

- When 0 is written to the SP bit
- When a stop condition has been generated (a stop condition is detected)
- When the ICSR2.AL (arbitration-lost) flag is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: Writing to the SP bit is not possible while the setting of the BBSY flag is 0 (bus free state).
Note: Do not set the SP bit to 1 while a restart condition is being generated.

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 1 for transmission or 0 for reception in response to the generation or detection of a start condition and setting of the R/W# bit. Although writing to the TRS bit is possible when the ICMR1.MTWP bit is set to 1, writing to this bit is not necessary during normal usage.
[Setting conditions]

- When a start condition is generated normally according to the start condition generation request (when a start condition is detected with the ST bit set to 1)
- When a restart condition is generated normally according to the restart condition generation request (when a restart condition is detected with the RS bit set to 1)
- When the R/W# bit added to the slave address is set to 0 in master mode
- When the address received in slave mode matches the address enabled in the ICSER register, with the R/W# bit set to 1
- When 1 is written to the TRS bit with the ICMR1.MTWP bit set to 1

[Clearing conditions]

- When a stop condition is detected
- The ICSR2.AL (arbitration-lost) flag being set to 1
- In master mode, reception of a slave address to which an R/W# bit with the value 1 is appended
- In slave mode, a match between the received address and the address enabled in the ICSER register when the value of the received R/W# bit is 0 (including cases where the received address is the general call address)
- In slave mode, a restart condition is detected (that is, a start condition is detected while the ICCR2.BBSY flag is 1 and the ICCR2.MST bit is 0)
- When 0 is written to the TRS bit with the ICMR1.MTWP bit set to 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

MST Bit (Master/Slave Mode)

This bit indicates master or slave mode.
The RIIC is in slave mode when the MST bit is set to 0 and is in master mode when the bit is set to 1. Combination of this bit and the TRS bit indicates the operating mode of the RIIC.
The value of the MST bit is automatically changed to 1 for master mode or 0 for slave mode by generating of a start condition and generating or detection of a stop condition, etc. Although writing to the MST bit is possible when the ICMR1.MTWP bit is set to 1, writing to this bit is not necessary during normal usage.
[Setting conditions]

- When a start condition is generated normally according to the start condition generation request (when a start condition is detected with the ST bit set to 1)
- When 1 is written to the MST bit with the ICMR1.MTWP bit set to 1

[Clearing conditions]

- When a stop condition is detected
- When the ICSR2.AL (arbitration-lost) flag is set to 1
- When 0 is written to the MST bit with the ICMR1.MTWP bit set to 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

BBSY Flag (Bus Busy Detection Flag)

The BBSY flag indicates whether the I2C-bus is occupied (bus busy state) or released (bus free state).
This bit is set to 1 when the SDAn line changes from high to low under the condition of SCLn line = high, assuming that a start condition has been generated.
The RIIC recognizes the SDAn line changing from low to high while the SCLn line is high as generation of the stop condition. After that, this flag becomes 0 if the RIIC does not detect a start condition during the bus free time (the period set in the ICBRL register).
[Setting condition]

- When a start condition is detected

[Clearing conditions]

- When the bus free time (specified in the ICBRL register) start condition is not detected after detecting a stop condition
- When 1 is written to the ICCR1.IICRST bit with the ICCR1.ICE bit set to 0 (RIIC reset)

37.2.3 I2C-bus Mode Register 1 (ICMR1)

Address(es): RIIC0.ICMR1 0008 8302h, RIIC1.ICMR1 0008 8322h, RIIC2.ICMR1 0008 8342h

	b7	b6	b5	b4	b3	b2	b1	b0
	MTWP	CKS[2:0]			BCWP	BC[2:0]		
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	BC[2:0]	Bit Counter	b2 b0 0 0 0: 9 bits 0 0 1: 2 bits 0 1 0: 3 bits 0 1 1: 4 bits 1 0 0: 5 bits 1 0 1: 6 bits 1 1 0: 7 bits 1 1 1: 8 bits	R/W*1
b3	BCWP	BC Write Protect	0: Enables a value to be written in the BC[2:0] bits. (This bit is read as 1.)	R/W*1
b6 to b4	CKS[2:0]	Internal Reference Clock Select	Select the internal reference clock (IICφ) source for the RIIC. b6 b4 0 0 0: PCLK/1 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	MTWP	MST/TRS Write Protect	0: Disables writing to the ICCR2.MST and TRS bits. 1: Enables writing to the ICCR2.MST and TRS bits.	R/W

Note 1. Rewrite the BC[2:0] bits and set the BCWP bit to 0 at the same time.

BC[2:0] Bits (Bit Counter)

These bits function as a counter that indicates the number of bits remaining to be transferred at the detection of a rising edge on the SCLn line. Although these bits are writable and readable, it is not necessary to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one (data is transferred with an additional acknowledgment bit) between transferred bytes when the SCLn line is low.

The values of the BC[2:0] bits return to 000b at the end of a data transfer including the acknowledgment bit or when a start condition including a restart condition is detected.

37.2.4 I2C-bus Mode Register 2 (ICMR2)

Address(es): RIIC0.ICMR2 0008 8303h, RIIC1.ICMR2 0008 8323h, RIIC2.ICMR2 0008 8343h

	b7	b6	b5	b4	b3	b2	b1	b0
	DLCS	SDDL[2:0]			—	TMOH	TMOL	TMOS
Value after reset:	0	0	0	0	0	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOS	Timeout Detection Time Select	0: Long mode is selected. 1: Short mode is selected.	R/W
b1	TMOL	Timeout L Count Control	0: Count-up is disabled while the SCLn line is low. 1: Count-up is enabled while the SCLn line is low.	R/W
b2	TMOH	Timeout H Count Control	0: Count-up is disabled while the SCLn line is high. 1: Count-up is enabled while the SCLn line is high.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	SDDL[2:0]	SDA Output Delay Counter	• When ICMR2.DLCS bit is 0 (IICφ) b6 b4 0 0: No output delay 0 1: 1 IICφ cycle 1 0: 2 IICφ cycles 1 1: 3 IICφ cycles 0 0: 4 IICφ cycles 0 1: 5 IICφ cycles 1 0: 6 IICφ cycles 1 1: 7 IICφ cycles • When ICMR2.DLCS bit is 1 (IICφ/2) b6 b4 0 0: No output delay 0 1: 1 or 2 IICφ cycles 1 0: 3 or 4 IICφ cycles 1 1: 5 or 6 IICφ cycles 0 0: 7 or 8 IICφ cycles 0 1: 9 or 10 IICφ cycles 1 0: 11 or 12 IICφ cycles 1 1: 13 or 14 IICφ cycles	R/W
b7	DLCS	SDA Output Delay Clock Source Select	0: The internal reference clock (IICφ) is selected as the clock source of the SDA output delay counter. 1: The internal reference clock divided by 2 (IICφ/2) is selected as the clock source of the SDA output delay counter.*1	R/W

Note 1. The DLCS bit setting of 1 (IICφ/2) only becomes valid when SCL pin is low. When SCL pin is high, the DLCS bit setting of 1 becomes invalid and the clock source becomes the internal reference clock (IICφ).

TMOS Bit (Timeout Detection Time Select)

This bit is used to select long mode or short mode for the timeout detection time when the timeout function is enabled (ICFER.TMOE bit is 1). When this bit is set to 0, long mode is selected. When this bit is set to 1, short mode is selected. In long mode, the timeout detection internal counter functions as a 16-bit counter. In short mode, the counter functions as a 14-bit counter. While the SCLn line is in the state that enables this counter as specified by bits TMOH and TMOL, the counter counts up in synchronization with the internal reference clock (IICφ) as a count source.

For details on the timeout function, refer to section 37.11.1, Timeout Function.

TMOL Bit (Timeout L Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCLn line is held low when the timeout function is enabled (ICFER.TMOE bit is 1).

TMOH Bit (Timeout H Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCLn line is held high when the timeout function is enabled (ICFER.TMOE bit is 1).

SDDL[2:0] Bits (SDA Output Delay Counter)

The SDA output can be delayed by the SDDL[2:0] setting. This counter works with the clock source selected by the DLCS bit. The setting of this function can be used for all types of SDA output, including the transmission of the acknowledgment bit.

Set the SDA output delay time to meet the I2C-bus specification (within the data valid time/data valid acknowledge time*1) or the SMBus specification (more than the data hold time (300 ns) and less than “clock low period – data setup time (250 ns)”). Note that, if a value outside the specification is set, communication with communication devices may malfunction or it may seemingly become a start condition or stop condition depending on the bus state.

For details on this function, refer to section 37.5, SDA Output Delay Function.

Note 1. Data valid time/data valid acknowledge 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+))

37.2.5 I2C-bus Mode Register 3 (ICMR3)

Address(es): RIIC0.ICMR3 0008 8304h, RIIC1.ICMR3 0008 8324h, RIIC2.ICMR3 0008 8344h

	b7	b6	b5	b4	b3	b2	b1	b0
	SMBS	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	NF[1:0]	Noise Filter Stage Select	b1 b0 0 0: Noise of up to one IICφ cycle is filtered out (single-stage filter). 0 1: Noise of up to two IICφ cycles is filtered out (2-stage filter). 1 0: Noise of up to three IICφ cycles is filtered out (3-stage filter). 1 1: Noise of up to four IICφ cycles is filtered out (4-stage filter).	R/W
b2	ACKBR	Received Acknowledge	0: 0 is received as the acknowledgment bit (ACK reception). 1: 1 is received as the acknowledgment bit (NACK reception).	R
b3	ACKBT	Transmit Acknowledge	0: 0 is to be sent as the acknowledgment bit (ACK transmission). 1: 1 is to be sent as the acknowledgment bit (NACK transmission).	R/W*1
b4	ACKWP	ACKBT Write Protect	0: Modification of the ACKBT bit is disabled. 1: Modification of the ACKBT bit is enabled.	R/W*1
b5	RDRFS	RDRF Flag Set Timing Select	0: The RDRF flag is set at the rising edge of the ninth SCL. (The SCLn line is not held low at the falling edge of the eighth clock pulse.) 1: The RDRF flag is set at the rising edge of the eighth SCL. (The SCLn line is held low at the falling edge of the eighth clock pulse.) 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 pulse and first clock pulse is not held low.) 1: WAIT (The period between ninth clock pulse and first clock pulse is held low.) Low-hold is released by reading the ICDRR register.	R/W*2
b7	SMBS	SMBus/I2C-bus Select	0: The I2C-bus is selected. 1: The SMBus is selected.	R/W

Note 1. Write to the ACKBT bit only while the ACKWP bit is already 1. If it is attempted to write 1 to both the ACKWP and ACKBT bits at the same time, the ACKBT bit will not be set to 1.

Note 2. The WAIT and RDRFS bits are valid only in receive mode (invalid in transmit mode).

NF[1:0] Bits (Noise Filter Stage Select)

These bits are used to select the number of stages in the digital noise filter.

For details on the digital noise filter function, refer to section 37.6, Digital Noise Filters.

Note: Set the noise range to be filtered out by the noise filter within a range less than the SCLn line high period or low period. If the noise filter width is set to a value of [the shorter one of either SCL high width or SCL low width] – 1.5 × t_{IICcyc} (cycle time of internal reference clock (IICφ)) or a greater value, the serial clock is regarded as noise by the noise filter function of the RIIC, which may prevent the RIIC from operating normally.

ACKBR Bit (Received Acknowledge)

This bit is used to store the value of the acknowledgment bit received from the receiver in transmit mode.
[Setting condition]

- When 1 is received as the acknowledgment bit with the ICCR2.TRS bit set to 1

[Clearing conditions]

- When 0 is received as the acknowledgment bit with the ICCR2.TRS bit set to 1
- When 1 is written to the ICCR1.IICRST bit while the ICCR1.ICE bit is 0 (RIIC reset)

ACKBT Bit (Transmit Acknowledge)

This bit is used to set the bit to be sent at the acknowledgment 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 generation is detected (when a stop condition is detected with the ICCR2.SP bit set to 1)
- When 1 is written to the ICCR1.IICRST bit while the ICCR1.ICE bit is 0 (RIIC reset)

ACKWP Bit (ACKBT Write Protect)

This bit is used to control the modification of the ACKBT bit.

RDRFS Bit (RDRF Flag Set Timing Select)

This bit is used to select the RDRF flag set timing in receive mode and also to select whether to hold the SCLn line low at the falling edge of the eighth SCL.
When the RDRFS bit is 0, the SCLn line is not held low at the falling edge of the eighth SCL, and the RDRF flag is set to 1 at the rising edge of the ninth SCL.
When the RDRFS bit is 1, the RDRF flag is set to 1 at the rising edge of the eighth SCL and the SCLn line is held low at the falling edge of the eighth SCL. 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 acknowledgment bit is sent. This enables processing to send ACK (ACKBT bit is 0) or NACK (ACKBT bit is 1) according to receive data.

WAIT Bit (WAIT)

This bit is used to control whether to hold the period between the ninth SCL and the first SCL low until the I2C-bus receive data register (ICDRR) is completely read each time single-byte data is received in receive mode.
When the WAIT bit is 0, the receive operation is continued without holding the period between the ninth and the first SCL 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 SCL until the ICDRR register value is read each time single-byte data is received. This enables receive operation in byte units.

Note: When the value of the WAIT bit is to be read, be sure to read the ICDRR register beforehand.

SMBS Bit (SMBus/I2C-bus Select)

Setting this bit to 1 selects the SMBus and enables the ICSER.HOAE bit.

37.2.6 I2C-bus Function Enable Register (ICFER)

Address(es): RIIC0.ICFER 0008 8305h, RIIC1.ICFER 0008 8325h, RIIC2.ICFER 0008 8345h

	b7	b6	b5	b4	b3	b2	b1	b0
	FMPE	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
Value after reset:	0	1	1	1	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOE	Timeout Function Enable	0: The timeout function is disabled. 1: The timeout function is enabled.	R/W
b1	MALE	Master Arbitration-Lost Detection Enable	0: Master arbitration-lost detection is disabled. (Disables the arbitration-lost detection function and does not clear the ICCR2.MST and TRS bits automatically when arbitration is lost.) 1: Master arbitration-lost detection is enabled. (Enables the arbitration-lost detection function and clears the ICCR2.MST and TRS bits automatically when arbitration is lost.)	R/W
b2	NALE	NACK Transmission Arbitration-Lost Detection Enable	0: NACK transmission arbitration-lost detection is disabled. 1: NACK transmission arbitration-lost detection is enabled.	R/W
b3	SALE	Slave Arbitration-Lost Detection Enable	0: Slave arbitration-lost detection is disabled. 1: Slave arbitration-lost detection is enabled.	R/W
b4	NACKE	NACK Reception Transfer Suspension Enable	0: Transfer operation is not suspended during NACK reception (transfer suspension disabled). 1: Transfer operation is suspended during NACK reception (transfer suspension enabled).	R/W
b5	NFE	Digital Noise Filter Enable	0: Digital noise filters are not used. 1: Digital noise filters are used.	R/W
b6	SCLE	SCL Synchronization Enable	0: SCL synchronization is disabled. 1: SCL synchronization is enabled.	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 and RIIC2, bit 7 is reserved.

TMOE Bit (Timeout Function Enable)

This bit is used to enable or disable the timeout function.

For details on the timeout function, refer to section 37.11.1, 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 data transfer when NACK is received 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 value of the received acknowledgment bit.

For details on the NACK reception transfer suspension function, refer to section 37.8.2, NACK Reception Transfer Suspension Function.

SCLE Bit (SCL Synchronization Enable)

This bit is used to specify whether to the SCL output is to be synchronized with the SCL input. Normally, set this bit to 1.

When the SCLE bit is set to 0 (SCL synchronization is disabled), the RIIC does not synchronize the SCL output with the SCL input. In this setting, the RIIC outputs the clock with the transfer rate set in registers ICBRH and ICBRL regardless of the SCLn line state. For this reason, if the load of the I2C-bus line is much larger than the specification value or if the SCL output overlaps in multiple masters, the short-cycle SCL that does not meet the specification may be output. When the SCL synchronization is not used, it also affects the generation of a start condition, restart condition, and stop condition, and the continuous output of additional SCL.

This bit must not be set to 0 except for checking the output of the set transfer rate.

FMPE Bit (Fast-Mode Plus Enable)

This bit is used to specify whether to use a slope control circuit for Fast-mode Plus (Fm+).

When this bit is set to 1, a slope control circuit conforming to the Fast-mode Plus (Fm+) slope control specification (tof) of the I2C-bus is selected. When this bit is set to 0, a slope control circuit conforming to the Standard-mode (Sm) and Fast-mode (Fm) slope control specification (tof) of the I2C-bus is selected.

Set this bit to 1 when using the transmission rate within a range up to 1 Mbps (Fast-mode Plus (Fm+)) of the I2C-bus specification. Set this bit to 0 when using the transmission rate at other rates (up to 100 kbps (Sm), up to 400 kbps (Fm)) or for SMBus (10 to 100 kbps).

37.2.7 I2C-bus Status Enable Register (ICSER)

Address(es): RIIC0.ICSER 0008 8306h, RIIC1.ICSER 0008 8326h, RIIC2.ICSER 0008 8346h

	b7	b6	b5	b4	b3	b2	b1	b0
	HOAE	—	DIDE	—	GCAE	SAR2E	SAR1E	SAR0E
Value after reset:	0	0	0	0	1	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	SAR0E	Slave Address Register 0 Enable	0: Slave address in registers SARL0 and SARU0 is disabled. 1: Slave address in registers SARL0 and SARU0 is enabled.	R/W
b1	SAR1E	Slave Address Register 1 Enable	0: Slave address in registers SARL1 and SARU1 is disabled. 1: Slave address in registers SARL1 and SARU1 is enabled.	R/W
b2	SAR2E	Slave Address Register 2 Enable	0: Slave address in registers SARL2 and SARU2 is disabled. 1: Slave address in registers SARL2 and SARU2 is enabled.	R/W
b3	GCAE	General Call Address Enable	0: General call address detection is disabled. 1: General call address detection is enabled.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	DIDE	Device-ID Address Detection Enable	0: Device-ID address detection is disabled. 1: Device-ID address detection is enabled.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	HOAE	Host Address Enable	0: Host address detection is disabled. 1: Host address detection is enabled.	R/W

SARyE Bit (Slave Address Register y Enable) (y = 0 to 2)

This bit is used to enable or disable the slave address set in registers SARLy and SARUy.

When this bit is set to 1, the slave address set in registers SARLy and SARUy is enabled and is compared with the received slave address.

When this bit is set to 0, the slave address set in registers SARLy and SARUy is disabled and is ignored even if it matches the received slave address.

GCAE Bit (General Call Address Enable)

This bit is used to specify whether to ignore the general call address (0000 000b + 0 (write): All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the RIIC recognizes the received slave address as the general call address independently of the slave addresses set in registers SARLy and SARUy (y = 0 to 2) and performs data receive operation.

When this bit is set to 0, the received slave address is ignored even if it matches the general call address.

DIDE Bit (Device-ID Address Detection Enable)

This bit is used to specify whether to recognize and execute the device-ID address when a device ID (1111 100b) is received in the first byte after a start condition or restart condition is detected.

When this bit is set to 1, if the received first byte matches the device ID, the RIIC recognizes that the device-ID address has been received. When the following R/W# bit is 0 (write), the RIIC recognizes the second and the following bytes as slave addresses and continues the receive operation.

When this bit is set to 0, the RIIC ignores the received first byte even if it matches the device ID address and recognizes the first byte as a normal slave address.

For details on the device-ID address detection, refer to section 37.7.3, Device-ID Address Detection.

HOAE Bit (Host Address Enable)

This bit is used to specify whether to ignore received host address (0001 000b) when the ICMR3.SMBS bit is 1.

When this bit is set to 1 while the ICMR3.SMBS bit is 1, if the received slave address matches the host address, the RIIC recognizes the received slave address as the host address independently of the slave addresses set in registers SARLy and SARUy (y = 0 to 2) and performs the receive operation.

When the ICMR3.SMBS bit or the HOAE bit is set to 0, the received slave address is ignored even if it matches the host address.

37.2.8 I2C-bus Interrupt Enable Register (ICIER)

Address(es): RIIC0.ICIER 0008 8307h, RIIC1.ICIER 0008 8327h, RIIC2.ICIER 0008 8347h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOIE	Timeout Interrupt Request Enable	0: Timeout interrupt (TMOI) request is disabled. 1: Timeout interrupt (TMOI) request is enabled.	R/W
b1	ALIE	Arbitration-Lost Interrupt Request Enable	0: Arbitration-lost interrupt (ALI) request is disabled. 1: Arbitration-lost interrupt (ALI) request is enabled.	R/W
b2	STIE	Start Condition Detection Interrupt Request Enable	0: Start condition detection interrupt (STI) request is disabled. 1: Start condition detection interrupt (STI) request is enabled.	R/W
b3	SPIE	Stop Condition Detection Interrupt Request Enable	0: Stop condition detection interrupt (SPI) request is disabled. 1: Stop condition detection interrupt (SPI) request is enabled.	R/W
b4	NAKIE	NACK Reception Interrupt Request Enable	0: NACK reception interrupt (NAKI) request is disabled. 1: NACK reception interrupt (NAKI) request is enabled.	R/W
b5	RIE	Receive Data Full Interrupt Request Enable	0: Receive data full interrupt (RXI) request is disabled. 1: Receive data full interrupt (RXI) request is enabled.	R/W
b6	TEIE	Transmission End Interrupt Request Enable	0: Transmission end interrupt (TEI) request is disabled. 1: Transmission end interrupt (TEI) request is enabled.	R/W
b7	TIE	Transmit Data Empty Interrupt Request Enable	0: Transmit data empty interrupt (TXI) request is disabled. 1: Transmit data empty interrupt (TXI) request is enabled.	R/W

TMOIE Bit (Timeout Interrupt Request Enable)

This bit is used to enable or disable timeout interrupt (TMOI) requests when the ICSR2.TMOF flag is set to 1. A TMOI interrupt request is canceled by setting the TMOF flag or the TMOIE bit to 0.

ALIE Bit (Arbitration-Lost Interrupt Request Enable)

This bit is used to enable or disable arbitration-lost interrupt (ALI) requests when the ICSR2.AL flag is set to 1. An ALI interrupt request is canceled by setting the AL flag or the ALIE bit to 0.

STIE Bit (Start Condition Detection Interrupt Request Enable)

This bit is used to enable or disable start condition detection interrupt (STI) requests when the ICSR2.START flag is set to 1. An STI interrupt request is canceled by setting the START flag or the STIE bit to 0.

SPIE Bit (Stop Condition Detection Interrupt Request Enable)

This bit is used to enable or disable stop condition detection interrupt (SPI) requests when the ICSR2.STOP flag is set to 1. An SPI interrupt request is canceled by setting the STOP flag or the SPIE bit to 0.

NAKIE Bit (NACK Reception Interrupt Request Enable)

This bit is used to enable or disable NACK reception interrupt (NAKI) requests when the ICSR2.NACKF flag is set to 1. An NAKI interrupt request is canceled by setting the NACKF flag or the NAKIE bit to 0.

RIE Bit (Receive Data Full Interrupt Request Enable)

This bit is used to enable or disable receive data full interrupt (RXI) requests when the ICSR2.RDRF flag is set to 1.

TEIE Bit (Transmission End Interrupt Request Enable)

This bit is used to enable or disable transmission end interrupt (TEI) requests when the ICSR2.TEND flag is set to 1. An TEI interrupt request is canceled by setting the TEND flag or the TEIE bit to 0.

TIE Bit (Transmit Data Empty Interrupt Request Enable)

This bit is used to enable or disable transmit data empty interrupt (TXI) requests when the ICSR2.TDRE flag is set to 1.

37.2.9 I2C-bus Status Register 1 (ICSR1)

Address(es): RIIC0.ICSR1 0008 8308h, RIIC1.ICSR1 0008 8328h, RIIC2.ICSR1 0008 8348h

	b7	b6	b5	b4	b3	b2	b1	b0
	HOA	—	DID	—	GCA	AAS2	AAS1	AAS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	AAS0	Slave Address 0 Detection Flag	0: Slave address 0 is not detected. 1: Slave address 0 is detected.	R/(W) *1
b1	AAS1	Slave Address 1 Detection Flag	0: Slave address 1 is not detected. 1: Slave address 1 is detected.	R/(W) *1
b2	AAS2	Slave Address 2 Detection Flag	0: Slave address 2 is not detected. 1: Slave address 2 is detected.	R/(W) *1
b3	GCA	General Call Address Detection Flag	0: General call address is not detected. 1: General call address is detected.	R/(W) *1
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	DID	Device-ID Address Detection Flag	0: Device-ID command is not detected. 1: Device-ID command is detected. • This bit is set to 1 when the first byte received immediately after a start condition is detected matches a value of (device ID (1111 100b) + 0 (write)).	R/(W) *1
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	HOA	Host Address Detection Flag	0: Host address is not detected. 1: Host address is detected. • This bit is set to 1 when the received slave address matches the host address (0001 000b).	R/(W) *1

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

AASy Flag (Slave Address y Detection Flag) (y = 0 to 2)

[Setting conditions]

For 7-bit address format: SARUy.FS bit = 0

- When the received slave address matches the SARLy.SVA[6:0] bits value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL in the first byte.

For 10-bit address format: SARUy.FS bit = 1

- When the received slave address matches a value of (11110b + SARUy.SVA[1:0] bits) and the following address matches the SARLy value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL in the second byte.

[Clearing conditions]

- When 0 is written to the AASy flag after reading the AASy flag to be 1
- When a stop condition is detected
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

For 7-bit address format: SARUy.FS bit = 0

- When the received slave address does not match the SARLy.SVA[6:0] bits value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 0 at the rising edge of the ninth SCL in the first byte.

For 10-bit address format: SARUy.FS bit = 1

- When the received slave address does not match a value of (11110b + SARUy.SVA[1:0] bits) with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 0 at the rising edge of the ninth SCL in the first byte.
- When the received slave address matches a value of (11110b + SARUy.SVA[1:0] bits) and the following address does not match the SARLy value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 0 at the rising edge of the ninth SCL in the second byte.

GCA Flag (General Call Address Detection Flag)

[Setting condition]

- When the received slave address matches the general call address (0000 000b + 0 (write)) with the ICSER.GCAE bit set to 1 (general call address detection is enabled)
  This flag is set to 1 at the rising edge of the ninth SCL in the first byte.

[Clearing conditions]

- When 0 is written to the GCA flag after reading GCA flag to be 1
- When a stop condition is detected
- When the received slave address does not match the general call address (0000 000b + 0 (write)) with the ICSER.GCAE bit set to 1 (general call address detection is enabled)
  This flag is set to 0 at the rising edge of the ninth SCL in the first byte.
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

DID Flag (Device-ID Address Detection Flag)

[Setting condition]

- When the first byte received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100b) + 0 (write)) with the ICSER.DIDE bit set to 1 (device-ID address detection is enabled)
  This flag is set to 1 at the rising edge of the ninth SCL in the first byte.

[Clearing conditions]

- When 0 is written to the DID flag after reading DID flag to be 1
- When a stop condition is detected
- When the first byte received immediately after a start condition or restart condition is detected does not match a value of (device ID (1111 100b)) with the ICSER.DIDE bit set to 1 (device-ID address detection is enabled)
  This flag is set to 0 at the rising edge of the ninth SCL in the first byte.
- When the first byte received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100b) + 0 (write)) and the second byte does not match any of slave addresses 0 to 2 with the ICSER.DIDE bit set to 1 (device-ID address detection is enabled)
  This flag is set to 0 at the rising edge of the ninth SCL in the second byte.
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

HOA Flag (Host Address Detection Flag)

[Setting condition]

- When the received slave address matches the host address (0001 000b) with the ICSER.HOAE bit set to 1 (host address detection is enabled)
  This flag is set to 1 at the rising edge of the ninth SCL in the first byte.

[Clearing conditions]

- When 0 is written to the HOA flag after reading HOA flag to be 1
- When a stop condition is detected
- When the received slave address does not match the host address (0001 000b) with the ICSER.HOAE bit set to 1

(host address detection is enabled)

This flag is set to 0 at the rising edge of the ninth SCL in the first byte.

- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

37.2.10 I2C-bus Status Register 2 (ICSR2)

Address(es): RIIC0.ICSR2 0008 8309h, RIIC1.ICSR2 0008 8329h, RIIC2.ICSR2 0008 8349h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOF	Timeout Detection Flag	0: Timeout is not detected. 1: Timeout is detected.	R/(W) *1
b1	AL	Arbitration-Lost Flag	0: Arbitration is not lost. 1: Arbitration is lost.	R/(W) *1
b2	START	Start Condition Detection Flag	0: Start condition is not detected. 1: Start condition is detected.	R/(W) *1
b3	STOP	Stop Condition Detection Flag	0: Stop condition is not detected. 1: Stop condition is detected.	R/(W) *1
b4	NACKF	NACK Detection Flag	0: NACK is not detected. 1: NACK is detected.	R/(W) *1
b5	RDRF	Receive Data Full Flag	0: The ICDRR register contains no receive data. 1: The ICDRR register contains receive data.	R/(W) *1
b6	TEND	Transmission End Flag	0: Data is being transmitted. 1: Data has been transmitted.	R/(W) *1
b7	TDRE	Transmit Data Empty Flag	0: The ICDRT register contains transmit data. 1: The ICDRT register contains no transmit data.	R

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

TMOF Flag (Timeout Detection Flag)

This flag is set to 1 when the RIIC recognizes timeout after the SCLn line state remains unchanged for a certain period.
[Setting condition]

- When the SCLn line state remains unchanged for the period specified by bits ICMR2.TMOH, TMOL, and TMOS while the ICFER.TMOE bit is 1 (the timeout function is enabled) in master mode or in slave mode and the received slave address matches.

[Clearing conditions]

- When 0 is written to the TMOF bit after reading TMOF = 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

AL Flag (Arbitration-Lost Flag)

This flag shows that bus mastership has been lost (loss in arbitration) due to a bus conflict or some other reason when a start condition is generated or an address and data are transmitted. The RIIC monitors the level on the SDAn line during transmission and, if the level on the line does not match the value of the bit being output, sets the value of the AL flag to 1 to indicate that the bus is occupied by another device.
The RIIC can also detect loss of arbitration during NACK transmission in receive 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 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 high (the SDAn pin is in the high-impedance state))
- When a start condition is detected while the ICCR2.ST bit is 1 (requests to generate a start condition) or the internal SDA output state does not match the SDAn line level
- When the ICCR2.ST bit is set to 1 (requests to generate a start condition) with the ICCR2.BBSY flag set to 1.

When NACK arbitration-lost detection is enabled: ICFER.NALE = 1

- When the internal SDA output state does not match the SDAn line level at the rising edge of SCL in the ACK period during NACK transmission in receive mode

When slave arbitration-lost detection is enabled: ICFER.SALE = 1

- When the internal SDA output state does not match the SDAn line level at the rising edge of SCL except for the ACK period during data transmission in slave transmit mode

[Clearing conditions]

- When 0 is written to the AL flag after reading AL = 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Table 37.4 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 generation error	When internal SDA output state does not match SDAn line level when a start condition is detected while the ICCR2.ST bit is 1
					When ICCR2.ST bit is set to 1 with ICCR2.BBSY flag set to 1
			1	Transmit data mismatch	When transmit data (including slave address) does not match the bus state in master transmit mode
×	1	×	1	NACK transmission mismatch	When ACK is detected during transmission of NACK in master receive mode or slave receive mode
×	×	1	1	Transmit data mismatch	When transmit data does not match the bus state in slave transmit mode

×: Don't care

START Flag (Start Condition Detection Flag)

[Setting condition]

- When a start condition (or a restart condition) is detected

[Clearing conditions]

- When 0 is written to the START bit after reading START = 1
- When a stop condition is detected
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

STOP Flag (Stop Condition Detection Flag)

[Setting condition]

- When a stop condition is detected

[Clearing conditions]

- When 0 is written to the STOP bit after reading STOP = 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

NACKF Flag (NACK Detection Flag)

[Setting condition]

- When ACK is not received (NACK is received) from the receiver in transmit mode with the ICFER.NACKE bit set to 1 (transfer suspension enabled)

[Clearing conditions]

- When 0 is written to the NACKF bit after reading NACKF = 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: When the NACKF flag is set to 1, the RIIC suspends data transmission/reception. Writing to the ICDRT register in transmit mode or reading from the ICDRR register in receive mode with the NACKF flag set to 1 does not enable data transmit/receive operation. To restart data transmission/reception, set the NACKF flag to 0.

RDRF Flag (Receive Data Full Flag)

[Setting conditions]

- When receive data has been transferred from the ICDRS register to the ICDRR register
  This flag is set to 1 at the rising edge of the eighth or ninth SCL (selected by the ICMR3.RDRFS bit)
- When the received slave address matches after a start condition (or a restart condition) is detected with the ICCR2.TRS bit set to 0

[Clearing conditions]

- When 0 is written to the RDRF bit after reading RDRF = 1
- When data is read from the ICDRR register
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

TEND Flag (Transmission End Flag)

[Setting condition]

- At the rising edge of the ninth SCL while the TDRE flag is 1

[Clearing conditions]

- When 0 is written to the TEND bit after reading TEND = 1
- When data is written to the ICDRT register
- When a stop condition is detected
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

TDRE Flag (Transmit Data Empty Flag)

[Setting conditions]

- When data has been transferred from the ICDRT register to the ICDRS register and the ICDRT register becomes empty
- When the ICCR2.TRS bit is set to 1
- When the received slave address matches while the TRS bit is 1

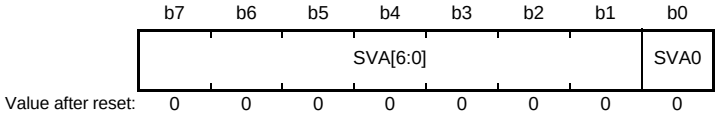
[Clearing conditions]

- When data is written to the ICDRT register
- When the ICCR2.TRS bit is set to 0
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: The NACKF flag becoming 1 while the ICFER.NACKE bit is 1 suspends data transmission and reception by the RIIC. Even if the next data for transmission has already been written to the ICDRT register (the TDRE flag is 0), the data in the ICDRT register is retained but not transferred to the ICDRS register. At this point, the TDRE flag does not become 1.

37.2.11 Slave Address Register Ly (SARLy) (y = 0 to 2)

Address(es): RIIC0.SARL0 0008 830Ah, RIIC1.SARL0 0008 832Ah, RIIC2.SARL0 0008 834Ah, RIIC0.SARL1 0008 830Ch, RIIC1.SARL1 0008 832Ch, RIIC2.SARL1 0008 834Ch, RIIC0.SARL2 0008 830Eh, RIIC1.SARL2 0008 832Eh, RIIC2.SARL2 0008 834Eh



Bit	Symbol	Bit Name	Description	R/W
b0	SVA0	10-Bit Address LSB	Set a slave address	R/W
b7 to b1	SVA[6:0]	7-Bit Address/10-Bit Address Lower Bits	Set a slave address	R/W

SVA0 Bit (10-Bit Address LSB)

When the 10-bit address format is selected (SARUy.FS bit is 1), this bit functions as the LSB of a 10-bit address and forms the lower 8 bits of a 10-bit address in combination with the SVA[6:0] bits.

When the ICSER.SARyE bit is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, this bit is valid. While the SARUy.FS bit or SARyE bit is 0, the setting of this bit is ignored.

SVA[6:0] Bits (7-Bit Address/10-Bit Address Lower Bits)

When the 7-bit address format is selected (SARUy.FS bit is 0), these bits function as a 7-bit address. When the 10-bit address format is selected (SARUy.FS bit is 1), these bits function as the lower 8 bits of a 10-bit address in combination with the SVA0 bit.

While the ICSER.SARyE bit is 0, the setting of these bits is ignored.

37.2.12 Slave Address Register Uy (SARUy) (y = 0 to 2)

Address(es): RIIC0.SARU0 0008 830Bh, RIIC1.SARU0 0008 832Bh, RIIC2.SARU0 0008 834Bh, RIIC0.SARU1 0008 830Dh, RIIC1.SARU1 0008 832Dh, RIIC2.SARU1 0008 834Dh, RIIC0.SARU2 0008 830Fh, RIIC1.SARU2 0008 832Fh, RIIC2.SARU2 0008 834Fh

	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 Select	0: The 7-bit address format is selected. 1: The 10-bit address format is selected.	R/W
b2, b1	SVA[1:0]	10-Bit Address Upper Bits	Set a 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 Select)

This bit is used to select 7-bit address or 10-bit address for slave address y (in registers SARLy and SARUy).

When the ICSER.SARyE bit is set to 1 (registers SARLy and SARUy enabled) and the SARUy.FS bit is 0, the 7-bit address format is selected for slave address y, the SARLy.SVA[6:0] bits setting is valid, and the settings of the SVA[1:0] bits and the SARLy.SVA0 bit are ignored.

When the ICSER.SARyE bit is set to 1 (registers SARLy and SARUy enabled) and the SARUy.FS bit is 1, the 10-bit address format is selected for slave address y and the settings of the SVA[1:0] bits and SARLy are valid.

While the ICSER.SARyE bit is 0 (registers SARLy and SARUy disabled), the setting of the SARUy.FS bit is invalid.

SVA[1:0] Bits (10-Bit Address Upper Bits)

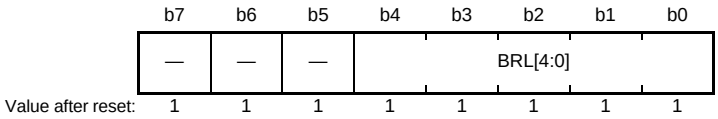
When the 10-bit address format is selected (FS = 1), these bits function as the upper 2 bits of a 10-bit address.

When the ICSER.SARyE bit is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, these bits are valid.

While the SARUy.FS bit or SARyE bit is 0, the setting of these bits is ignored.

37.2.13 I2C-bus Bit Rate Low-Level Register (ICBRL)

Address(es): RIIC0.ICBRL 0008 8310h, RIIC1.ICBRL 0008 8330h, RIIC2.ICBRL 0008 8350h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	BRL[4:0]	Bit Rate Low Period	Low period of SCL	R/W
b7 to b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

ICBRL is a 5-bit register to set the low period of SCL.
It also works to generate the data setup time for automatic SCL low-hold operation (refer to section 37.8, Automatic Low-Hold Function for SCL); when the RIIC is used only in slave mode, this register needs to be set to a value longer than the data setup time*1.
ICBRL counts the low period with the internal reference clock (IICφ) specified by the ICMR1.CKS[2:0] bits.
If the digital noise filter is enabled (the ICFER.NFE bit is 1), set the ICBRL register to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICMR3.NF[1:0] bits.

Note 1. Data setup time (tSU: DAT)
250 ns (up to 100 kbps: Standard-mode (Sm))
100 ns (up to 400 kbps: Fast-mode (Fm))
50 ns (up to 1 Mbps: Fast-mode plus (Fm+))

37.2.14 I2C-bus Bit Rate High-Level Register (ICBRH)

Address(es): RIIC0.ICBRH 0008 8311h, RIIC1.ICBRH 0008 8331h, RIIC2.ICBRH 0008 8351h

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	BRH[4:0]	Bit Rate High Period	High period of SCL	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 period of SCL. ICBRH is valid in master mode. If the RIIC is used only in slave mode, this register need not to set the high period.
ICBRH counts the high period with the internal reference clock (IICφ) specified by the ICMR1.CKS[2:0] bits.
If the digital noise filter is enabled (the ICFER.NFE bit is 1), set the ICBRH register to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICMR3.NF[1:0] bits.

The I2C transfer rate and the SCL duty are calculated using the following expression.
Transfer rate = 1 / {[(ICBRH + 1) + (ICBRL + 1)] / IICφ*1 + SCLn line rise time [tr] + SCLn line fall time [tf]}
Duty cycle = {SCLn line rise time [tr]*2 + (ICBRH + 1) / IICφ} / {SCLn line fall time [tf]*2 + (ICBRL + 1) / IICφ}

Note 1. IICφ = PCLK × Division ratio
Note 2. The SCLn line rise time [tr] and SCLn line fall time [tf] depend on the total bus line capacitance [Cb] and the pull-up resistor [Rp]. For details, refer to the I2C-bus specification from NXP Semiconductors.

Table 37.5 lists examples of ICBRH/ICBRL settings.

Table 37.5 Examples of ICBRH/ICBRL Settings for Transfer Rate

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	8			10			12.5		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	100b	22 (F6h)	25 (F9h)	101b	13 (EDh)	15 (EFh)	101b	16 (F0h)	20 (F4h)
50	010b	16 (F0h)	19 (F3h)	010b	21 (F5h)	24 (F8h)	011b	12 (ECh)	15 (EFh)
100	001b	15 (EFh)	18 (F2h)	001b	19 (F3h)	23 (F7h)	001b	24 (F8h)	29 (FDh)
400	000b	4 (E4h)	10 (EAh)	000b	5 (E5h)	12 (ECh)	000b	7 (E7h)	16 (F0h)
1000	000b	2 (E2h)	3 (E3h)	000b	2 (E2h)	4 (E4h)	000b	3 (E3h)	6 (E6h)

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	16			20			25		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	101b	22 (F6h)	25 (F9h)	110b	13 (EDh)	15 (EFh)	110b	16 (F0h)	20 (F4h)
50	011b	16 (F0h)	19 (F3h)	011b	21 (F5h)	24 (F8h)	100b	12 (ECh)	15 (EFh)
100	010b	15 (EFh)	18 (F2h)	010b	19 (F3h)	23 (F7h)	010b	24 (F8h)	29 (FDh)
400	000b	9 (E9h)	20 (F4h)	000b	11 (EBh)	25 (F9h)	001b	7 (E7h)	16 (F0h)
1000	000b	4 (E4h)	7 (E7h)	000b	5 (E5h)	9 (E9h)	000b	6 (E6h)	12 (ECh)

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	30			32			33		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	110b	20 (F4h)	24 (F8h)	110b	22 (F6h)	25 (F9h)	110b	22 (F6h)	26 (FAh)
50	100b	15 (EFh)	18 (F2h)	100b	16 (F0h)	19 (F3h)	100b	17 (F1h)	20 (F4h)
100	011b	14 (EEh)	17 (F1h)	011b	15 (EFh)	18 (F2h)	011b	16 (F0h)	19 (F3h)
400	001b	8 (E8h)	19 (F3h)	001b	9 (E9h)	20 (F4h)	001b	9 (E9h)	21 (F5h)
1000	000b	7 (E7h)	14 (EEh)	000b	8 (E8h)	16 (F0h)	000b	8 (E8h)	16 (F0h)

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	40			50			60		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	111b	13 (EDh)	15 (EFh)	111b	16 (F0h)	20 (F4h)	111b	20 (F4h)	24 (F8h)
50	100b	21 (F5h)	24 (F8h)	100b	26 (FAh)	31 (FFh)	101b	15 (EFh)	18 (F2h)
100	011b	19 (F3h)	23 (F7h)	011b	24 (F8h)	29 (FDh)	100b	14 (EEh)	17 (F1h)
400	001b	11 (EBh)	25 (F9h)	010b	7 (E7h)	16 (F0h)	010b	8 (E8h)	19 (F3h)
1000	000b	10 (EAh)	20 (F4h)	000b	12 (ECh)	24 (F8h)	000b	15 (EFh)	29 (FDh)

Note: ICBRH/ICBRL settings in these tables are calculated using the following values:
SCLn line rise 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 fall time (tf): 400 kbps or less (Sm/Fm): 300 ns, 1 Mbps or less (Fm+): 120 ns
For the specified values of rise time (tr) and fall time (tf) of the SCLn signal, refer to the I2C-bus specification from NXP Semiconductors.

37.2.15 I2C-bus Transmit Data Register (ICDRT)

Address(es): RIIC0.ICDRT 0008 8312h, RIIC1.ICDRT 0008 8332h, RIIC2.ICDRT 0008 8352h

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

When the ICDRT register detects a space in the I2C-bus shift register (ICDRS), it transfers the transmit data that has been written to the ICDRT register to the ICDRS register and starts transmitting data in transmit mode.
The double-buffer structure of the ICDRT register and the ICDRS register allows continuous transmit operation if the next transmit data has been written to the ICDRT register while the ICDRS register data is being transmitted.
The ICDRT register can always be read and written. Write transmit data to the ICDRT register once when a transmit data empty interrupt (TXI) request is generated.

37.2.16 I2C-bus Receive Data Register (ICDRR)

Address(es): RIIC0.ICDRR 0008 8313h, RIIC1.ICDRR 0008 8333h, RIIC2.ICDRR 0008 8353h

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

When 1 byte of data has been received, the received data is transferred from the I2C-bus shift register (ICDRS) to the ICDRR register to enable the next data to be received.
The double-buffer structure of the ICDRS register and the ICDRR register allows continuous receive operation if the received data has been read from the ICDRR register while the ICDRS register is receiving data.
The ICDRR register cannot be written. Read data from the ICDRR register once when a receive data full interrupt (RXI) request is generated.
If the ICDRR register receives the next receive data before the current data is read from the ICDRR register (while the ICSR2.RDRF flag is 1), the RIIC automatically holds the SCL line low one cycle before the RDRF flag is set to 1 next.

37.2.17 I2C-bus Shift Register (ICDRS)

The ICDRS register is an 8-bit shift register to transmit and receive data.
During transmission, transmit data is transferred from the ICDRT register to the ICDRS register and is sent from the SDAn pin. During reception, data is transferred from the ICDRS register to the ICDRR register after 1 byte of data has been received.
The ICDRS register cannot be accessed directly.

37.3 Operation

37.3.1 Communication Data Format

The I2C-bus format consists of 8-bit data and 1-bit acknowledgment. The first byte following a start condition or restart condition is an address byte used to specify a slave device with which the master device communicates. The specified slave is valid until a new slave is specified or a stop condition is generated.
Figure 37.2 shows the I2C-bus format, and Figure 37.3 shows the I2C-bus timing.

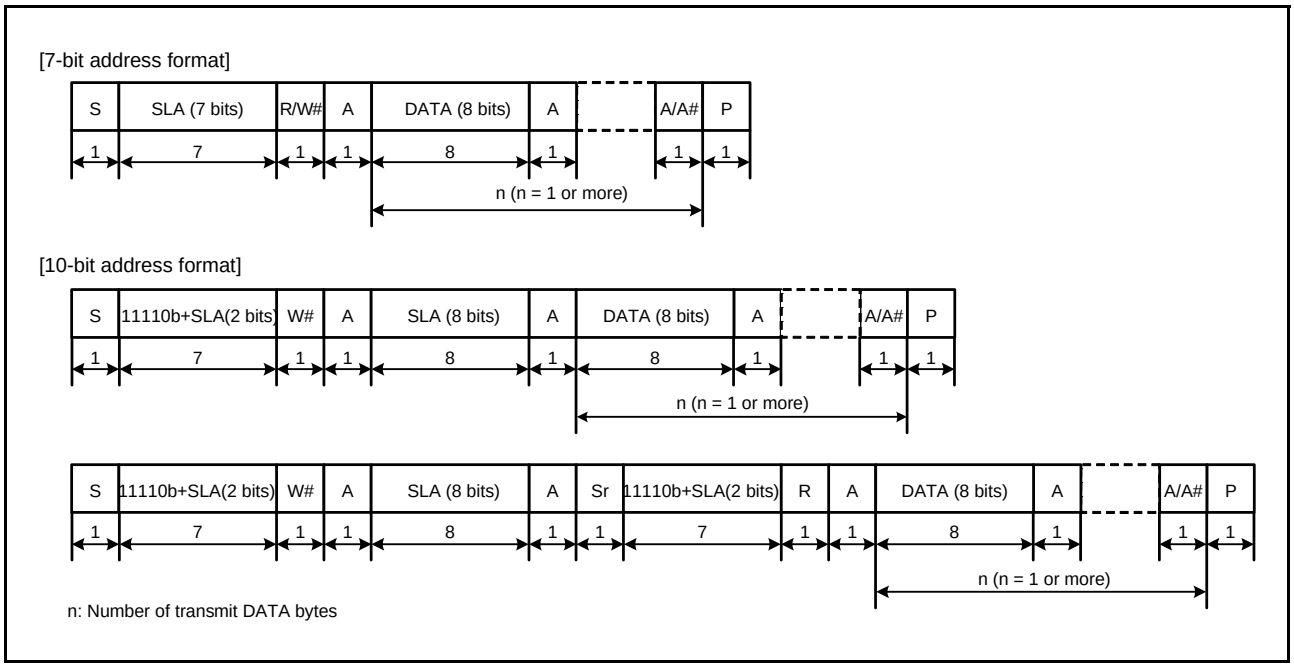


Figure 37.2 I2C-bus Format

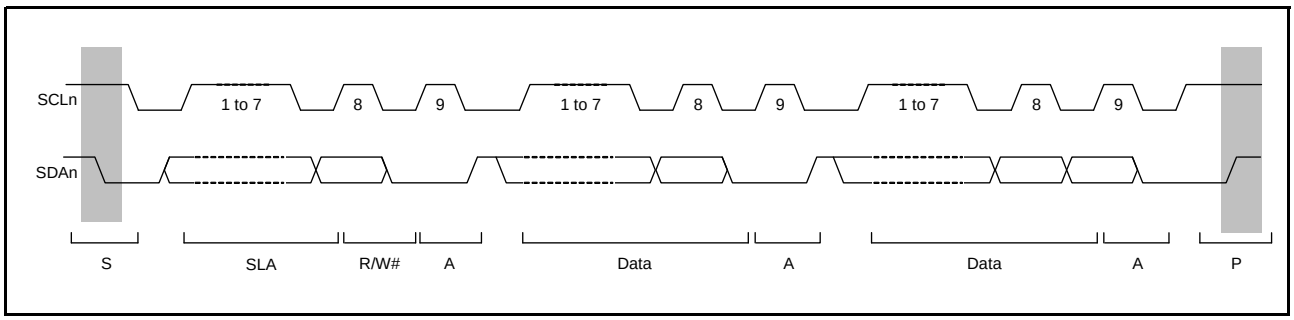


Figure 37.3 I2C-bus Timing (SLA = 7 Bits)

S:	Start condition. The master device drives the SDAn line low from high while the SCLn line is high.
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 receiver drives the SDAn line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)
A#:	Not Acknowledge. The receiver drives the SDAn line high.
Sr:	Restart condition. The master device drives the SDAn line low from high after the setup time has elapsed with the SCLn line high.
DATA:	Transmitted or received data
P:	Stop condition. The master device drives the SDAn line high from low while the SCLn line is high.

37.3.2 Initial Settings

Before starting data transmission and reception, initialize the RIIC according to the procedure in Figure 37.4.
Set the ICCR1.ICE bit to 1 (internal reset) after setting the ICCR1.IICRST bit to 1 (RIIC reset) with the ICCR1.ICE bit set to 0 (SCLn and SDAn pins in inactive state). This initializes the various flags and internal state of the ICSR1 register. After that, set registers SARLy, SARUy, ICSER, ICMR1, ICBRH, and ICBRL (y = 0 to 2), and set the other registers as necessary (for initial settings of the RIIC, see Figure 37.4). When the necessary register settings have been completed, set the ICCR1.IICRST bit to 0 (releases the RIIC reset). This step is not necessary if initialization of the RIIC has already been completed.

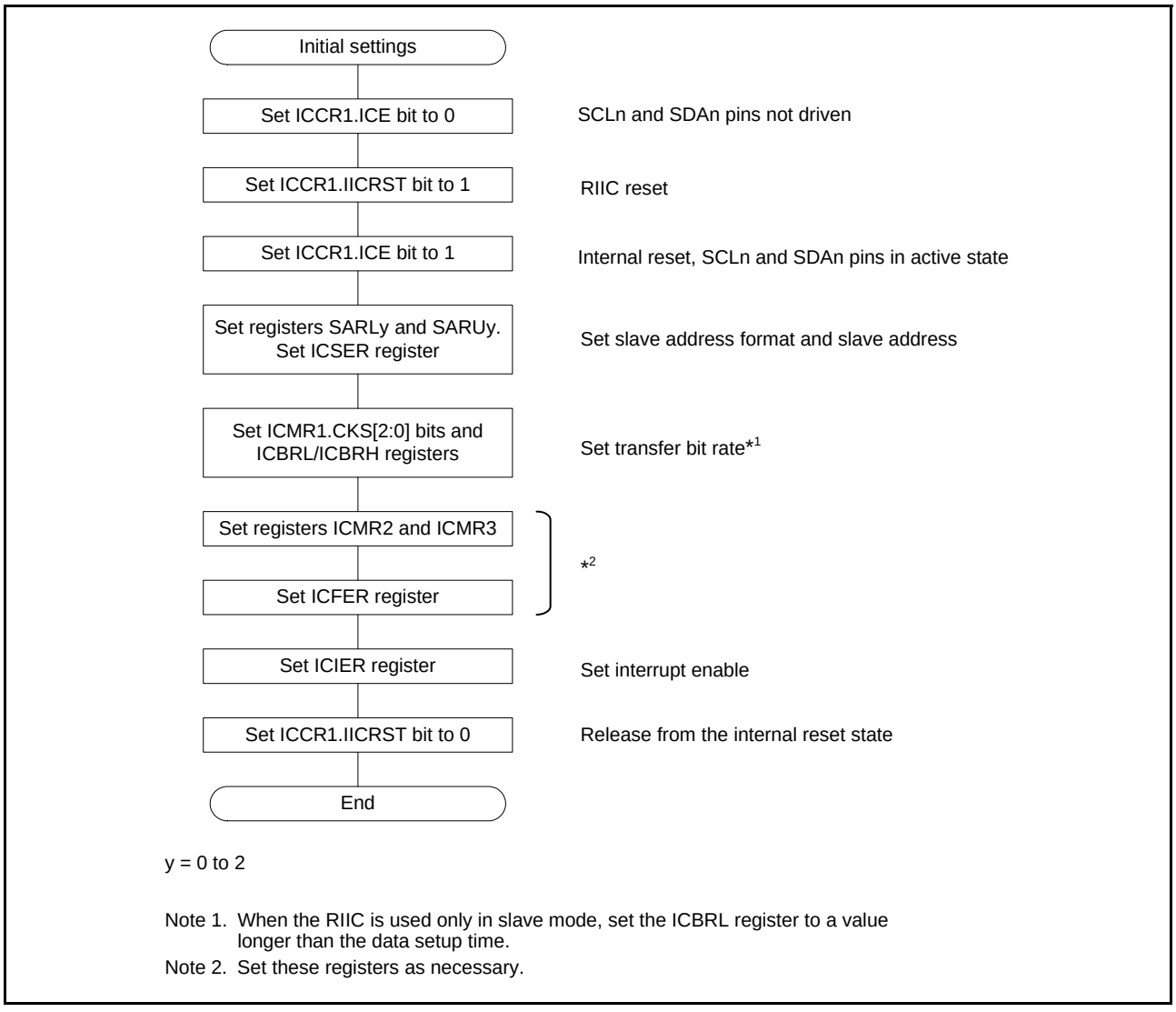


Figure 37.4 Example of RIIC Initialization Flowchart

37.3.3 Master Transmit Operation

In master transmit operation, the RIIC generates clock signals and sends data as the master device, and the slave device returns acknowledgments. Figure 37.5 shows an example of usage of master transmission and Figure 37.6 to Figure 37.8 show the timing of operations in master transmission.
The following describes the procedure and operations for master transmission.

(1) Initial settings. For details, refer to section 37.3.2, Initial Settings.
(2) Read the ICCR2.BBSY flag to check that the bus is open, and then set the ICCR2.ST bit to 1 (requests to generate a start condition). Upon receiving the request, the RIIC generates a start condition. At the same time, the BBSY flag and the ICSR2.START flag are automatically set to 1 and the ST bit is automatically set to 0. At this time, if the start condition is detected and the internal levels for the SDA output state and the levels on the SDAn line have matched while the ST bit is 1, the RIIC recognizes that generating of the start condition as requested by the ST bit has been successfully completed, and bits MST and TRS in the ICCR2 register are automatically set to 1, placing the RIIC in master transmit mode. The ICSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS bit to 1.
(3) Check that the ICSR2.TDRE flag is 1, and then write the value for transmission (the slave address and the R/W# bit) to the ICDRT register. Once the data for transmission are written to the ICDRT register, the TDRE flag is automatically set to 0, the data are transferred from the ICDRT register to the ICDRS register, and the TDRE flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRS bit is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the RIIC continues in master transmit mode.
Because the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to generate a stop condition.
For data transmission with an address in the 10-bit format, start by writing 1111 0b, the 2 higher-order bits of the slave address, and W to the ICDRT register as the first address transmission. Then, as the second address transmission, write the 8 lower-order bits of the slave address to the ICDRT register.
(4) After confirming that the ICSR2.TDRE flag is 1, write the data for transmission to the ICDRT register. The RIIC automatically holds the SCLn line low until the data for transmission are ready or a stop condition is generated.
(5) After all bytes of data for transmission have been written to the ICDRT register, wait until the ICSR2.NACKF or ICSR2.TEND flag becomes 1, and then set the ICCR2.SP bit to 1 (requests to generate a stop condition). Upon receiving a stop condition generation request, the RIIC generates the stop condition.
(6) Upon detecting the stop condition, the RIIC automatically sets bits MST and TRS in the ICCR2 register to 00b and enters slave receive mode. Furthermore, it automatically sets the TDRE and TEND flags to 0, and sets the ICSR2.STOP flag to 1.
(7) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

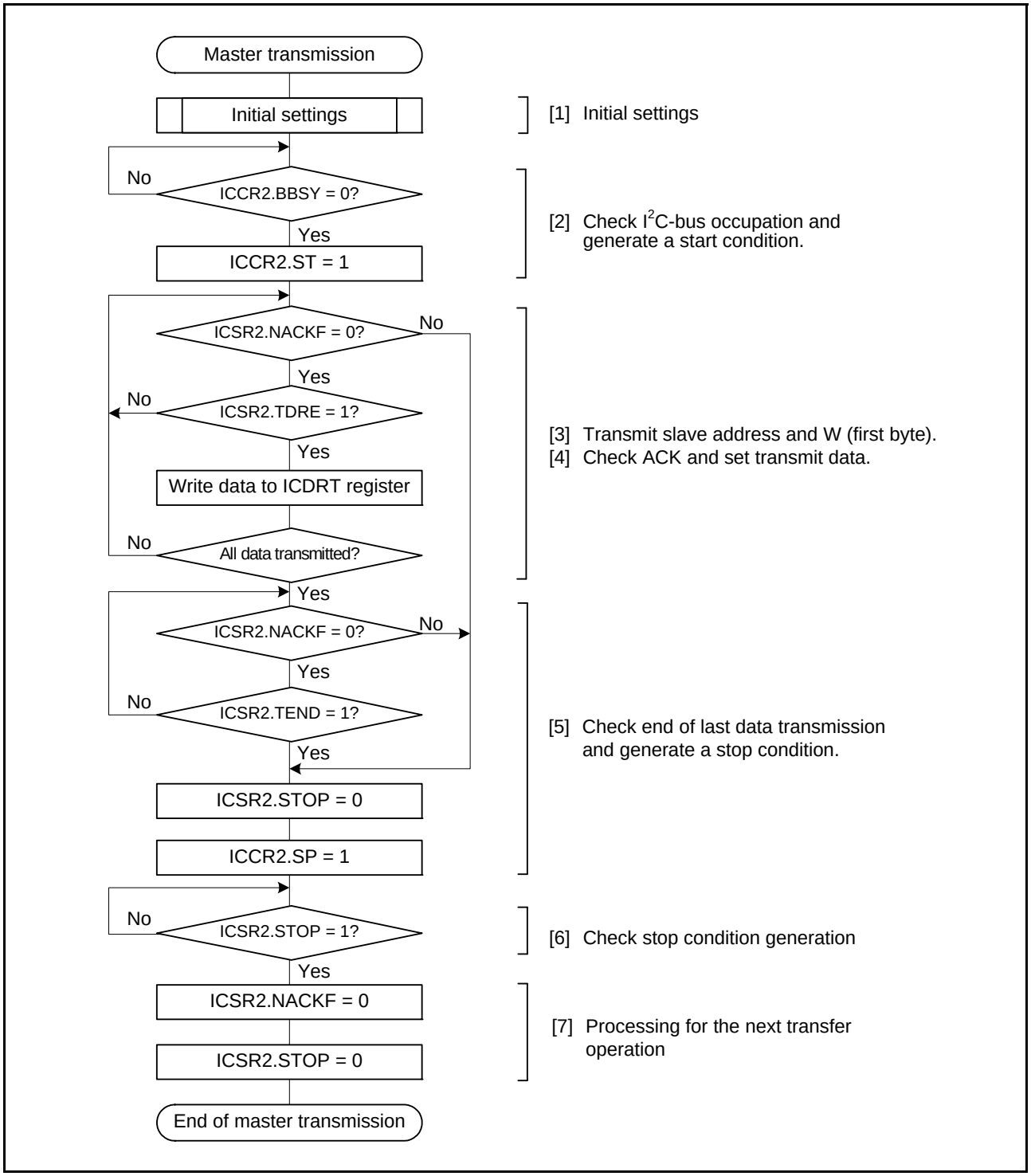


Figure 37.5 Example of Master Transmission Flowchart

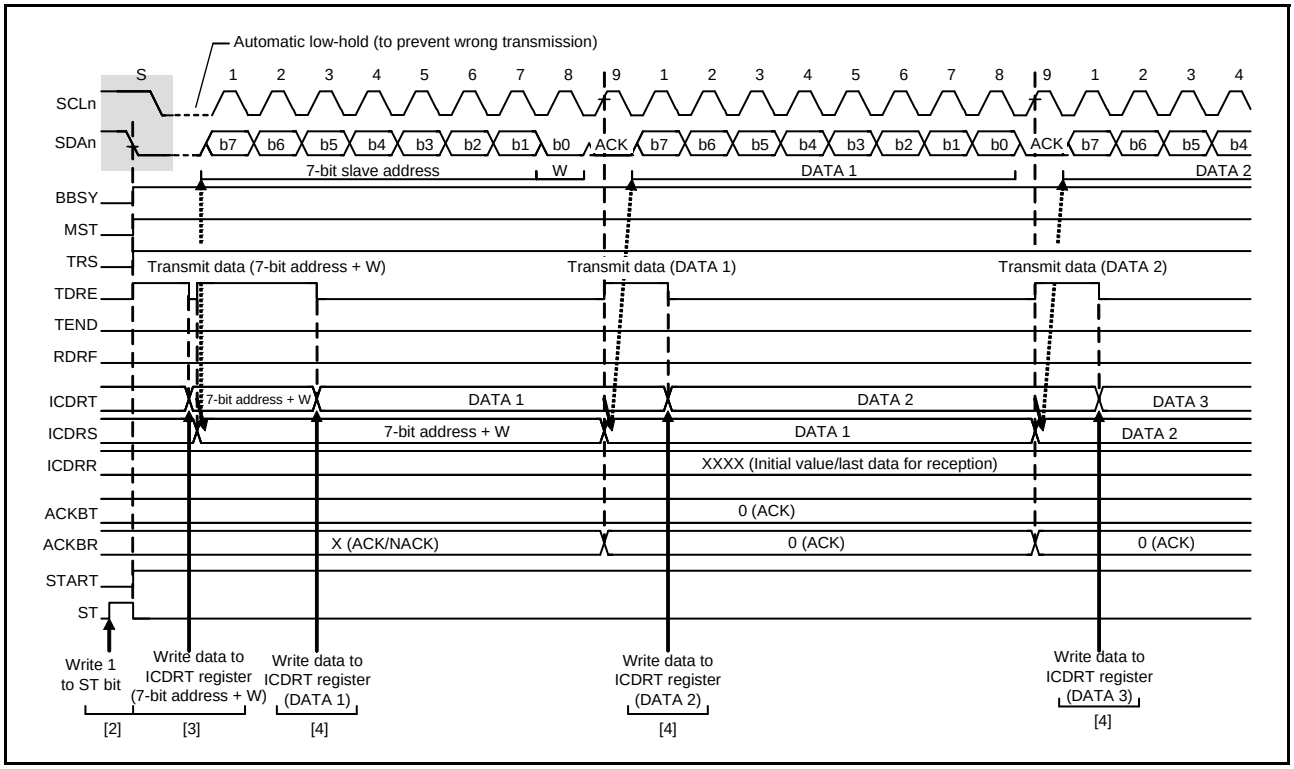


Figure 37.6 Master Transmit Operation Timing (1) (7-Bit Address Format)

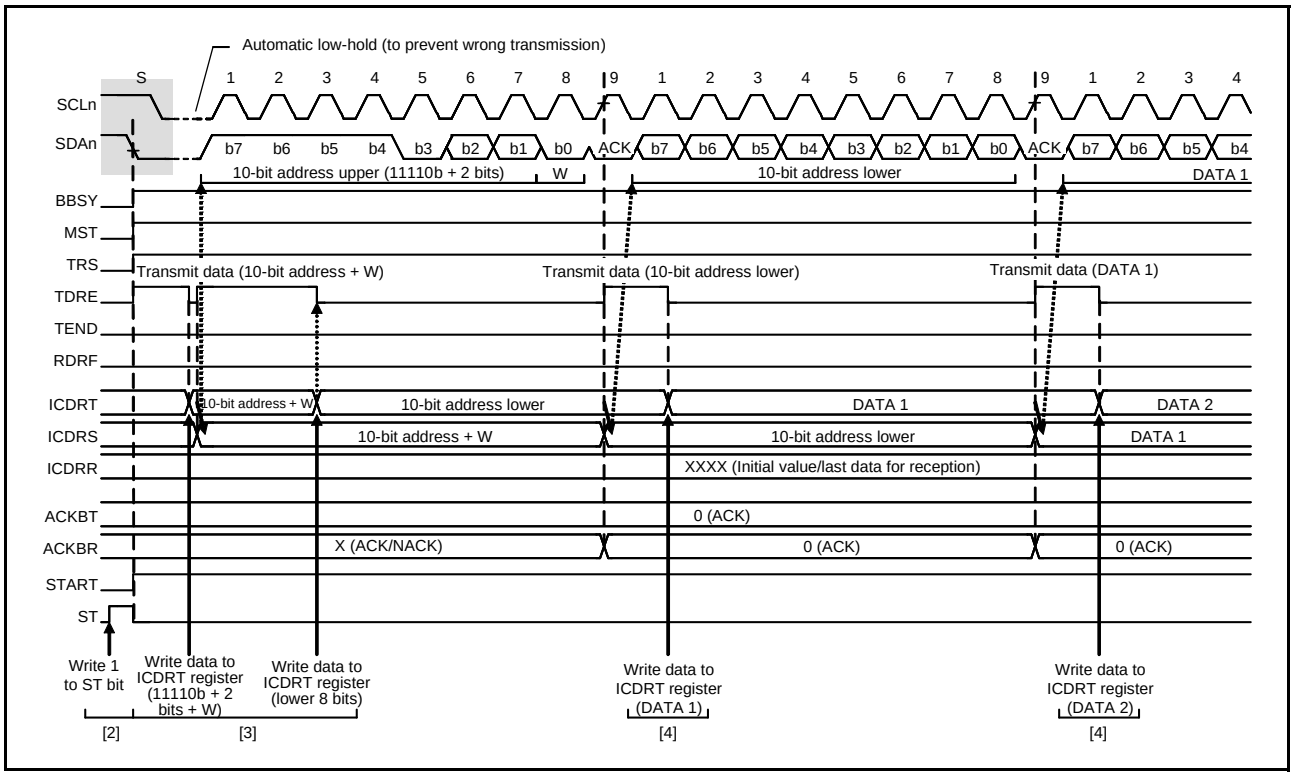


Figure 37.7 Master Transmit Operation Timing (2) (10-Bit Address Format)

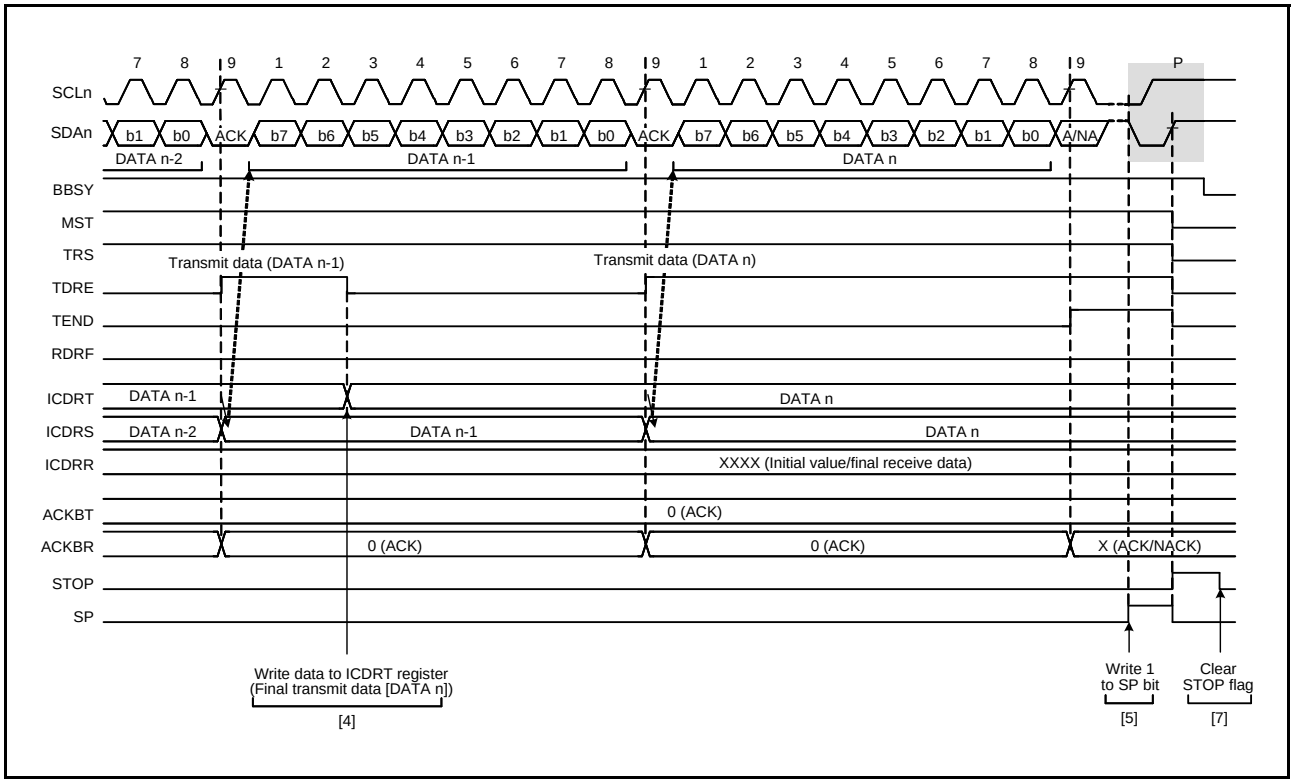


Figure 37.8 Master Transmit Operation Timing (3)

37.3.4 Master Receive Operation

In master receive operation, the RIIC as a master device generates clock signals, receives data from the slave device, and returns acknowledgments. Because the RIIC must start by sending a slave address to the corresponding slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.
Figure 37.9 and Figure 37.10 show examples of usage of master reception (7-bit address format) and Figure 37.11 to Figure 37.13 show the timing of operations in master reception.
The following describes the procedure and operations for master reception.

(1) Initial settings. For details, refer to section 37.3.2, Initial Settings.

(2) Read the ICCR2.BBSY flag to check that the bus is open, and then set the ICCR2.ST bit to 1 (requests to generate a start condition). Upon receiving the request, the RIIC generates a start condition. When the RIIC detects the start condition, the BBSY flag and the ICSR2.START flag are automatically set to 1 and the ST bit is automatically set to 0. At this time, if the start condition is detected and the levels for the SDA output and the levels on the SDAn line have matched while the ST bit is 1, the RIIC recognizes that generating of the start condition as requested by the ST bit has been successfully completed, and bits MST and TRS in the ICCR2 register are automatically set to 1, placing the RIIC in master transmit mode. The ICSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS bit to 1.

(3) Check that the ICSR2.TDRE flag is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to the ICDRT register. Once the data for transmission are written to the ICDRT register, the TDRE flag is automatically set to 0, the data are transferred from the ICDRT register to the ICDRS register, and the TDRE flag is again set to 1. Once the byte containing the slave address and R/W# bit has been transmitted, the value of the ICCR2.TRS bit is automatically updated to select transmit or receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 1, the TRS bit is set to 0 on the rising edge of the ninth SCL, placing the RIIC in master receive mode. At this time, the TDRE flag is set to 0 and the ICSR2.RDRF flag is automatically set to 1.

Because the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to generate a stop condition.
For master reception from a device with a 10-bit address, start by using master transmission to send the 10-bit address, and then generate a restart condition. After that, transmitting 1111 0b, the two higher-order bits of the slave address, and the R bit places the RIIC in master receive mode.

(4) Dummy read the ICDRR register after confirming that the ICSR2.RDRF flag is 1; this makes the RIIC start output of the SCL and start data reception.

(5) After 1 byte of data has been received, the ICSR2.RDRF flag is set to 1 on the rising edge of the eighth or ninth SCL (the clock signal) as selected by the ICMR3.RDRFS bit. Reading the ICDRR register at this time will produce the received data, and the RDRF flag is automatically set to 0 at the same time. Furthermore, the value of the acknowledgment bit received during the ninth SCL is returned as the value set in the ICMR3.ACKBT bit. Furthermore, if the next byte to be received is the next to last byte, set the ICMR3.WAIT bit to 1 (for wait insertion) before reading the ICDRR register (containing the second byte from last). As well as enabling NACK output even in the case of delays in processing to set the ICMR3.ACKBT bit to 1 (NACK) in step (6), due to other interrupts, etc., this fixes the SCLn line to low on the falling edge of the ninth clock pulse in reception of the last byte, so the state is such that generating a stop condition is possible.

(6) When the ICMR3.RDRFS bit is 0 and the slave device must be notified that it is to end transfer for data reception after transfer of the next (final) byte, set the ICMR3.ACKBT bit to 1 (NACK).

(7) After reading the byte before last from the ICDRR register, if the value of the ICSR2.RDRF flag is confirmed to be 1, write 1 to the ICCR2.SP bit (requests to generate a stop condition) and then read the last byte from the ICDRR register. When the ICDRR register is read, the RIIC is released from the wait state and generates the stop condition after low-level output in the ninth clock pulse is completed or the SCLn line is released from the low-hold state.

(8) Upon detecting the stop condition, the RIIC automatically sets bits MST and TRS in the ICCR2 register to 00b and enters slave receive mode. Furthermore, detection of the stop condition leads to setting of the ICSR2.STOP flag to 1.

(9) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

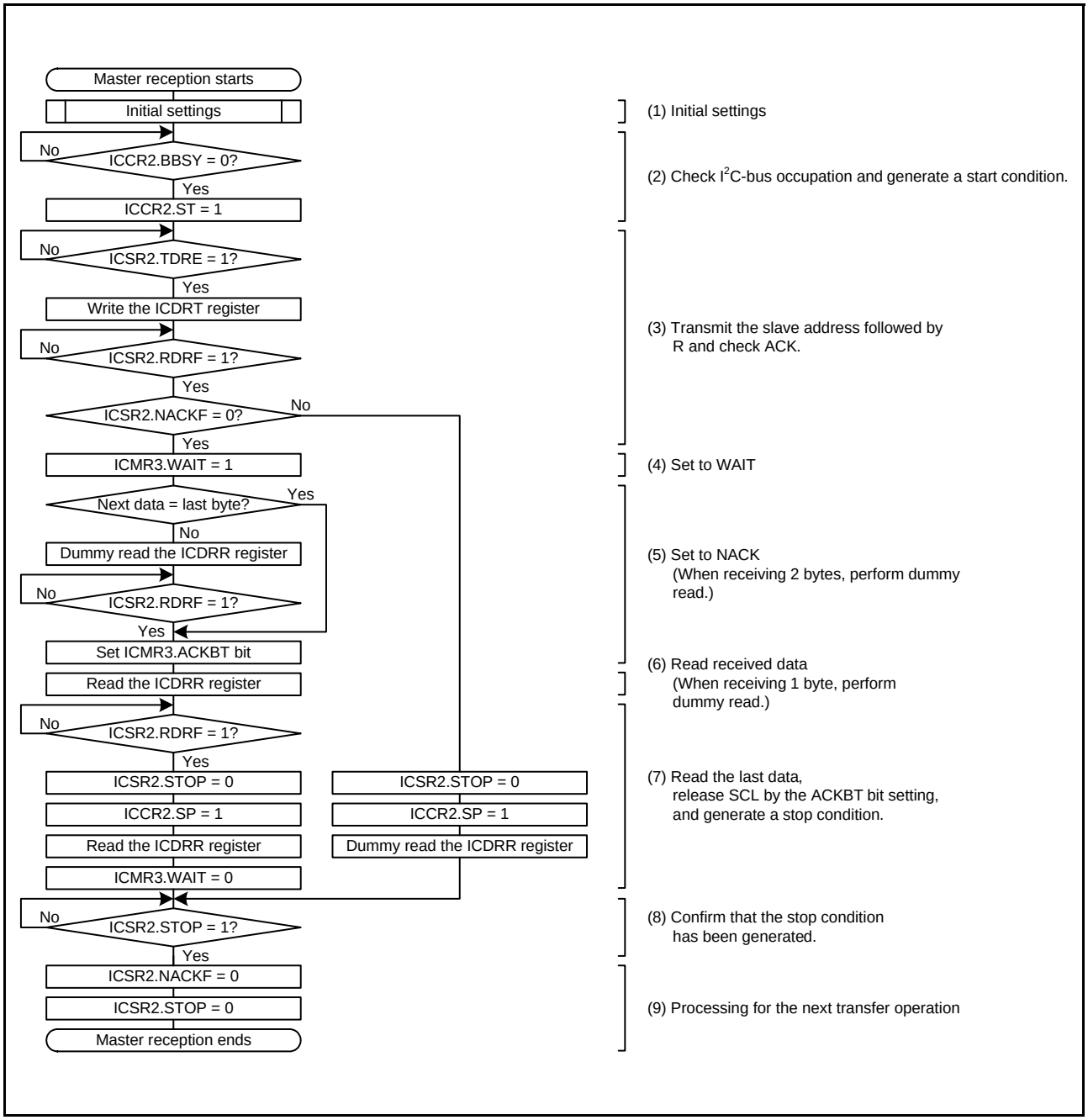


Figure 37.9 Example of Master Reception (7-Bit Address Format, 1 or 2 bytes)

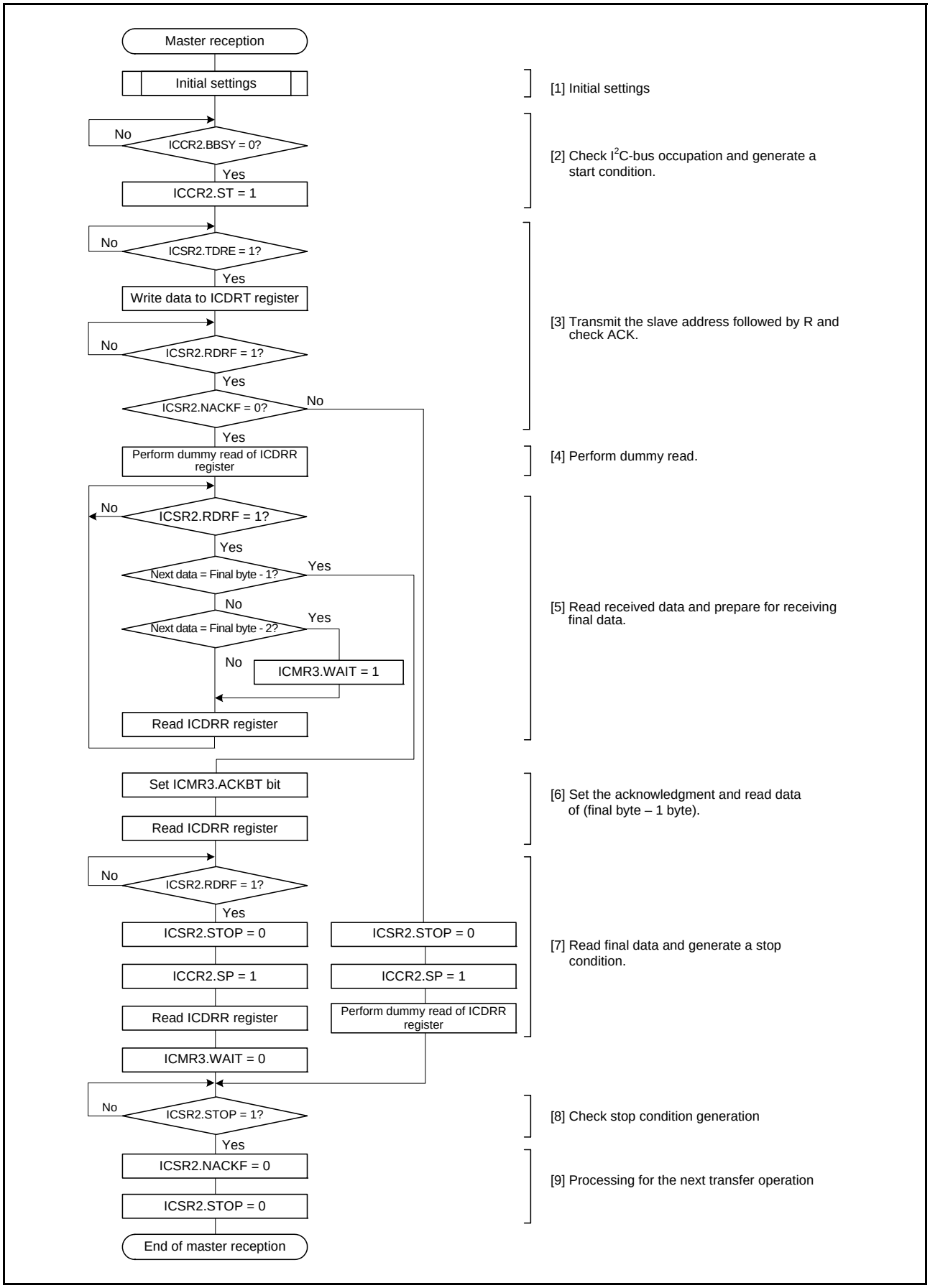


Figure 37.10 Example of Master Reception (7-Bit Address Format, 3 Bytes or More)

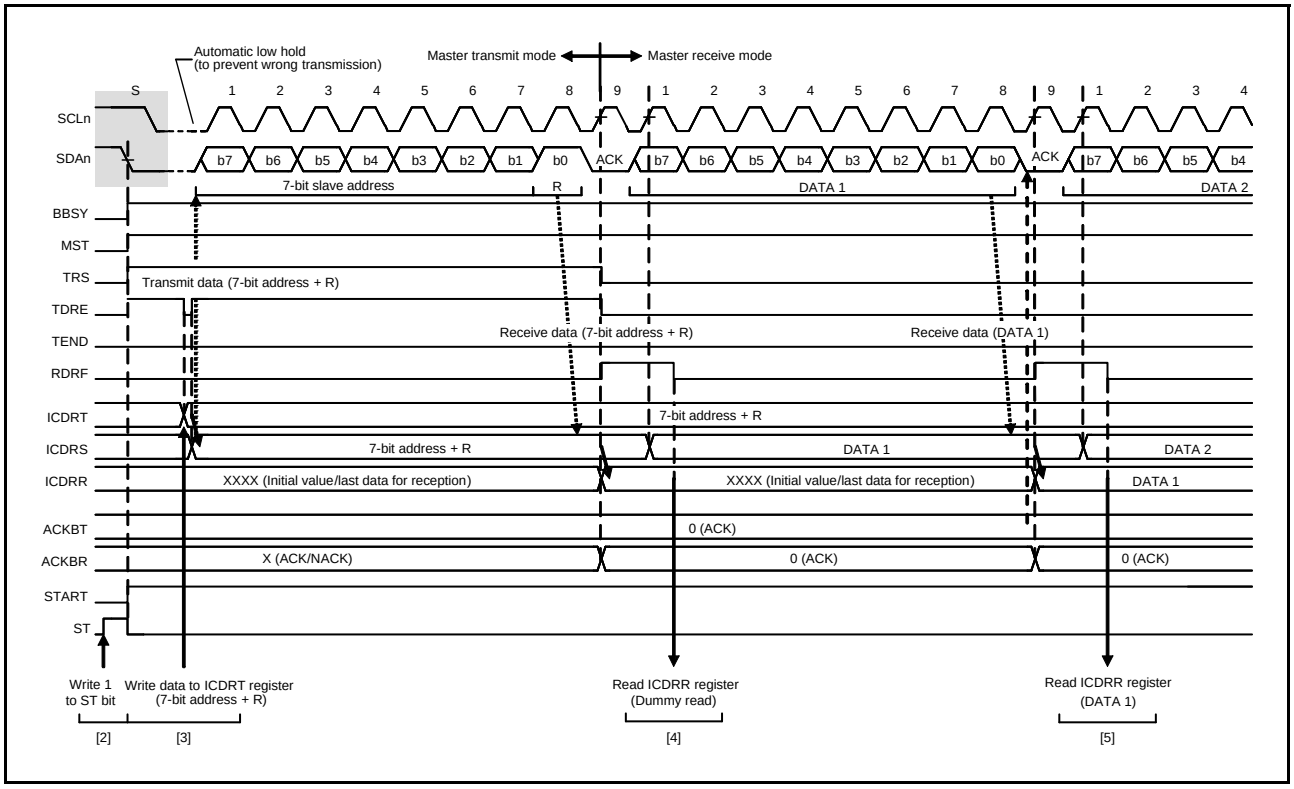


Figure 37.11 Master Receive Operation Timing (1) (7-Bit Address Format, When RDRFS bit is 0)

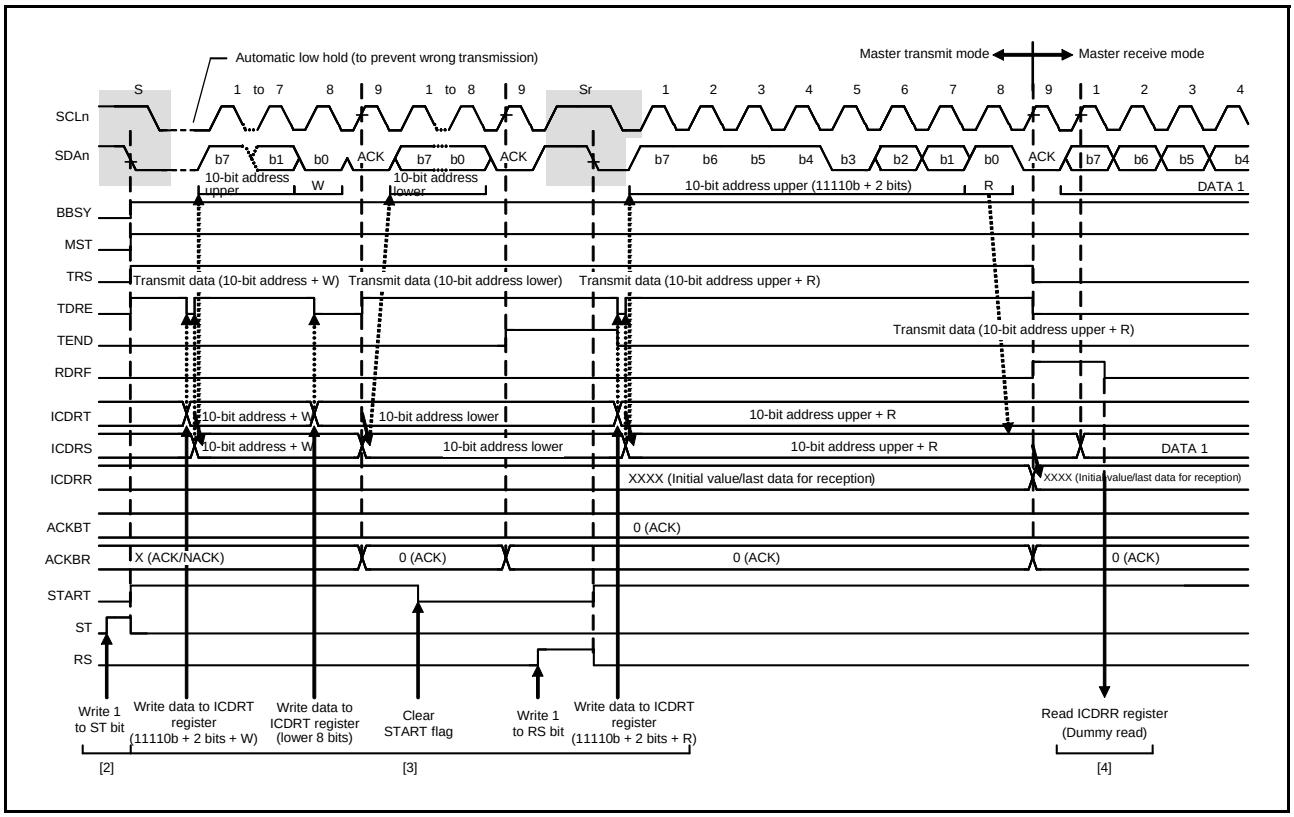


Figure 37.12 Master Receive Operation Timing (2) (10-Bit Address Format, When RDRFS bit is 0)

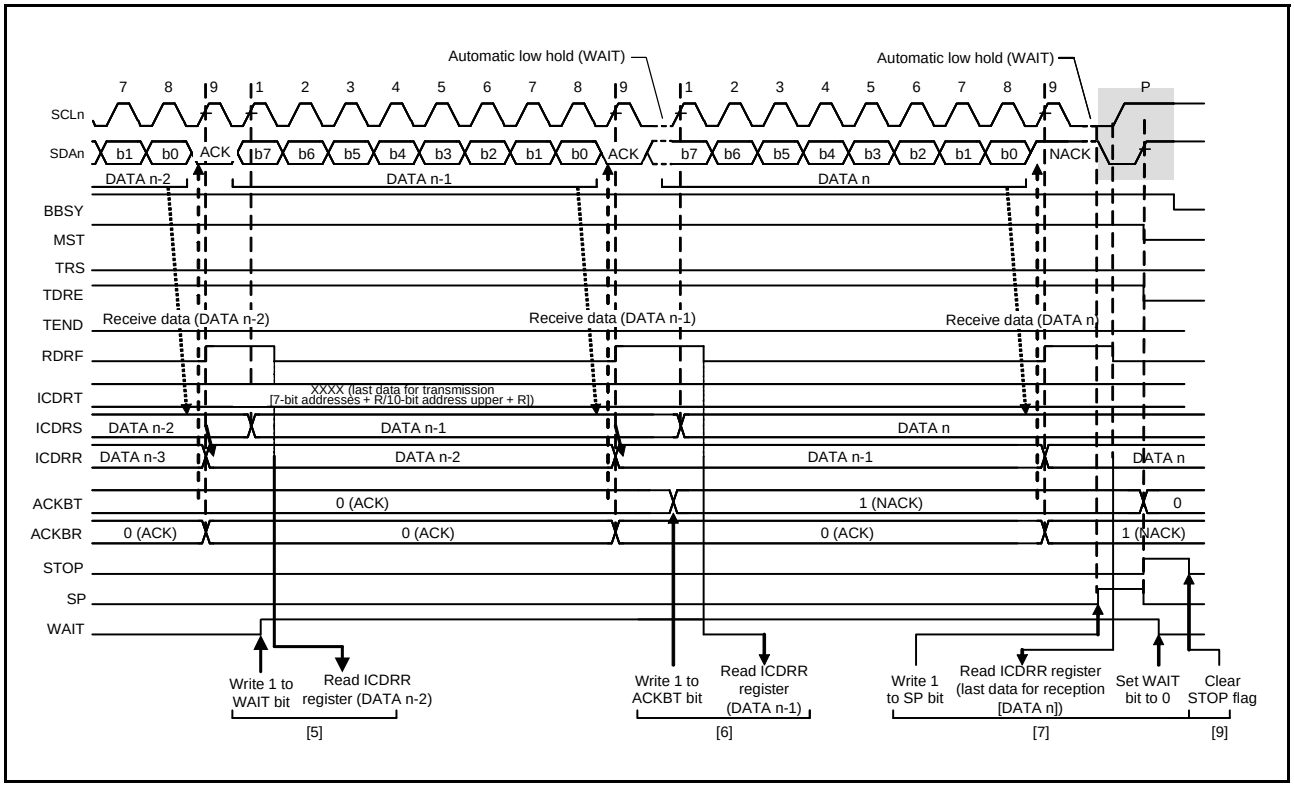


Figure 37.13 Master Receive Operation Timing (3) (When RDRFS bit is 0)

37.3.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL, the RIIC transmits data as a slave device, and the master device returns acknowledgments.

Figure 37.14 shows an example of usage of slave transmission and Figure 37.15 and Figure 37.16 show the timing of operations in slave transmission.

The following describes the procedure and operations for slave transmission.

(1) Initial settings. For details, refer to section 37.3.2, Initial Settings.
After initial settings, the RIIC will stay in the standby state until it receives a slave address that it matches.

(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits ICSR1.HOA, GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth SCL (the clock signal) and outputs the value set in the ICMR3.ACKBT bit to the acknowledgment bit on the ninth SCL. If the value of the R/W# bit that was also received at this time is 1, the RIIC automatically places itself in slave transmit mode by setting both the ICCR2.TRS bit and the ICSR2.TDRE flag to 1.

(3) After the ICSR2.TDRE flag is confirmed to be 1, write the data for transmission to the ICDRT register. At this time, if the RIIC does not receive ACK from the master device (receives an NACK signal) while the ICFER.NACKE bit is 1, the RIIC suspends transfer of the next data.

(4) Wait until the ICSR2.TEND flag is set to 1 while the ICSR2.TDRE flag is 1, after the ICSR2.NACKF flag is set to 1 or the last byte for transmission is written to the ICDRT register. When the ICSR2.NACKF flag or the TEND flag is 1, the RIIC drives the SCLn line low on the falling edge of ninth SCL.

(5) When the ICSR2.NACKF flag or the ICSR2.TEND flag is 1, dummy read the ICDRR register to complete the processing. This releases the SCLn line.

(6) Upon detecting the stop condition, the RIIC automatically sets bits ICSR1.HOA, GCA, and AASy (y = 0 to 2), flags ICSR2.TDRE and TEND, and the ICCR2.TRS bit to 0, and enters slave receive mode.

(7) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

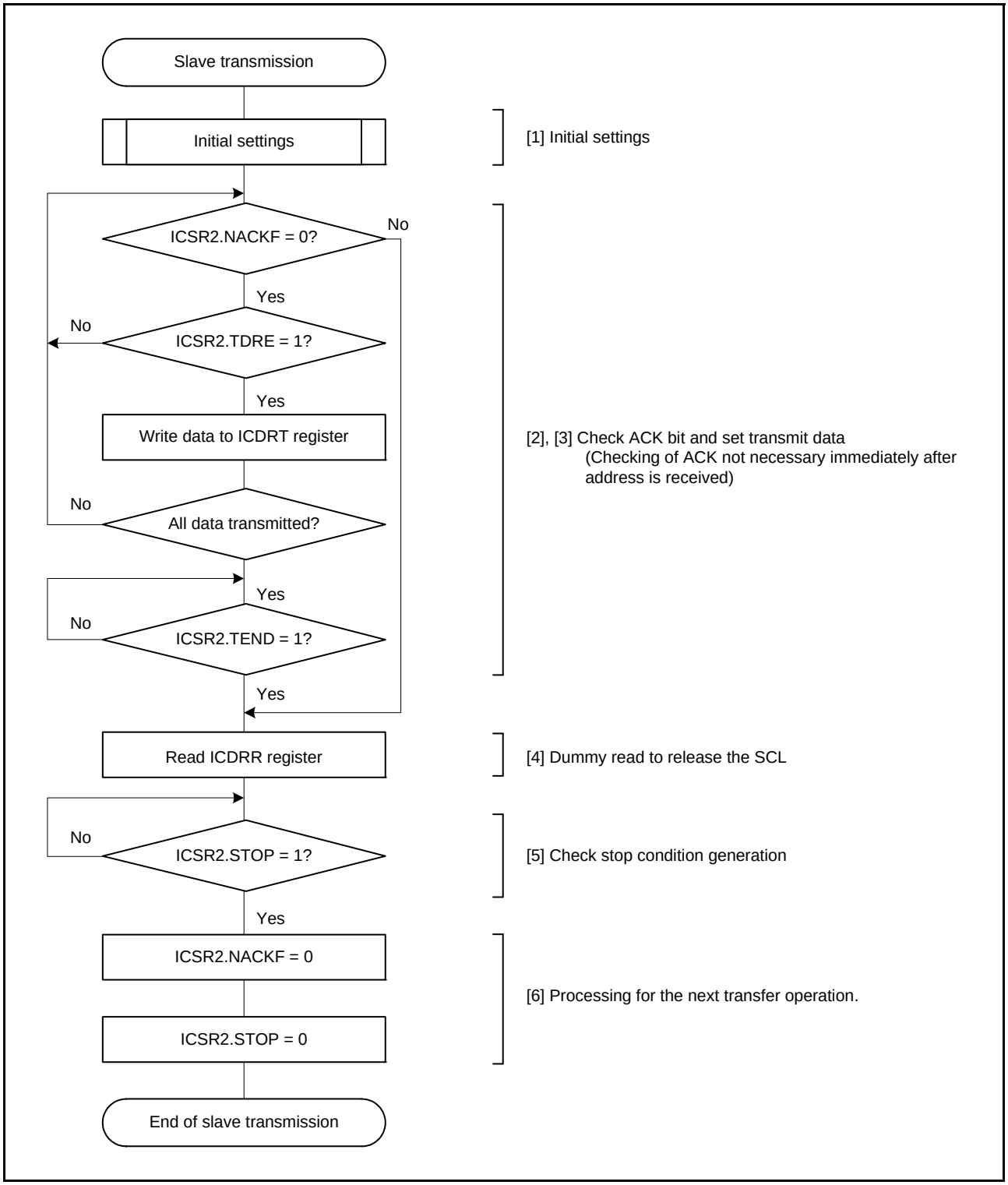


Figure 37.14 Example of Slave Transmission Flowchart

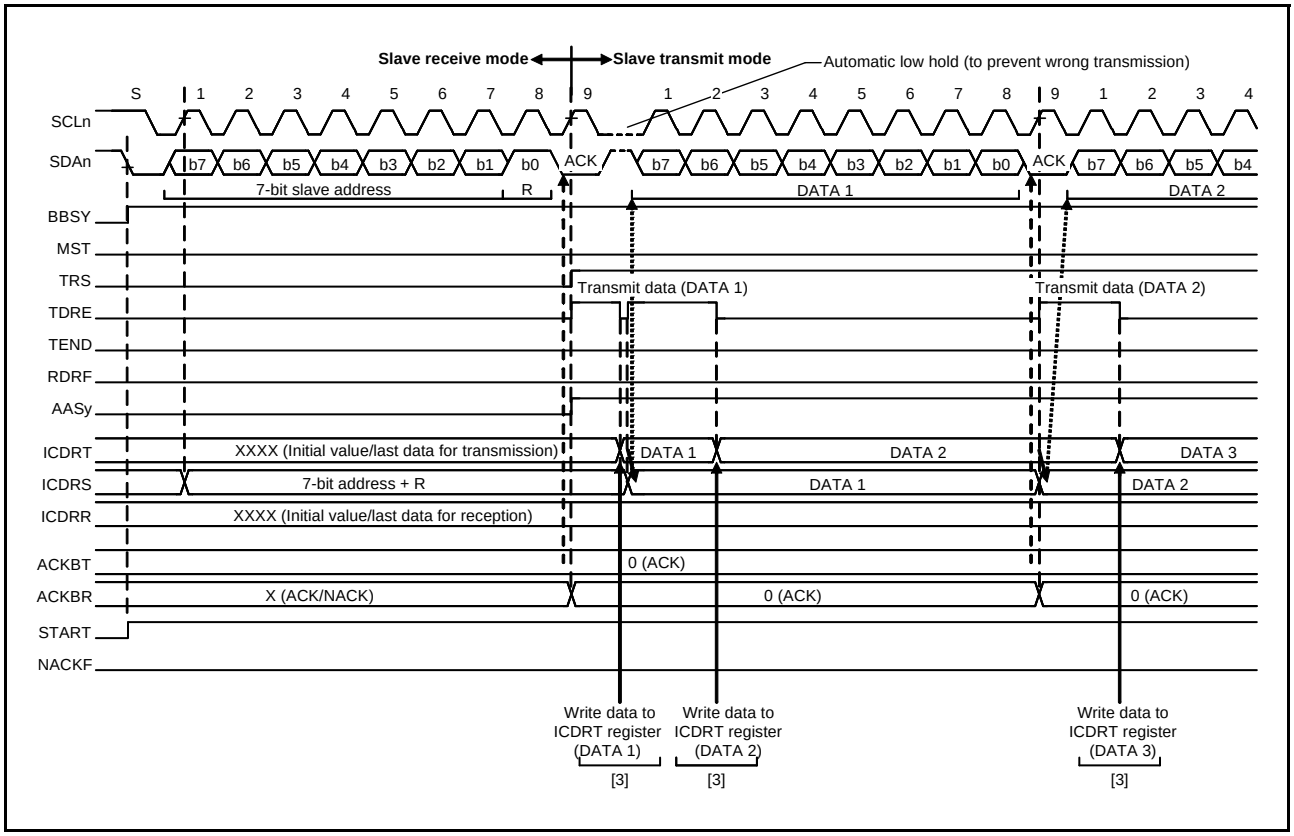


Figure 37.15 Slave Transmit Operation Timing (1) (7-Bit Address Format)

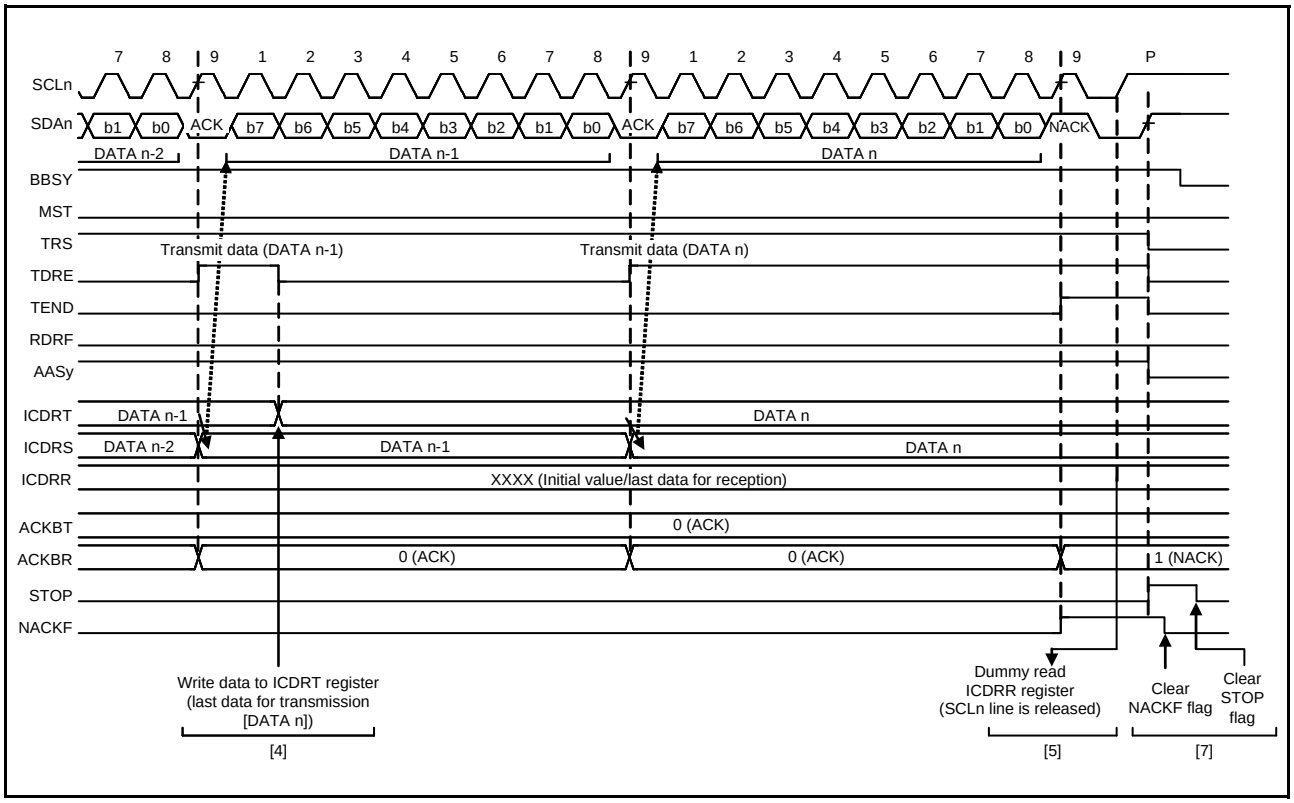


Figure 37.16 Slave Transmit Operation Timing (2)

37.3.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL and transmit data, and the RIIC returns acknowledgments as a slave device.

Figure 37.17 shows an example of usage of slave reception and Figure 37.18 and Figure 37.19 show the timing of operations in slave reception.

The following describes the procedure and operations for slave reception.

(1) Initial settings. For details, refer to section 37.3.2, Initial Settings.
After initial settings, the RIIC will stay in the standby state until it receives a slave address that it matches.

(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits ICSR1.HOA, GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth SCL (the clock signal) and outputs the value set in the ICMR3.ACKBT bit to the acknowledgment bit on the ninth SCL. If the value of the R/W# bit that was also received at this time is 0, the RIIC continues to place itself in slave receive mode and sets the ICSR2.RDRF flag to 1.

(3) After the ICSR2.STOP flag is confirmed to be 0 and the ICSR2.RDRF flag to be 1, dummy read the ICDRR register (the dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower 8 bits when the 10-bit address format is selected).

(4) When the ICDRR register is read, the RIIC automatically sets the ICSR2.RDRF flag to 0. If reading of the ICDRR register is delayed and a next byte is received while the RDRF flag is still set to 1, the RIIC holds the SCLn line low from one SCL cycle before the timing with which RDRF should be set. In this case, reading the ICDRR register releases the SCLn line from being held low.
When the ICSR2.STOP flag is 1 and the ICSR2.RDRF flag is also 1 or when all the data is completely received, read the ICDRR register.

(5) Upon detecting the stop condition, the RIIC automatically clears bits ICSR1.HOA, GCA, and AASy (y = 0 to 2) to 0.

(6) After checking that the ICSR2.STOP flag is 1, set the ICSR2.STOP flag to 0 for the next transfer operation.

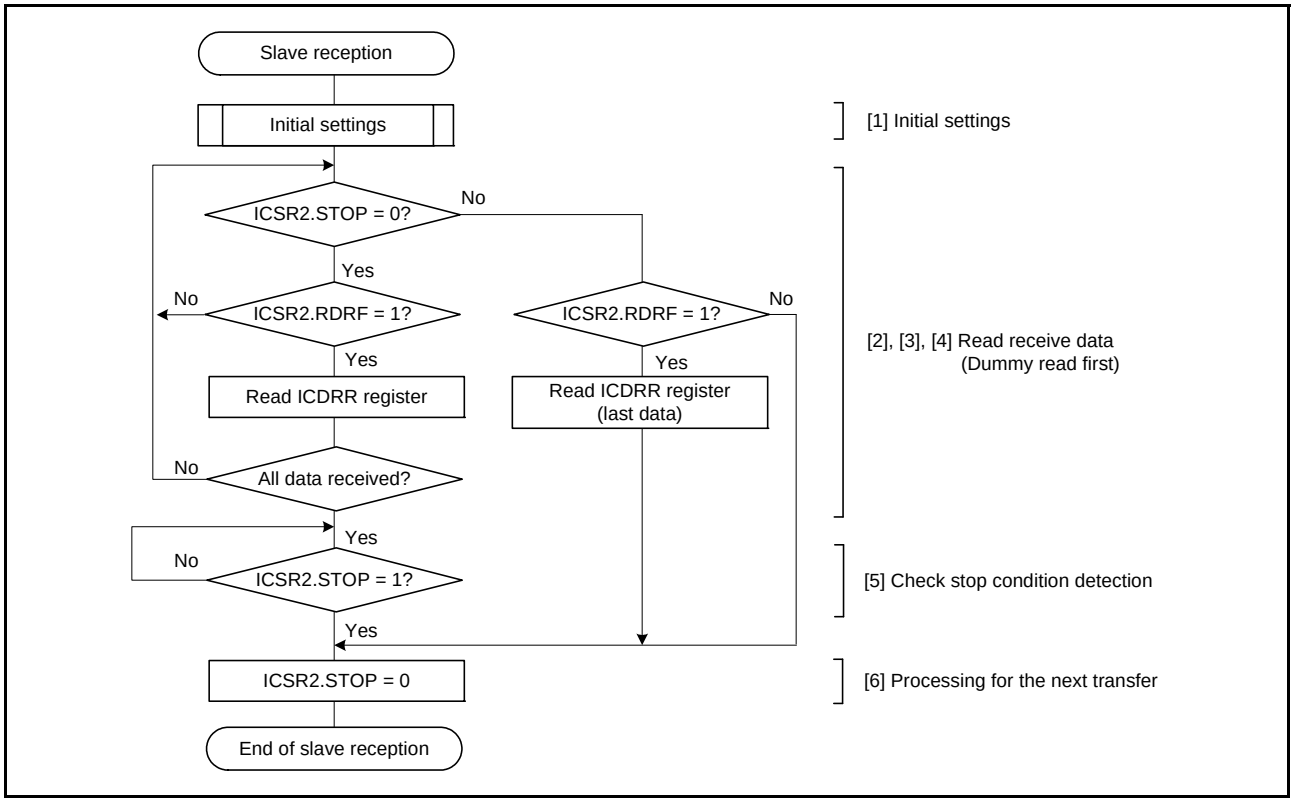


Figure 37.17 Example of Slave Reception Flowchart

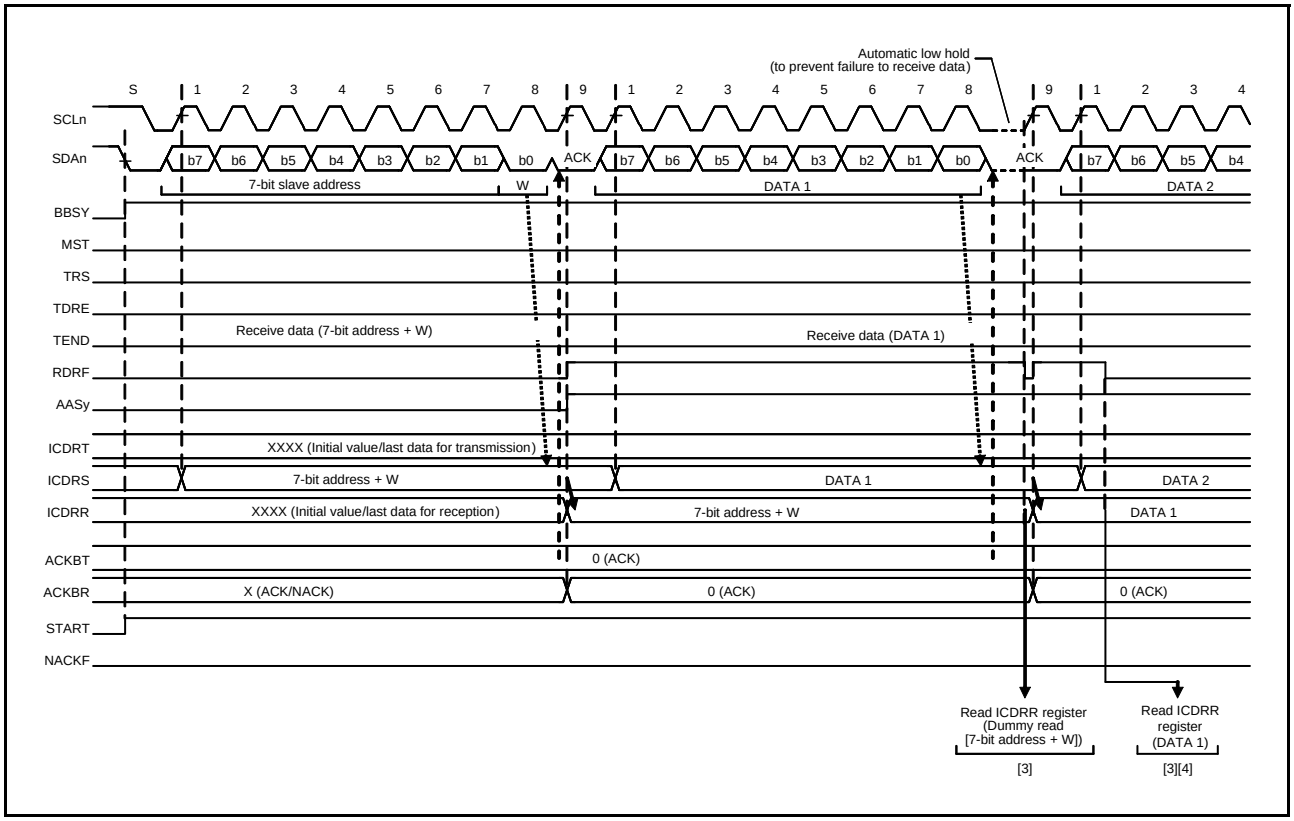


Figure 37.18 Slave Receive Operation Timing (1) (7-Bit Address Format, when RDRFS bit is 0)

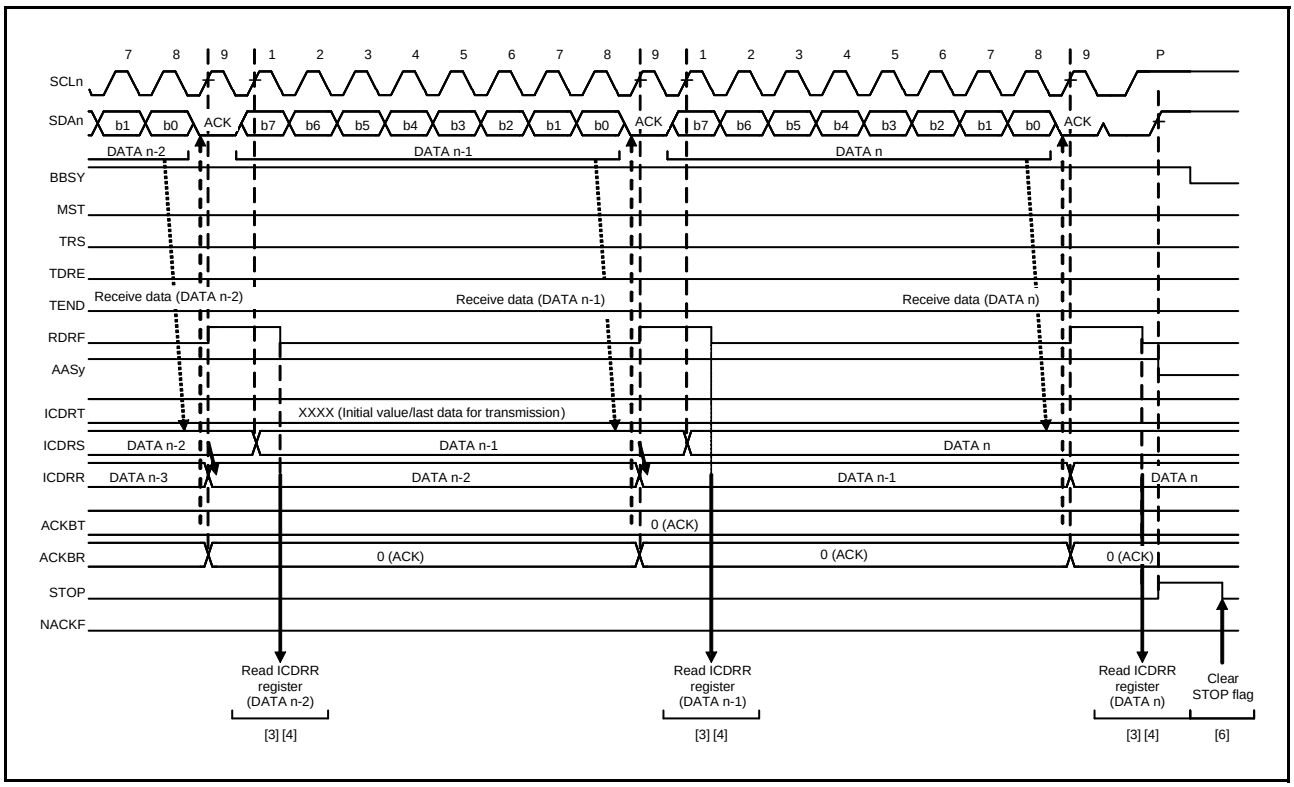


Figure 37.19 Slave Receive Operation Timing (2) (when RDRFS bit is 0)

37.4 SCL Synchronization Circuit

In generation of the SCL, the RIIC starts counting out the value for width at high level specified in the ICBRH register when it detects a rising edge on the SCLn line and drives the SCLn line low once counting of the width at high level is complete. When the RIIC detects the falling edge of the SCLn line, it starts counting out the width at low period specified in the ICBRL register, and then stops driving the SCLn line (releases the line) once counting of the width at low level is complete. The SCL 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 synchronization of the SCL signals must be handled bit by bit, the RIIC is equipped with a facility (the SCL synchronization circuit) for obtaining bit-by-bit synchronization of the SCL signals by monitoring the SCLn line while in master mode.

When the RIIC has detected a rising edge on the SCLn line and thus started counting out the width at high level specified in the ICBRH register, and the level on the SCLn line falls because an SCL signal is being generated by another master device, the RIIC stops counting when it detects the falling edge, drives the level on the SCLn line low, and starts counting out the width at low level specified in the ICBRL register. When the RIIC finishes counting out the width at low level, it stops driving the SCLn line low (i.e. releases the line). At this time, if the width at low level of the SCL signal from the other master device is longer than the width at low level set in the RIIC, the width at low level of the SCL signal will be extended. Once the width at low level for the other master device has ended, the SCL signal rises because the SCLn line has been released. When the RIIC finishes outputting the low period of the SCL, the SCLn line is released and the SCL rises. That is, in cases of contention of SCL signals from more than one master, the width at high level of the SCL signal is synchronized with that of the clock having the narrower width, and the width at low level of the SCL signal is synchronized with that of the clock having the broader width. However, such synchronization of the SCL signal is only enabled when the ICFER.SCLE bit is set to 1.

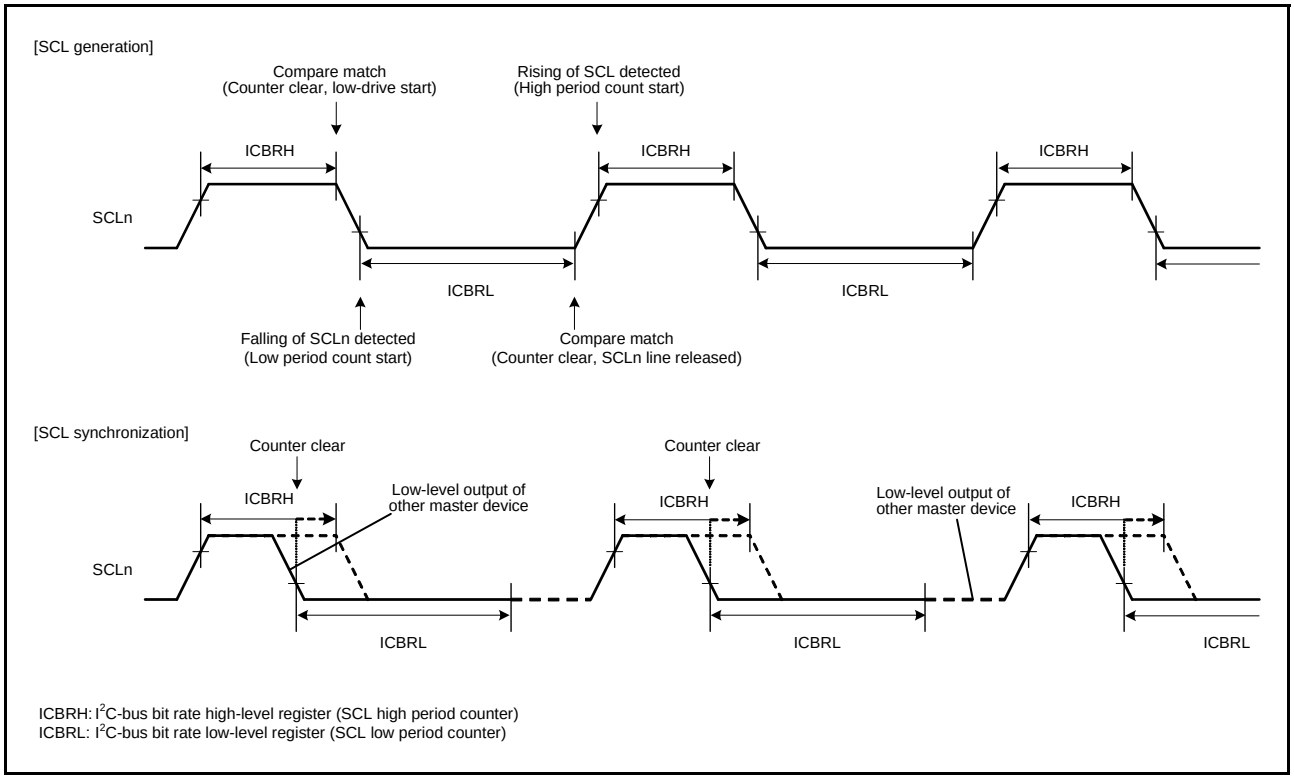


Figure 37.20 Generation and Synchronization of the SCL Signal from the RIIC

37.5 SDA Output Delay Function

The RIIC module incorporates a function for delaying output on the SDA line. The delay can be applied to all output (generation of the start, restart, and stop conditions, data, and the ACK and NACK signals) on the SDA line.

The SDA output delay function is used to delay the SDA output timing relative to falling edges of SCL to ensure that the SDA signal changes while the SCL is low and can be used to prevent erroneous operation of communications devices. This function is also used to satisfy the 300 ns (min.) data hold time prescribed by the SMBus specification.

The output delay function is enabled by setting the ICMR2.SDDL[2:0] bits to any value other than 000b, and disabled by setting the same bits to 000b.

When the SDA output delay function is enabled (the SDDL[2:0] bits are not “000b”), the SDA output delay counter counts the number of cycles set in the SDDL[2:0] bits of the count source selected by the ICMR2.DLCS bit (the internal reference clock (IICφ) or internal reference clock divided by 2 (IICφ/2)). On completion of counting of cycles of delay, the RIIC changes the bit being output as the SDA signal (generation of the start, restart, or stop condition, a new bit, or an ACK or NACK signal).

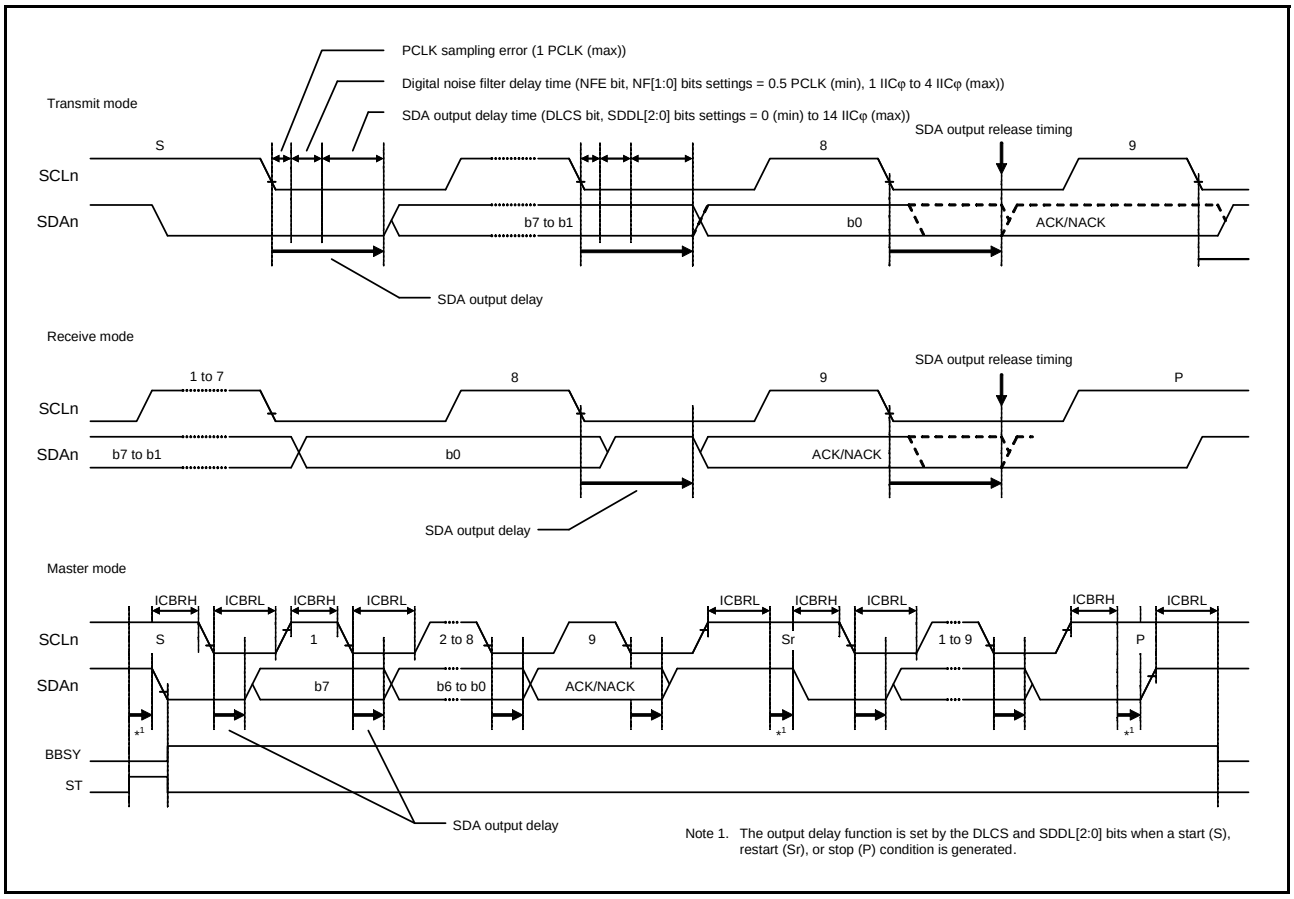


Figure 37.21 SDA Output Delay Function

37.6 Digital Noise Filters

The states of the SCLn and SDAn pins are conveyed to the internal circuitry through digital noise filters. Figure 37.22 is a block diagram of the digital noise filter.

The on-chip digital noise filter of each RIIC consists of four flip-flop circuit stages connected in series and a match detection circuit.

The number of effective stages in the digital noise filter is selected by the ICMR3.NF[1:0] bits. The selected number of effective stages determines the noise-filtering capability as a period from one to four cycles of IICφ.

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 for the number of effective flip-flop circuit stages selected by the ICMR3.NF[1:0] bits, the signal level is conveyed to the subsequent stages. 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 relatively 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.

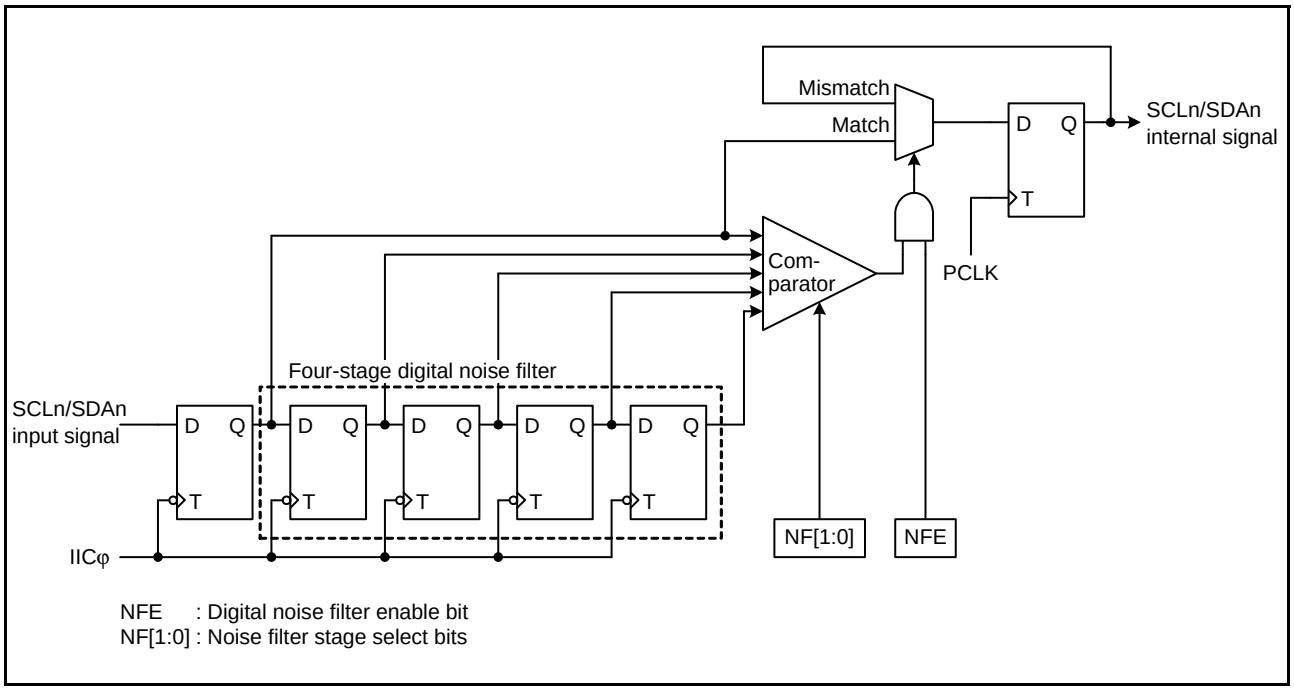


Figure 37.22 Block Diagram of the Digital Noise Filter

37.7 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.

37.7.1 Slave-Address Match Detection

The RIIC can set three unique slave addresses, and has a slave address detection function for each unique slave address. When the ICSER.SARyE bit (y = 0 to 2) is set to 1, the slave addresses set in registers SARUy and SARLy (y = 0 to 2) can be detected.

When the RIIC detects a match with its set slave address, the corresponding ICSR1.AASy flag (y = 0 to 2) is set to 1 on the rising edge of the ninth SCL, and the ICSR2.RDRF flag or the ICSR2.TDRE flag is set to 1 according to the level of the R/W# bit. This causes a receive data full interrupt (RXI) or transmit data empty interrupt (TXI) to be generated. The AASy flag is used to identify which slave address has been specified.

Figure 37.23 to Figure 37.25 show the AASy flag set timing in three cases.

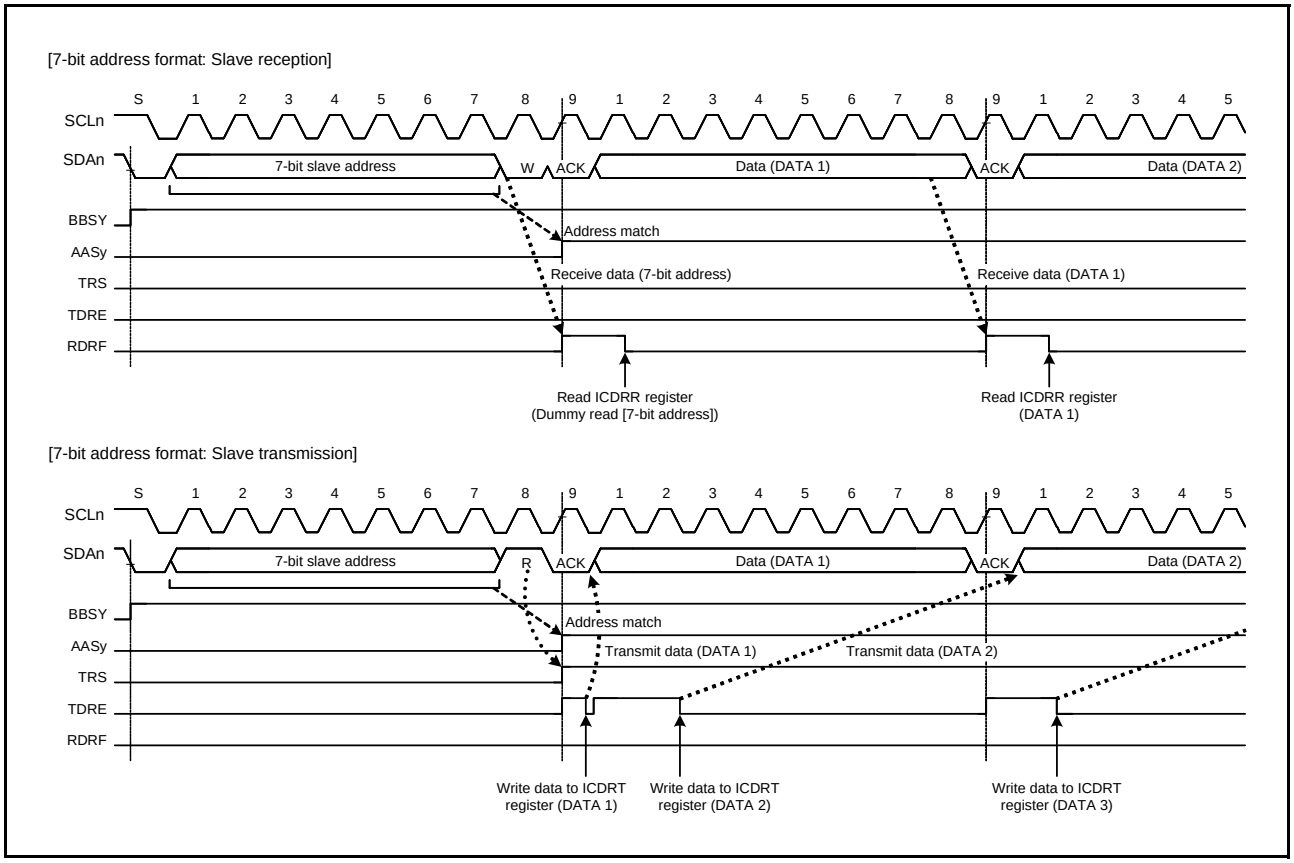


Figure 37.23 AASy Flag Set Timing with 7-Bit Address Format Selected

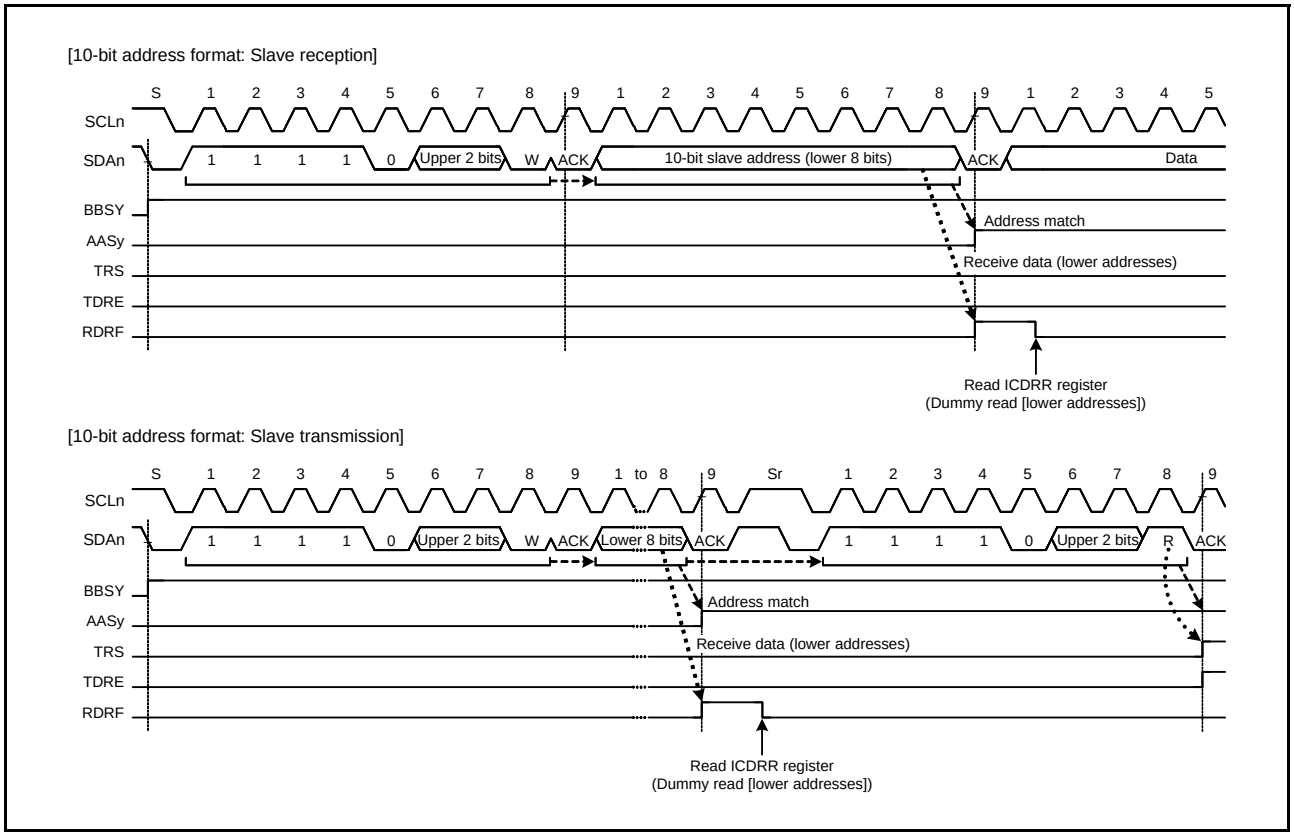


Figure 37.24 AASy Flag Set Timing with 10-Bit Address Format Selected

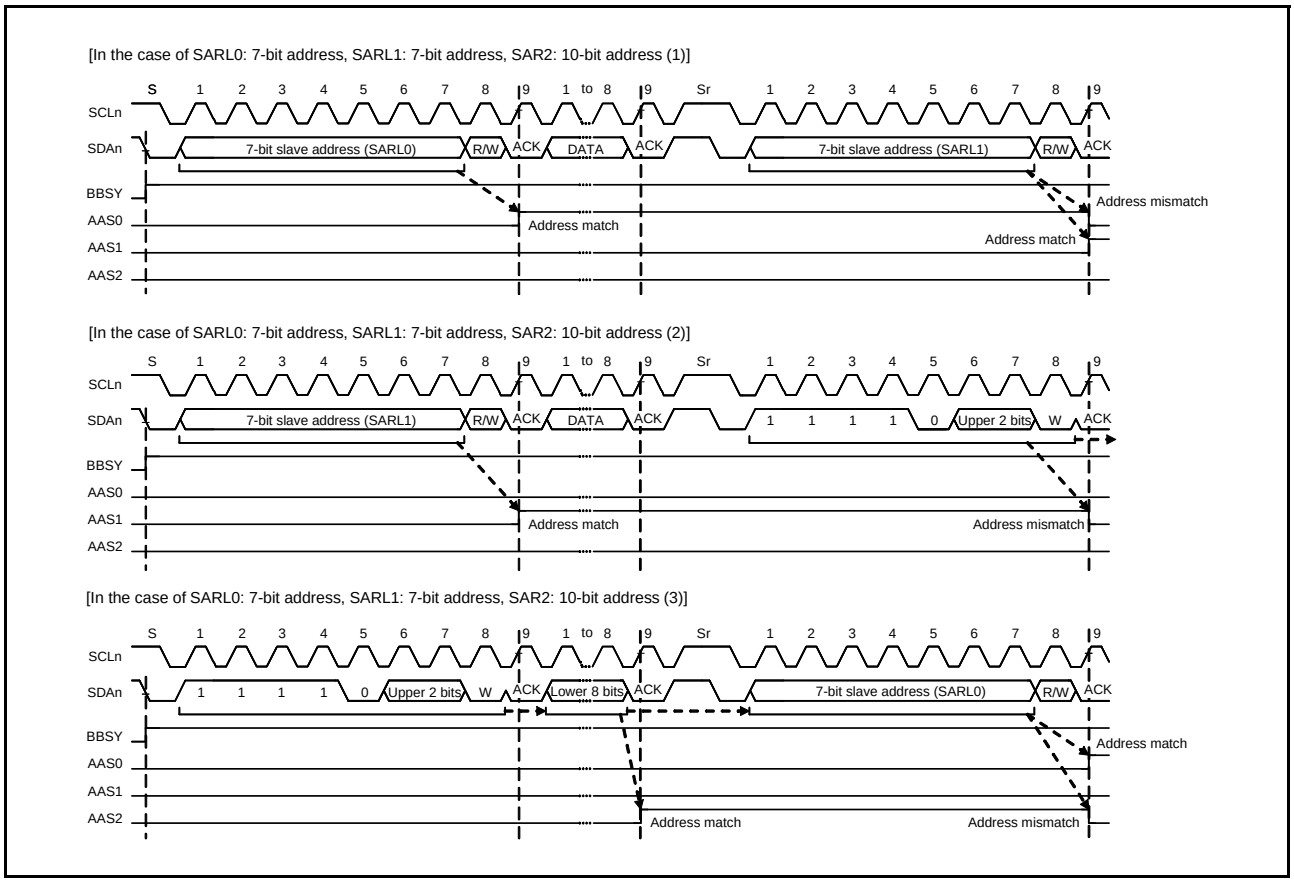


Figure 37.25 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed

37.7.2 Detection of the General Call Address

The RIIC also has a facility for detecting the general call address (0000 000b + 0 (write)). This is enabled by setting the ICSER.GCAE bit to 1.

If the address following a start or restart condition is 0000 000b + 1 (read) (start byte), the RIIC recognizes this as the address of a slave device with an “all-zero” address but not as the general call address.

When the RIIC detects the general call address, both the ICSR1.GCA flag and the ICSR2.RDRF flag are set to 1 on the rising edge of the ninth SCL. This leads to the generation of a receive data full interrupt (RXI). The value of the GCA flag can be confirmed to recognize that the general call address has been transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

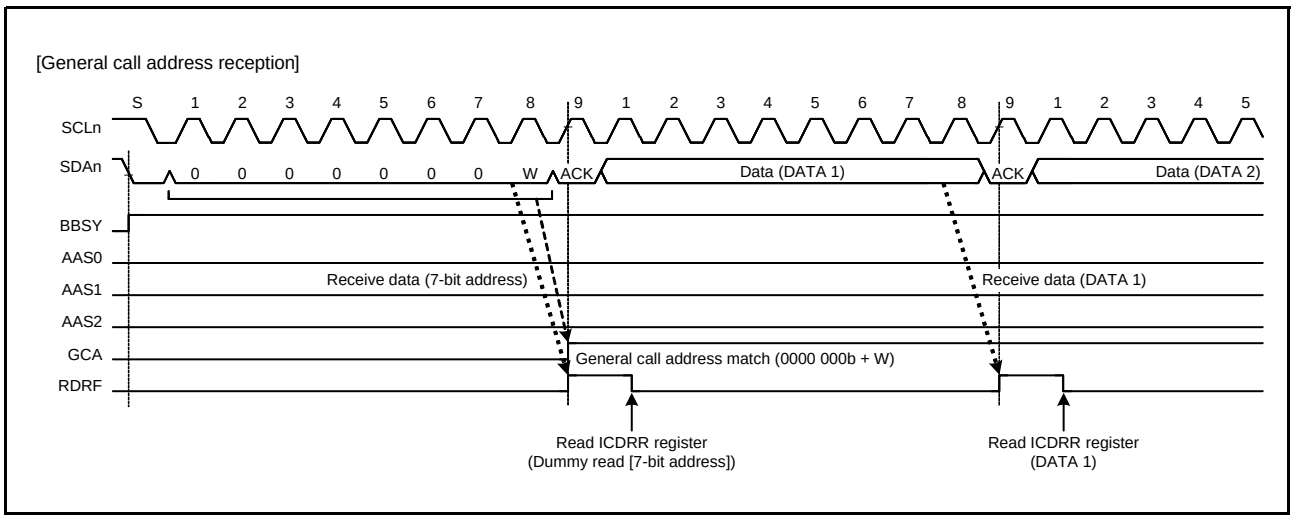


Figure 37.26 Timing of GCA Flag Setting during Reception of General Call Address

37.7.3 Device-ID Address Detection

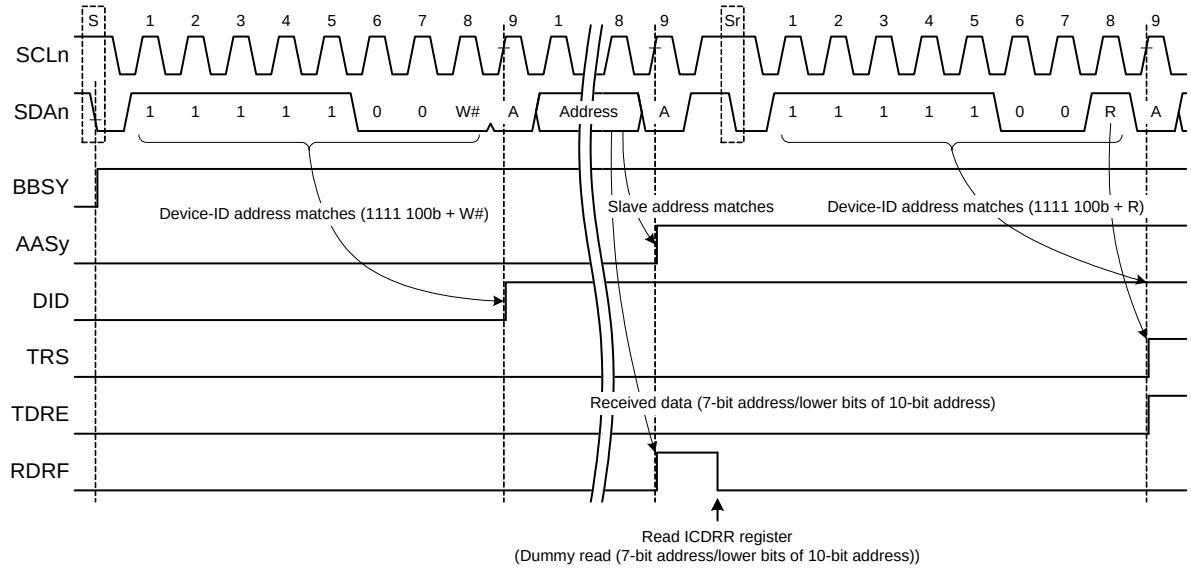
The RIIC module has a function to detect device-ID addresses complying with the I2C-bus specification. When the RIIC receives 1111 100b as the first seven bits of the first byte following a start condition or a restart condition while the ICSER.DIDE bit is 1, the RIIC recognizes the address as a device-ID address, sets the ICSR1.DID flag to 1 on the rising edge of the ninth SCL when the following R/W# bit is 0, and then compares the second and following bytes with its own slave address. If the received address matches the value in the slave address register, the RIIC sets the corresponding ICSR1.AASy flag (y = 0 to 2) to 1.

After that, when the first byte received after a start or restart condition is generated matches the device ID address (1111 100b) again and the following R/W# bit is 1, the RIIC does not compare the second and subsequent bytes and sets the ICSR2.TDRE flag to 1.

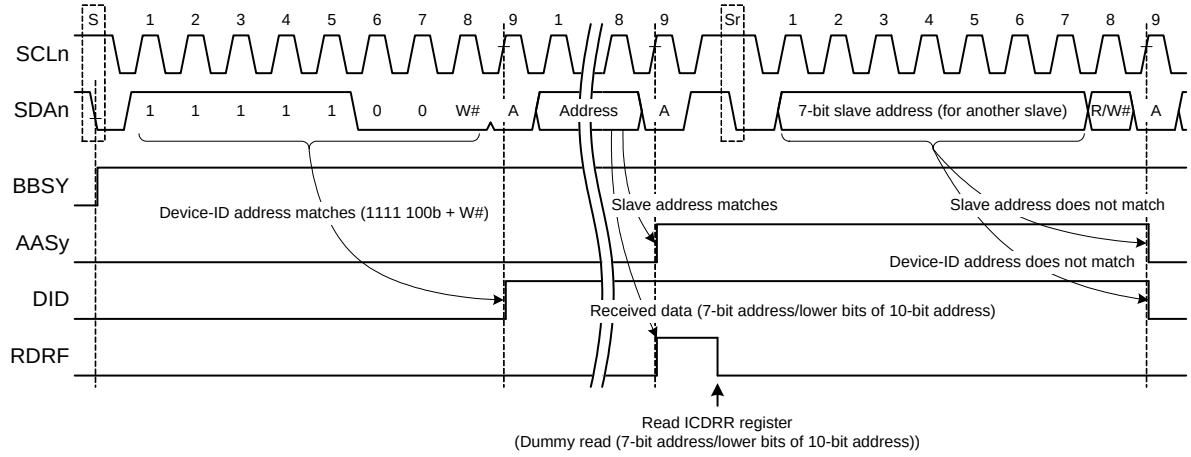
In the device-ID address detection function, the RIIC sets the DID flag to 0 if a match with the RIIC's own slave address is not obtained or a match with the device ID address is not obtained after a match with the RIIC's own slave address and the detection of a restart condition. If the first byte after detection of a start or restart condition matches the device ID address (1111 100b) and the R/W# bit is 0, the RIIC sets the DID flag to 1 and compares the second and subsequent bytes with the RIIC's slave address. If the R/W# bit is 1, the DID flag holds the previous value and the RIIC does not compare the second and subsequent bytes. Therefore, the reception of a device-ID address can be checked by reading the DID flag after confirming that TDRE flag is 1.

Furthermore, prepare the device-ID fields (3 bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal data for transmission. For details of the information that must be included in device-ID fields, contact NXP Semiconductors.

[Device-ID reception]



[When the address following a restart condition is not a device-ID address]



[When the address before the device-ID address + R had not matched the slave address]

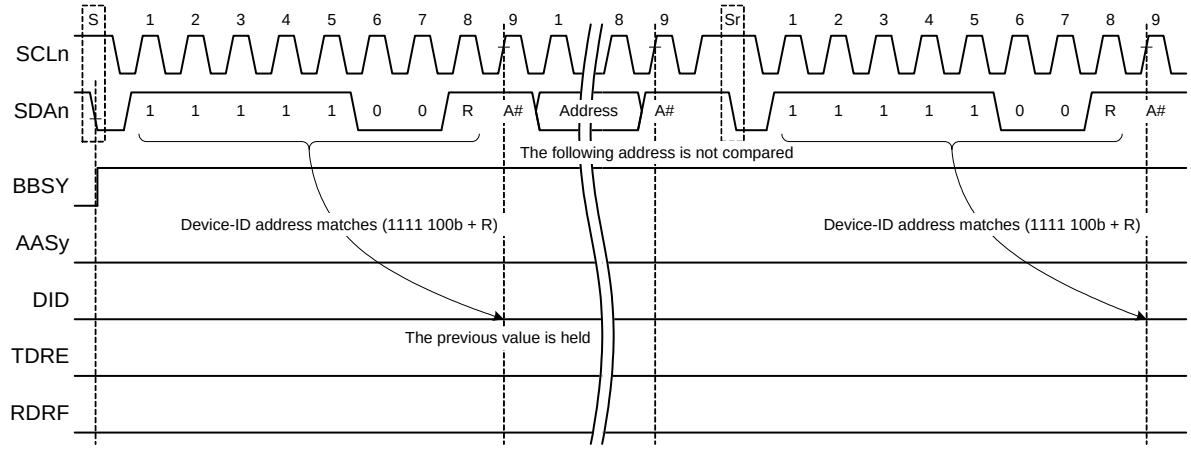


Figure 37.27 Set/Clear Timing of the AASy and DID Flags during Reception of Device-ID Address

37.7.4 Host Address Detection

The RIIC has a function to detect the host address while the SMBus is operating. When the ICSER.HOAE bit is set to 1 while the ICMR3.SMBS bit is 1, the RIIC can detect the host address (0001 000b) in slave receive mode (bits MST and TRS in the ICCR2 register are 00b).

When the RIIC detects the host address, the ICSR1.HOA flag is set to 1 at the rising edge of the ninth SCL, and at the same time, the ICSR2.RDRF flag is set to 1 when the R/W# bit is 0 (Wr bit). This causes a receive data full interrupt (RXI) to be generated. The HOA flag is used to recognize that the host address was sent from the smart battery or other devices.

If the bit following the host address (0001 000b) is an Rd bit (R/W# bit is 1), the RIIC can also detect the host address. After the host address is detected, the RIIC operates in the same manner as normal slave operation.

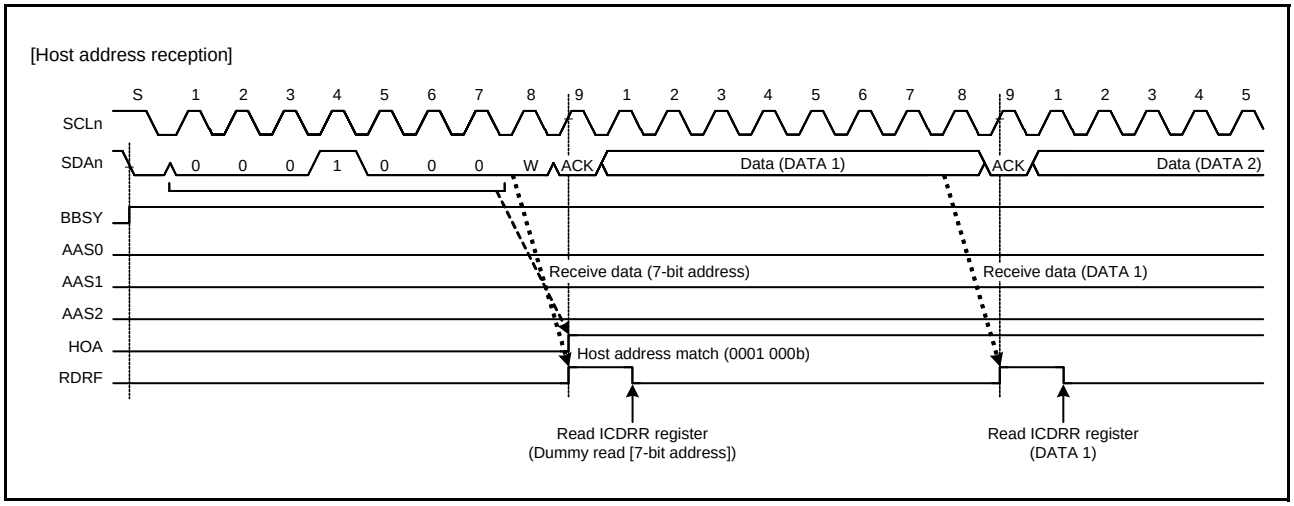


Figure 37.28 HOA Flag Set Timing during Reception of Host Address

37.8 Automatic Low-Hold Function for SCL

37.8.1 Function to Prevent Wrong Transmission of Transmit Data

If the shift register (ICDRS) is empty when data have not been written to the I2C-bus transmit data register (ICDRT) with the RIIC in transmission mode (ICCR2.TRS bit is 1), the SCLn line is automatically held low over the intervals shown below. This low period is extended until data for transmission have been written, which prevents the unintended transmission of erroneous data.

Master transmit mode

- Low period after a start condition or restart condition is generated
- Low period between the ninth clock pulse of one transfer and the first clock pulse of the next

Slave transmit mode

- Low period between the ninth clock pulse of one transfer and the first clock pulse of the next

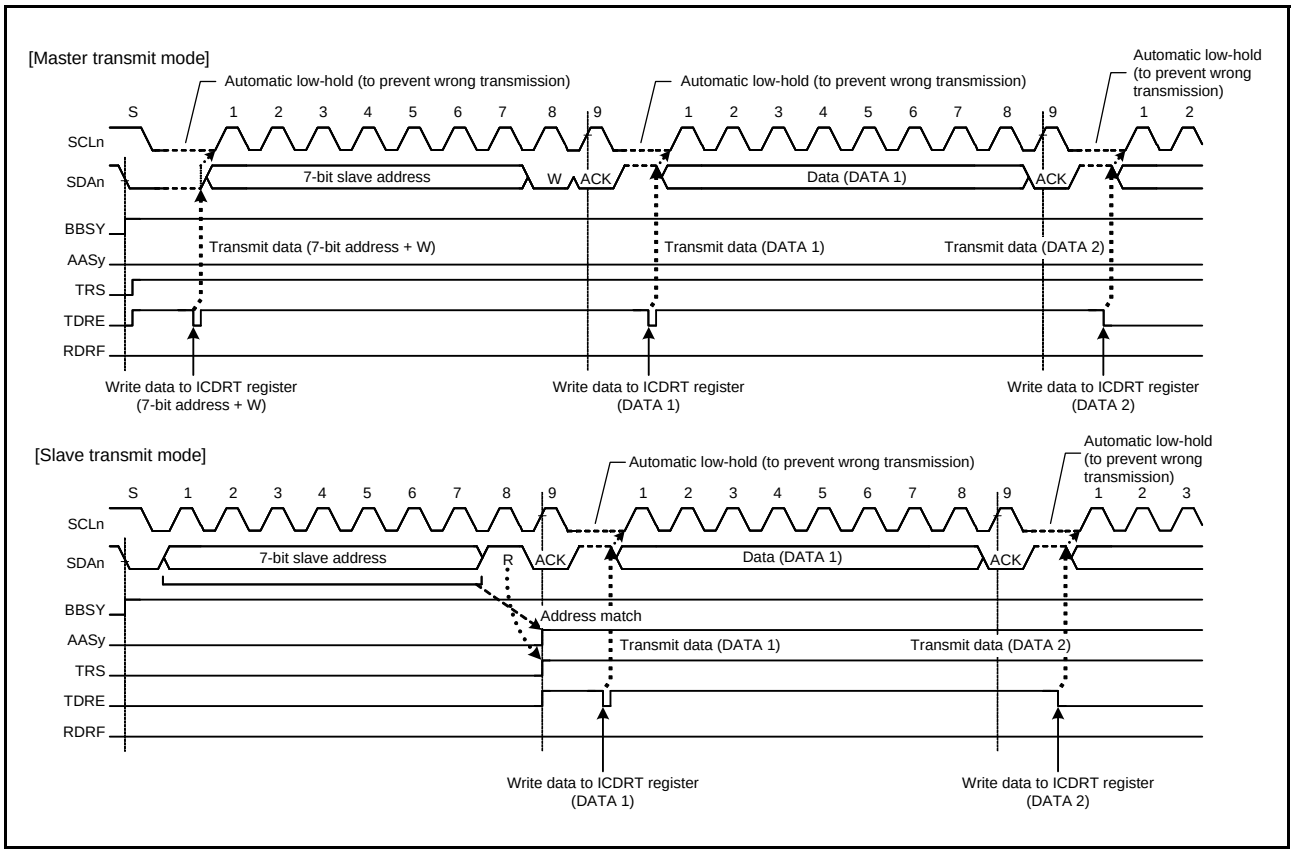


Figure 37.29 Automatic Low-Hold Operation in Transmit Mode

37.8.2 NACK Reception Transfer Suspension Function

The RIIC has a function to suspend transfer operation when NACK is received in transmit mode (ICCR2.TRS bit is 1). This function is enabled when the ICFER.NACKE bit is set to 1 (transfer suspension enabled). If the next transmit data has already been written (ICSR2.TDRE flag is 0) when NACK is received, next data transmission at the falling edge of the ninth SCL 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 data transmission is suspended (ICSR2.NACKF flag is 1) by this function, the following data transmission and data reception are not started. To resume data transfer, set the NACKF flag to 0. In master transmit mode, restart data transfer by setting the NACKF flag to 0 after generating a restart condition, or restart data transfer from a start condition after generating a stop condition then setting the NACKF flag to 0.

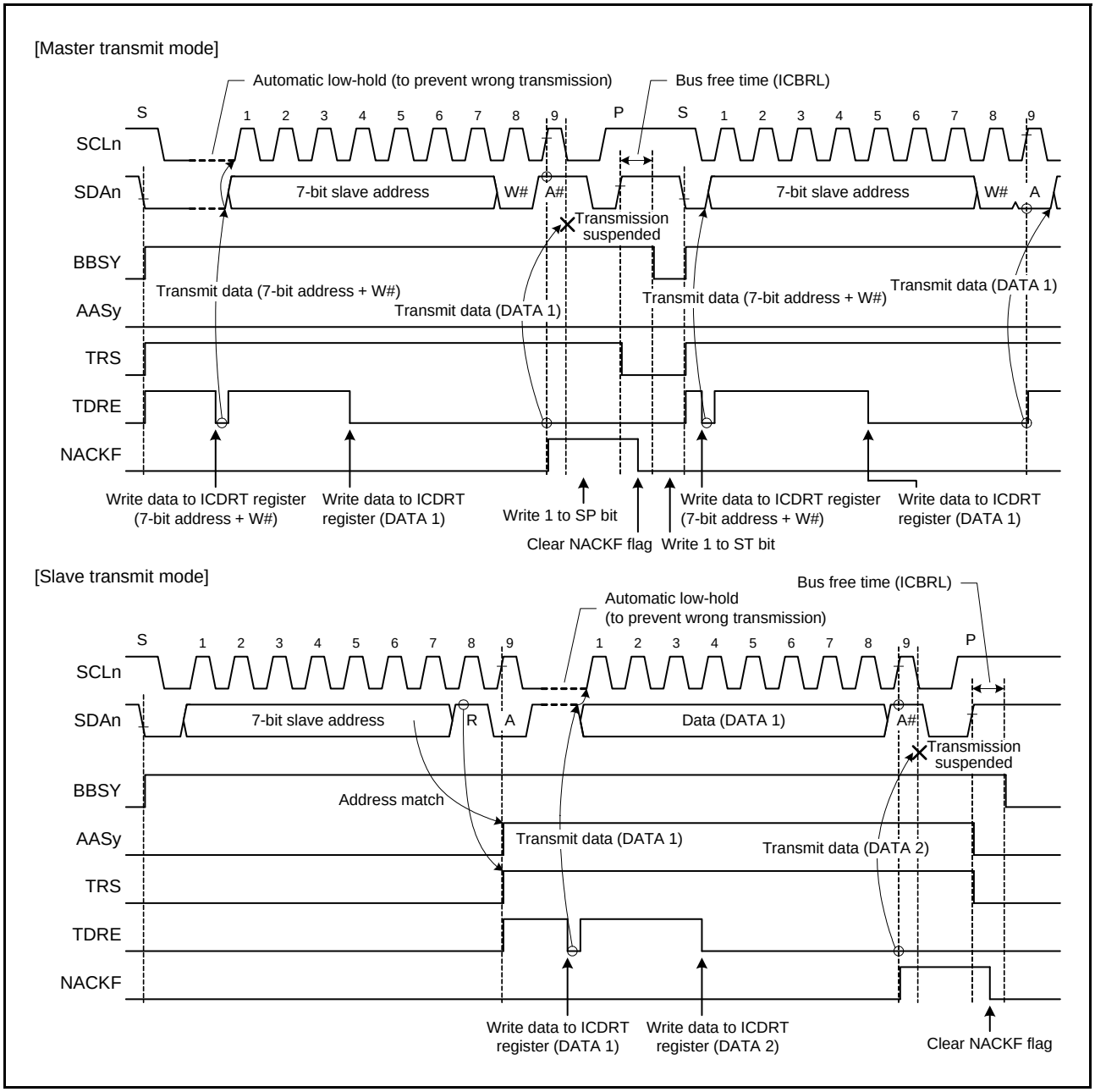


Figure 37.30 Suspension of Data Transmission When NACK is Received (NACKE = 1)

37.8.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (ICDRR) read is delayed for a period of one transfer byte or more with receive data full (ICSR2.RDRF flag is 1) in receive mode (ICCR2.TRS bit is 0), the RIIC holds the SCLn line low automatically immediately before the next data is received to prevent failure to receive data.
This function to prevent failure to receive data using the automatic low-hold function is also enabled even if the read processing of the final receive data is delayed and, in the meantime, the RIIC's own slave address is designated after a stop condition is generated. 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 generated.
Sections in which the SCLn line is held low can be selected with a combination of the WAIT and RDRFS bits in the ICMR3 register.

(1) 1-Byte Receive Operation and Automatic Low-Hold Function Using the WAIT Bit

When the ICMR3.WAIT bit is set to 1, the RIIC performs 1-byte receive operation using the WAIT bit function. Furthermore, when the ICMR3.RDRFS bit is 0, the RIIC automatically sends the ICMR3.ACKBT bit value for the acknowledgment bit in the period from the falling edge of the eighth SCL to the falling edge of the ninth SCL, and automatically holds the SCLn line low at the falling edge of the ninth SCL using the WAIT bit function. This low-hold is released by reading data from the ICDRR register, which enables bytewise receive operation.
The WAIT bit function is enabled for receive bytes after a match with the RIIC's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

(2) 1-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the RDRFS Bit

When the ICMR3.RDRFS bit is set to 1, the RIIC performs 1-byte receive operation using the RDRFS bit function. When the RDRFS bit is set to 1, the ICSR2.RDRF flag (receive data full) is set to 1 at the rising edge of the eighth SCL, and the SCLn line is automatically held low at the falling edge of the eighth SCL. This low-hold is released by writing a value to the ICMR3.ACKBT bit, but cannot be released by reading data from the ICDRR register, which enables receive operation by the ACK/NACK transmission control according to the data received in byte units.
The RDRFS bit function is enabled for receive bytes after a match with the RIIC's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

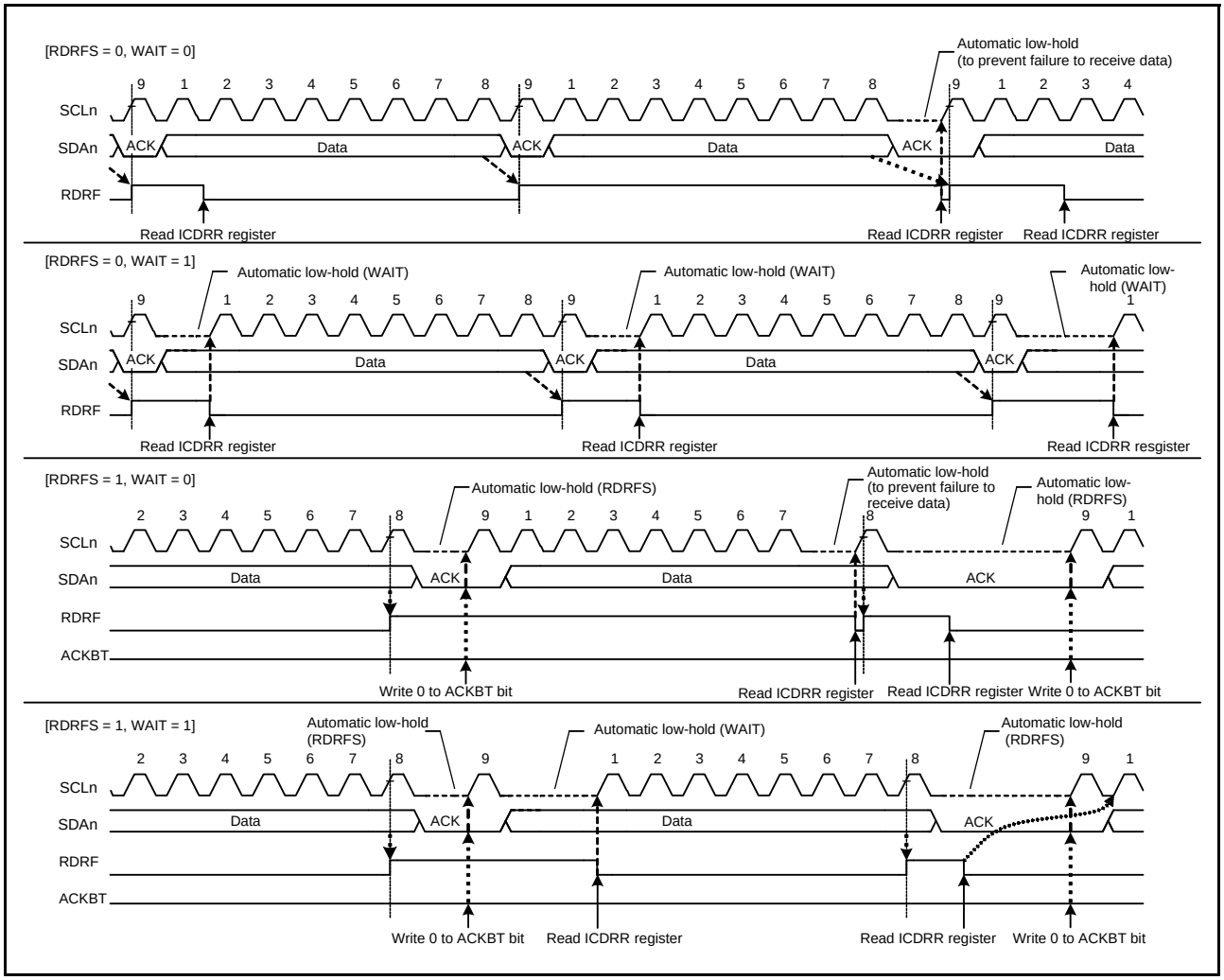


Figure 37.31 Automatic Low-Hold Operation in Receive Mode (Using RDRFS and WAIT Bits)

37.9 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I2C-bus specification, the RIIC has functions to prevent double-generation of a start condition, to detect arbitration-lost during transmission of NACK, and to detect arbitration-lost in slave transmit mode.

37.9.1 Master Arbitration-Lost Detection (MALE Bit)

The RIIC drives the SDAn line low to generate a start condition. However, if the SDAn line has already been driven low by another master device generating a start condition, the RIIC causes arbitration to be lost, so priority is given to transfer by the other master device. Similarly, if the ICCR2.ST bit is set to 1 while the ICCR2.BBSY flag is 1 (bus busy state), arbitration is lost, so priority is given to transfer by the other master device. No start condition is generated in this case.
When a start condition is generated successfully, if the data for transmission including the address bits (i.e. the internal SDA output level) and the level on the SDAn line do not match (the high output as the internal SDA output; i.e. the SDAn pin is in the high-impedance state) and the low is detected on the SDAn line, the RIIC loses the arbitration.
After a master arbitration-lost is generated, 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 master arbitration-lost is detected when the following conditions are met while the ICFER.MALE bit is 1 (master arbitration-lost detection enabled).

Conditions for detecting master arbitration-lost

- Non-matching of the internal level for output on SDA and the level on the SDAn line after a start condition was generated by setting the ICCR2.ST bit to 1 while the ICCR2.BBSY flag was set to 0 (erroneous generation of a start condition)
- Setting of the ICCR2.ST bit to 1 while the BBSY flag is set to 1 (start condition double-generation error)
- When the transmit data excluding acknowledgment bit (internal SDA output level) does not match the level on the SDAn line in master transmit mode (bits MST and TRS in the ICCR2 register = 11b)

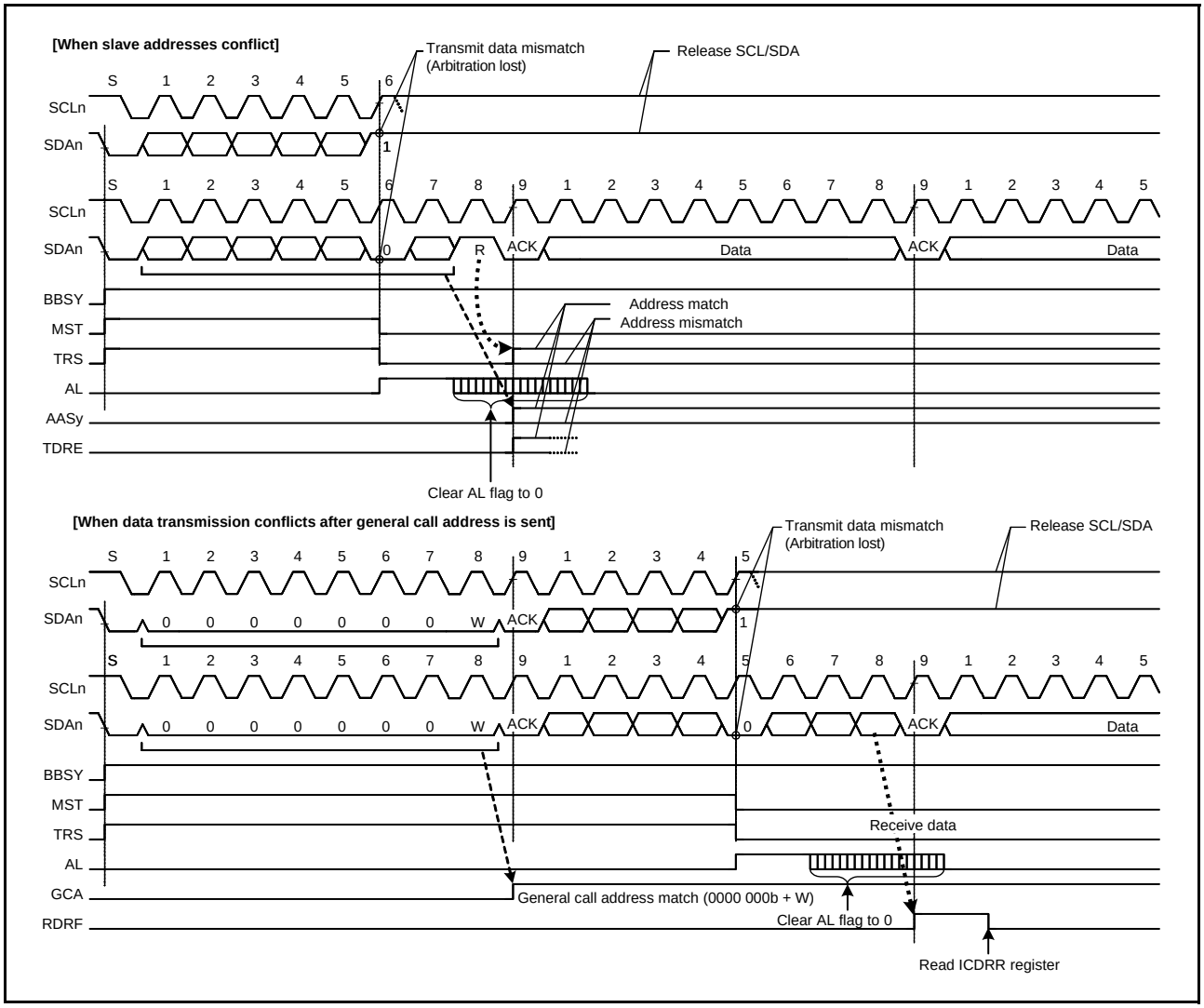


Figure 37.32 Examples of Master Arbitration-Lost Detection (MALE = 1)

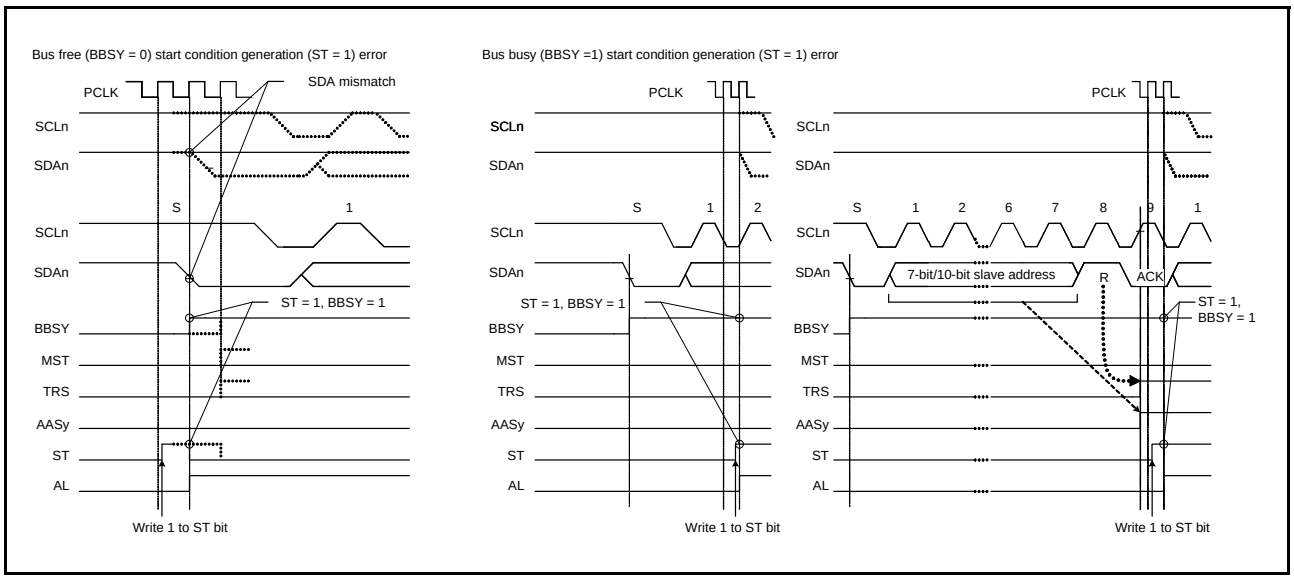


Figure 37.33 Arbitration-Lost When a Start Condition is Generated (MALE = 1)

37.9.2 NACK Transmission Arbitration-Lost Detection Function (NALE Bit)

The RIIC has a function to cause arbitration to be lost if the internal SDA output level does not match the level on the SDAn line (the high output as the internal SDA output; i.e. the SDAn pin is in the high-impedance state) and the low is detected on the SDAn line during transmission of NACK in receive mode. Arbitration is lost due to a conflict of NACK transmission and ACK transmission when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such conflict occurs when multiple master devices send/receive the same information through a single slave device. Figure 37.34 shows an example of NACK transmission arbitration-lost detection.

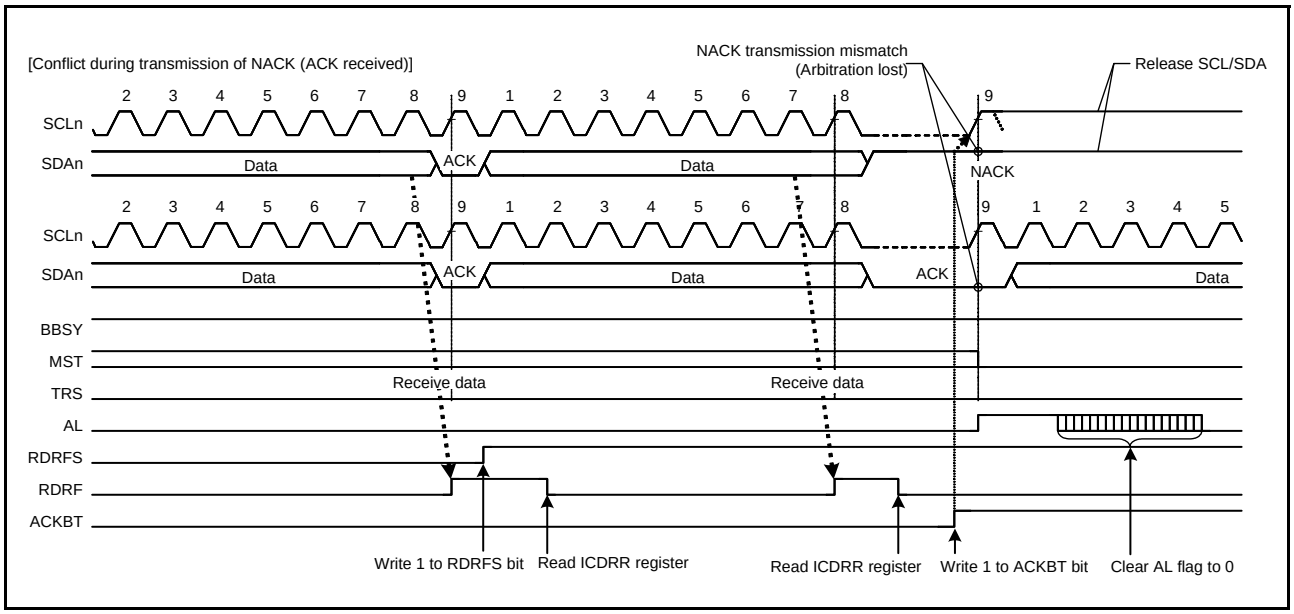


Figure 37.34 Example of NACK transmission arbitration-lost Detection (NALE = 1)

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 second byte 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 generates a stop condition. Therefore, the generation of the stop condition conflicts with the SCL 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 a NACK transmission arbitration-lost occurs, the RIIC immediately cancels the slave address matched state and enters slave receive mode. This prevents a stop condition from being generated, preventing a communication failure on the bus. Also, in the ARP command processing of SMBus, it is possible to omit surplus processing (FFh transmission processing) after NACK transmission when the UDID (Unique Device Identifier) of “Assign Address” does not match and after the NACK transmission of the “Get UDID (General)” after the “Assign Address” is confirmed.

The RIIC detects NACK transmission arbitration-lost when the following condition is met with the ICFER.NALE bit set to 1 (NACK transmission arbitration-lost is enabled).

Condition for detecting NACK transmission arbitration-lost

- When the internal SDA output level does not match the SDAn line (ACK is received) during transmission of NACK (ICMR3.ACKBT bit = 1)

37.9.3 Slave Arbitration-Lost Detection (SALE Bit)

The RIIC has a function to cause arbitration to be lost if the data for transmission (i.e. the internal SDA output level) and the level on the SDAn line do not match (the high output as the internal SDA output; i.e. the SDAn pin is in the high-impedance state and the low is detected on the SDAn line in slave transmit mode). The slave arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identifier) of SMBus.
When the slave arbitration-lost occurs, the RIIC is immediately released from the slave address matched state and enters slave receive mode. This function can detect a data conflict during UDID transmission of SMBus and omit surplus processing (FFh transmission processing) after the data conflict.
The RIIC detects slave arbitration-lost when the following condition is met with the ICFER.SALE bit set to 1 (slave arbitration-lost detection enabled).

Condition for detecting slave arbitration-lost

- When transmit data excluding acknowledgment bit (internal SDA output level) does not match the SDAn line in slave transmit mode (bits MST and TRS in the ICCR2 register are 01b)

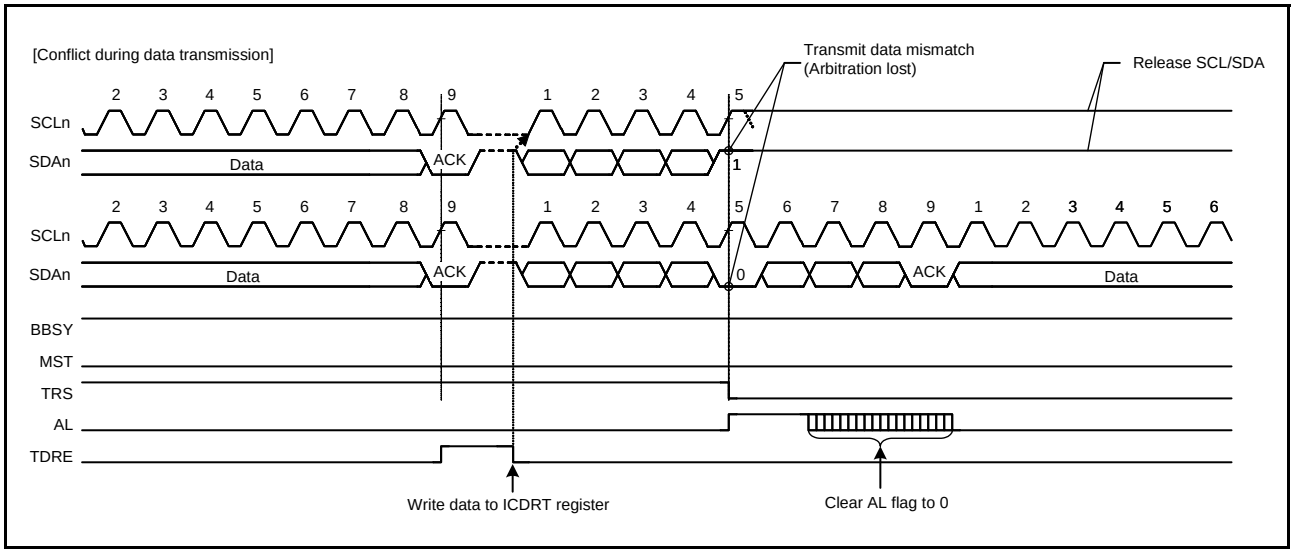


Figure 37.35 Example of Slave Arbitration-Lost Detection (SALE = 1)

37.10 Start Condition/Restart Condition/Stop Condition Generating Function

37.10.1 Generating a Start Condition

The RIIC generates a start condition when the ICCR2.ST bit is set to 1.
When the ST bit is set to 1, a start condition generation request is made and the RIIC generates a start condition when the ICCR2.BBSY flag is 0 (bus free state). When a start condition is generated normally, the RIIC automatically shifts to the master transmit mode.
A start condition is generated in the following sequence.

Start condition generation

(1) Drive the SDAn line low (high to low).
(2) Secure the time set in the ICBRH register and the start condition hold time.
(3) Drive the SCLn line low (high to low).
(4) Detect the low level on the SCLn line and secure the low period of the signal on the SCLn line set in the ICBRL register.

37.10.2 Generating a Restart Condition

The RIIC generates a restart condition when the ICCR2.RS bit is set to 1.
When the RS bit is set to 1, a restart condition generation request is made and the RIIC generates a restart condition when the ICCR2.BBSY flag is 1 (bus busy state) and the ICCR2.MST bit is 1 (master mode).
A restart condition is generated in the following sequence.

Restart condition generation

(1) Release the SDAn line.
(2) Secure the low period of the signal on the SCLn line set in the ICBRL register.
(3) Release the SCLn line (low to high).
(4) Detect the high level on the SCLn line and secure the time set in the ICBRL register and the restart condition setup time.
(5) Drive the SDAn line low (high to low).
(6) Secure the time set in the ICBRH register and the restart condition hold time.
(7) Drive the SCLn line low (high to low).
(8) Detect the low level on the SCLn line and secure the low period of the signal on the SCLn line set in the ICBRL register.

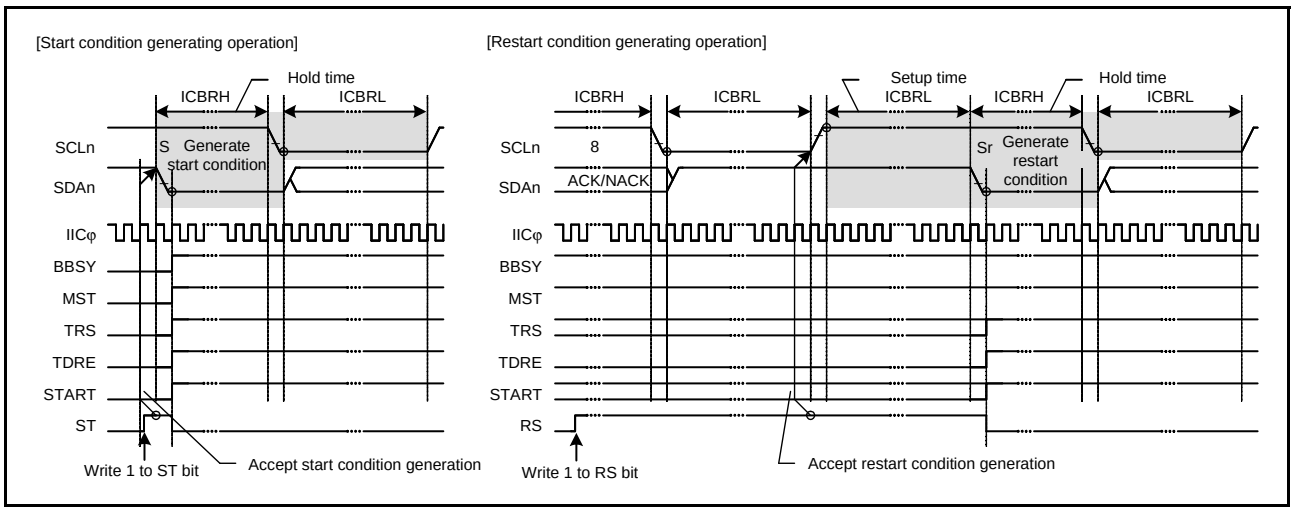


Figure 37.36 Start Condition/Restart Condition Generation Timing (ST and RS Bits)

37.10.3 Generating a Stop Condition

The RIIC generates a stop condition when the ICCR2.SP bit is set to 1.
When the SP bit is set to 1, a stop condition generation request is made and the RIIC generates a stop condition when the ICCR2.BBSY flag is 1 (bus busy state) and the ICCR2.MST bit is 1 (master mode).
A stop condition is generated in the following sequence.

Stop condition generation

(1) Drive the SDAn line low (high to low).
(2) Secure the low period of the signal on the SCLn line set in the ICBRL register.
(3) Release the SCLn line (low to high).
(4) Detect the high level on the SCLn line and secure the time set in the ICBRH register and the stop condition setup time.
(5) Release the SDAn line (low to high).
(6) Secure the time set in the ICBRL register and the bus free time.
(7) Set the BBSY flag to 0 (to release the bus mastership).

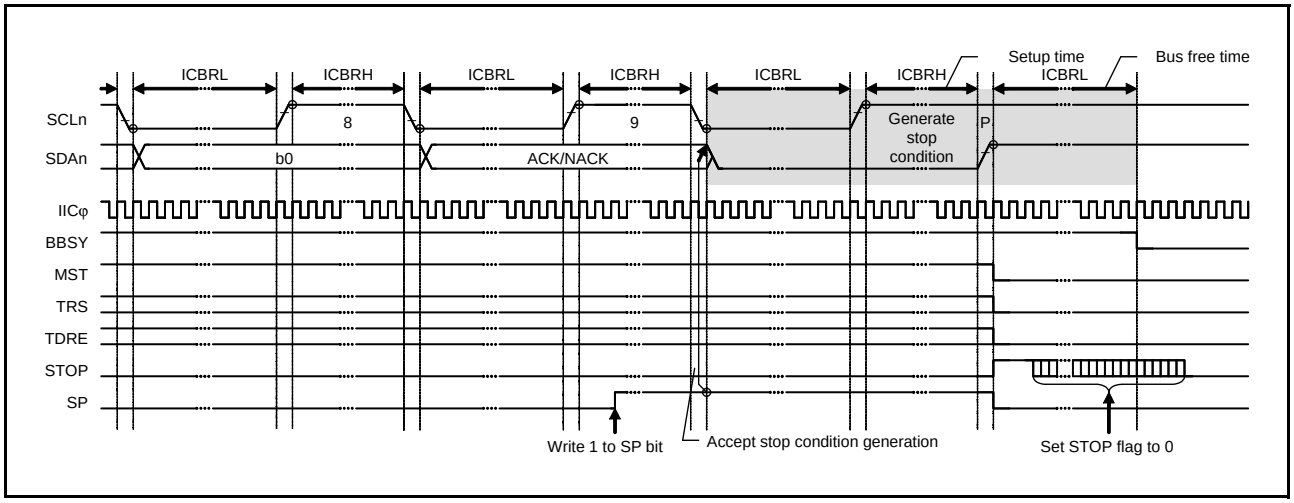


Figure 37.37 Stop Condition Generation Timing (SP Bit)

37.11 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 additional SCL to release the bus from a hung state due to clock signals being out of synchronization, the RIIC reset function, and internal reset function.
By checking bits SCLO, SDAO, SCLI, and SDAI in the ICCR1 register, it is possible to see whether the RIIC or its partner in communications is placing the low level on the SCLn or SDAn lines.

37.11.1 Timeout Function

The RIIC includes a timeout function for detecting when the SCLn line has been stuck longer than the predetermined time. The RIIC can detect an abnormal bus state by monitoring that the SCLn line is stuck low or high for a predetermined time.
The timeout function monitors the SCLn line state and counts the low period or high period using the internal counter. The timeout function resets the internal counter each time the SCLn line changes (rising or falling), but continues to count unless the SCLn line changes. If the internal counter overflows due to no SCLn line change, the RIIC can detect the timeout and report the bus hung state.
This timeout function is enabled when the ICFER.TMOE bit is 1. It detects a hung state that the SCLn line is stuck low or high during the following conditions:

- The bus is busy (ICCR2.BBSY flag is 1) in master mode (ICCR2.MST bit is 1).
- The RIIC's own slave address is detected (ICSR1 register is not 00h) and the bus is busy (ICCR2.BBSY flag is 1) in slave mode (ICCR2.MST bit is 0).
- The bus is free (ICCR2.BBSY flag is 0) while generation of a start condition is requested (ICCR2.ST bit is 1).

The internal counter of the timeout function works using the internal reference clock (IICφ) set by the ICMR1.CKS[2:0] bits as a count source. It functions as a 16-bit counter when long mode is selected (ICMR2.TMOS bit is 0) or a 14-bit counter when short mode is selected (TMOS bit is 1).
The SCLn line level (low/high or both levels) during which this counter is activated can be selected by the setting of bits TMOH and TMOL in the ICMR2 register. If both bits TMOL and TMOH are set to 0, the internal counter does not work.

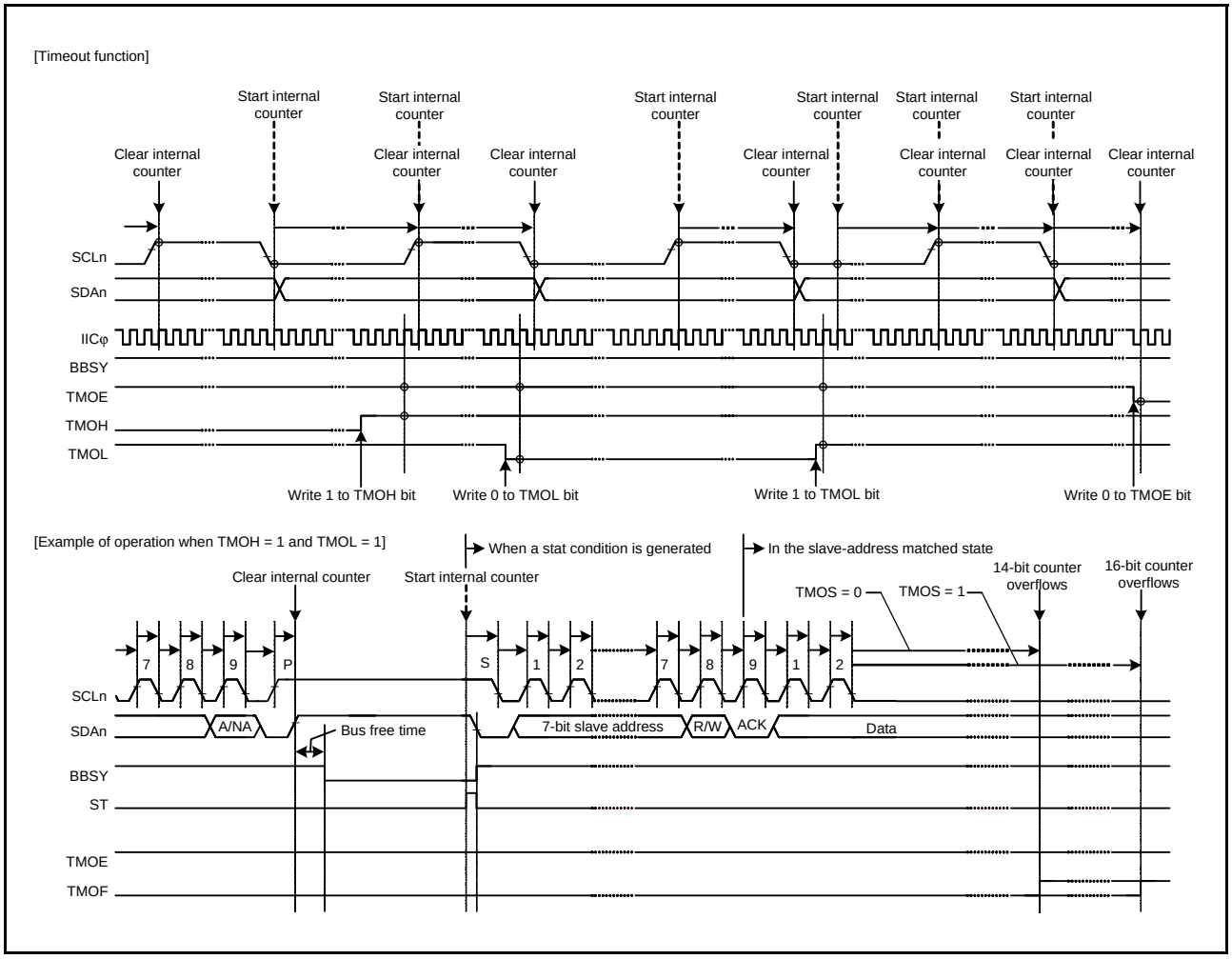


Figure 37.38 Timeout Function

37.11.2 Additional SCL Output Function

In master mode, the RIIC module has a facility for the output of additional SCL to release the SDAn line from being held low by the slave device 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 from the state of being stuck low by including additional SCL output from the RIIC with single cycles of the SCL as the unit if the RIIC cannot generate a stop condition because the slave device is holding the SDAn line low. Do not use this function in normal situations. Using it when communications are proceeding correctly will lead to malfunctions.
When the ICCR1.CLO bit is set to 1, an additional clock pulse at the frequency set by the ICMR1.CKS[2:0] bits and the ICBRH and ICBRL registers is output from the SCLn pin. After output of this clock pulse, the CLO bit automatically becomes 0. The SCLn pin is held low when the ICCR2.BBSY flag is 1 and held high when the BBSY flag is 0.
Consecutive additional clock pulses can be output by writing 1 to the CLO bit after confirming the CLO bit to be 0.
When the RIIC module is in master mode and the slave device is holding the SDAn line low 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 additional SCL output function can be used to output additional clock pulses one by one to make the slave device release the SDAn line from being held low, thus recovering the bus from an unusable state. Release of the SDAn line by the slave device can be monitored by reading the ICCR1.SDAI bit. After confirming release of the SDAn line by the slave device, complete communications by regenerating a stop condition.
Use this function with the ICFER.MALE bit set to 0 (master arbitration-lost detection disabled).

Conditions for using the ICCR1.CLO bit

- When the bus is free (ICCR2.BBSY flag is 0) or in master mode (ICCR2.MST bit is 1 and ICCR2.BBSY flag is 1)
- When the communication device does not hold the SCLn line low

Figure 37.39 shows the operation timing of the additional SCL output function (CLO bit).

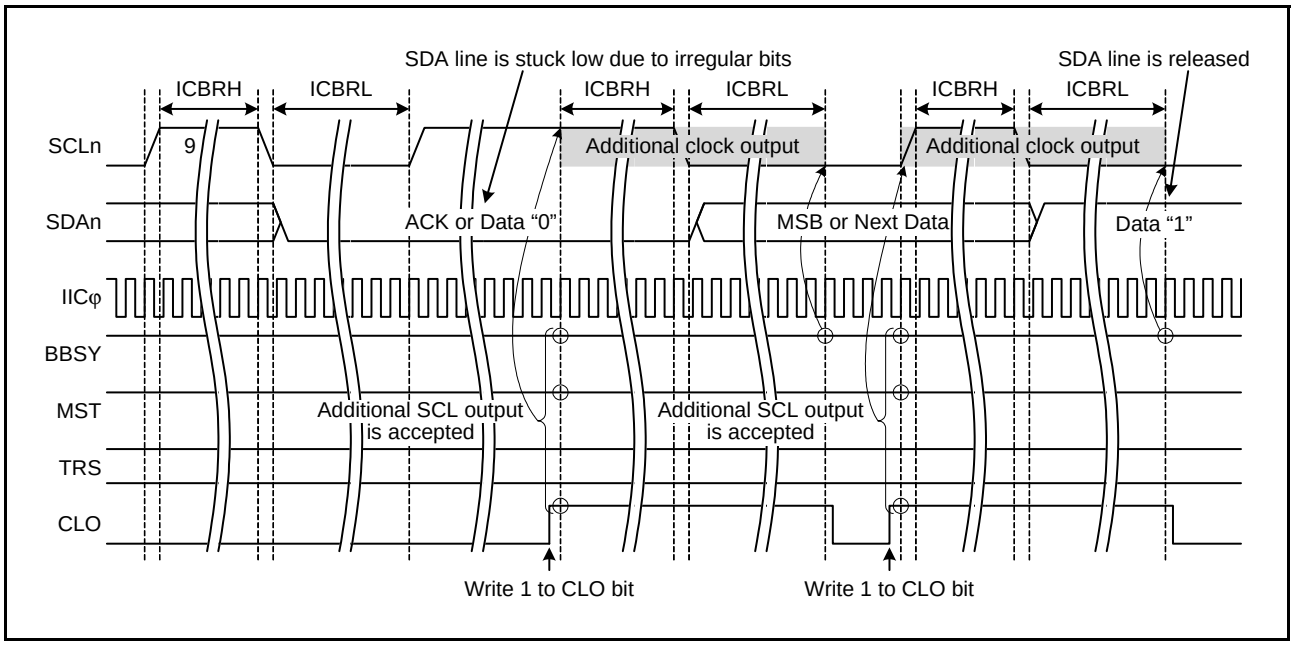


Figure 37.39 Additional SCL Output Function (CLO Bit)

37.11.3 RIIC Reset and Internal Reset

The RIIC module incorporates a function for resetting itself. There are two types of reset. One is referred to as an RIIC reset; this initializes all registers including the ICCR2.BBSY flag. The other is referred to as an internal reset; this releases the RIIC from the slave-address matched state and initializes the internal counter while retaining other settings. After applying a reset, be sure to set the ICCR1.IICRST bit to 0.

Both types of reset are effective for release from bus-hung states because both restore the output state of the SCLn and SDAn pins to the high-impedance state.

Applying a reset during slave operation may lead to a loss of synchronization between the master device clock and the slave device clock, so avoided this where possible. Note that monitoring of the bus state, such as for the presence of a start condition, is not possible during an RIIC reset (bits ICE and IICRST in the ICCR1 register are 01b).

For a detailed description of the RIIC and internal resets, refer to section 37.14, Initialization of Registers and Functions When a Reset is Applied or a Condition is Detected.

37.12 SMBus Operation

The RIIC is available for data communication conforming to the SMBus (Version 2.0). To perform SMBus communication, set the ICMR3.SMBS bit to 1. To use the transfer rate within a range of 10 kbps to 100 kbps of the SMBus specification, set the ICMR1.CKS[2:0] bits, the ICBRH register, and the ICBRL register. In addition, determine the values of the ICMR2.DLCS bit and the ICMR2.SDDL[2:0] bits to meet the data hold time (300 ns (min.)). If the RIIC is used only as a slave device, the transfer rate setting is not necessary, whereas the ICBRL register needs to be set to a value longer than the data setup time (250 ns).
For the SMBus device default address (1100 001b), use one of the slave address registers L0 to L2 (registers SARL0, SARL1, and SARL2), and set the corresponding SARUy.FS bit (7-bit/10-bit address format select) (y = 0 to 2) to 0 (7-bit address format).
When transmitting the UDID (Unique Device Identifier), set the ICFER.SALE bit to 1 to enable the slave arbitration-lost detection function.

37.12.1 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 cumulative clock low extend time (slave device) ($T_{LOW:SEXT}$: 25 ms (max.)) of the SMBus specification.
If the time measured with the MTU or TMR exceeds the detect clock low timeout ($T_{TIMEOUT}$: 25 ms (min.)) of the SMBus specification, the slave device must release the bus by writing 1 to the ICCR1.IICRST bit to apply an internal reset of the RIIC. When an internal reset is applied to the RIIC, it stops driving the SCLn and SDAn pins of the bus and makes the SCLn/SDAn pin outputs high-impedance, thus releasing the bus.

(2) Measuring timeout of master device

The following periods (timeout interval: $T_{LOW:MEXT}$) must be measured for master devices in SMBus communication.

- From start condition to acknowledgment bit
- Between acknowledgment bits
- From acknowledgment 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 transmission end interrupt (TEI) or receive data full interrupt (RXI) of the RIIC. The measured timeout period must be within the cumulative clock low extend time (master device) ($T_{LOW:MEXT}$: 10 ms (max.)) of the SMBus specification, and the total of all $T_{LOW:MEXT}$ from start condition to stop condition must be within $T_{LOW:SEXT}$: 25 ms (max.).
For the ACK receive timing (rising edge of the ninth SCL), monitor the ICSR2.TEND flag in master transmit mode (master transmitter) and the ICSR2.RDRF flag in master receive mode (master receiver). For this reason, perform bytewise transmit operation in master transmit mode, and hold the ICMR3.RDRFS bit 0 until the byte just before reception of the final byte in master receive mode. While the RDRFS bit is 0, the RDRF flag is set to 1 at the rising edge of the ninth SCL.
If the period measured with the MTU or TMR exceeds the cumulative clock low extend time (master device) ($T_{LOW:MEXT}$: 10 ms (max.)) of the SMBus specification or the total of measured periods exceeds the detect clock low timeout ($T_{TIMEOUT}$: 25 ms (min.)) of the SMBus specification, the master device must stop the transaction by generating a stop condition. In master transmit mode, immediately stop the transmit operation (writing data to the ICDRT register).

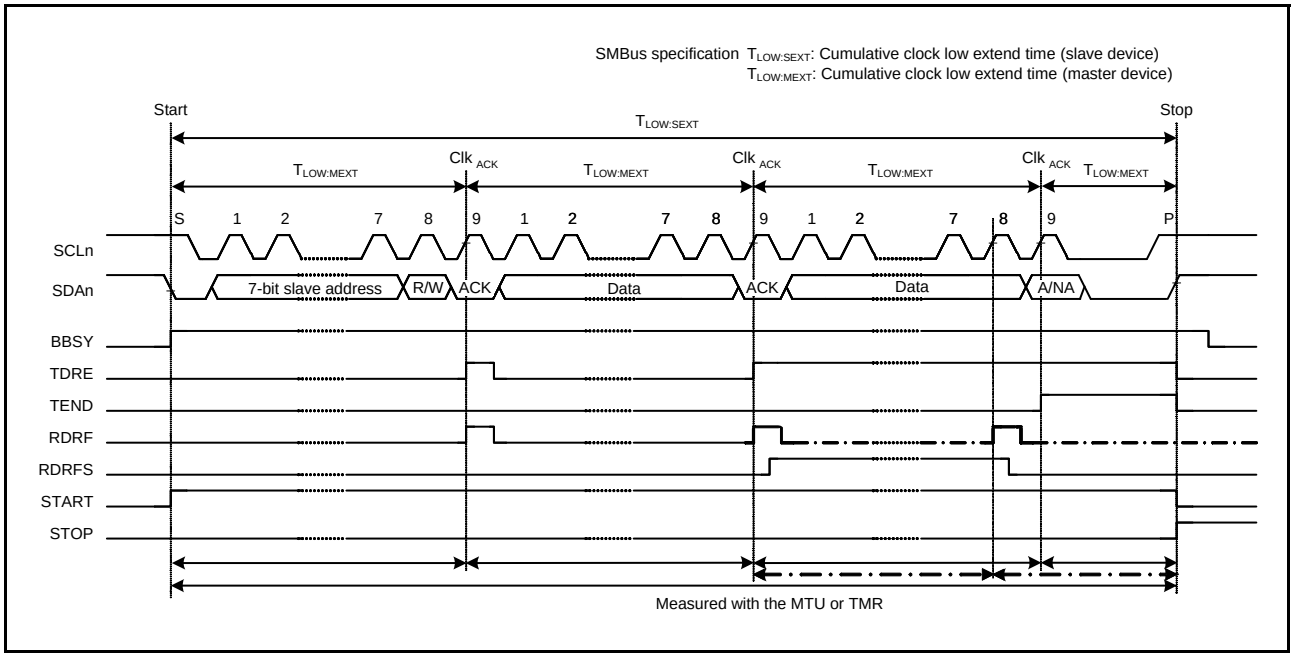


Figure 37.40 SMBus Timeout Measurement

37.12.2 Packet Error Code (PEC)

This MCU incorporates a CRC calculator. The CRC calculator enables transmission of a packet error code (PEC) or checking the received data of the SMBus in data communication of the RIIC. For the CRC generating polynomials of the CRC calculator, refer to section 43, CRC Calculator (CRCA).

The PEC data in master transmit mode can be generated by writing all transmit data to the CRC data input register (CRCDIR) in the CRC calculator.

The PEC data in master receive mode can be checked by writing all receive data to CRCDIR in the CRC calculator and comparing the obtained value in the CRC data output register (CRCDOR) with the received PEC data.

To send ACK or NACK according to the match or mismatch result when the final byte is received as a result of the PEC code check, set the ICMR3.RDRFS bit to 1 before the rising edge of the eighth SCL during reception of the final byte, and hold the SCLn line low at the falling edge of the eighth clock pulse.

37.12.3 SMBus Host Notification Protocol (Notify ARP Master Command)

In communications over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of its own slave address or to request its own slave address from the SMBus host.

For a product of this MCU to operate as an SMBus host (or ARP master), the host address (0001 000b) sent from the slave device must be detected as a slave address, so the RIIC has a function for detecting the host address. To detect the host address as a slave address, set the ICMR3.SMBS bit and the ICSER.HOAE bit to 1. Operation after the host address has been detected is the same as normal slave operation.

37.13 Interrupt Sources

The RIIC generates four types of interrupt requests: communication error or communication event (arbitration-lost detection, NACK detection, timeout detection, start condition detection, and stop condition detection), receive data full, transmit data empty, and transmission end.

Table 37.6 lists details of the several interrupt requests. The receive data full and transmit data empty interrupt requests allow the DTC or DMAC to start data transfer.

Table 37.6 Interrupt Sources

Symbol	Interrupt Source	Interrupt Flag	Start DTC/DMA Transfer	Interrupt Generation Condition
EEI	Communication error/ communication event	AL	Not possible	AL = 1 and ALIE = 1
		NACKF		NACKF = 1 and NAKIE = 1
		TMOF		TMOF = 1 and TMOIE = 1
		START		START = 1 and STIE = 1
		STOP		STOP = 1 and SPIE = 1
RXI*2	Receive data full	RDRF	Possible	RDRF = 1 and RIE = 1
TXI*1	Transmit data empty	TDRE	Possible	TDRE = 1 and TIE = 1
TEI*3	Transmission end	TEND	Not possible	TEND = 1 and TEIE = 1

Note: There is a delay time between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. Thus, when an interrupt flag has been cleared or an interrupt request has been masked, read the relevant flag again to check whether clearing or masking has been completed, and then return from interrupt handling. Returning from interrupt handling without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.

Note 1. Because TXI is an edge-detected interrupted, it does not require clearing. Furthermore, the ICSR2.TDRE flag (a condition for TXI) is automatically set to 0 when data for transmission are written to the ICDRT register or a stop condition is detected (ICSR2.STOP flag is 1).

Note 2. Because RXI is an edge-detected interrupted, it does not require clearing. Furthermore, the ICSR2.RDRF flag (a condition for RXI) is automatically set to 0 when data are read from the ICDRR register.

Note 3. When using the TEI interrupt, clear the ICSR2.TEND flag in the TEI interrupt handling.
Note that the ICSR2.TEND flag is automatically set to 0 when data for transmission are written to the ICDRT register or a stop condition is detected (ICSR2.STOP flag is 1).

Clear the each flag or mask the interrupt request during interrupt handling.

37.13.1 Buffer Operation for TXI and RXI Interrupts

If the conditions for generating a TXI and RXI interrupt are satisfied while the corresponding ICU.IRn.IR flag is 1, the interrupt request is not output for the ICU but retained internally (the capacity for internal retention is one request per source).

An interrupt request that was being retained internally is output to the ICU when the value of the IR flag becomes 0.

Internally retained interrupt requests are automatically cleared under normal conditions of usage.

Internally retained interrupt requests can also be cleared by writing 0 to the corresponding interrupt enable bit in the ICIER register.

37.14 Initialization of Registers and Functions When a Reset is Applied or a Condition is Detected

The RIIC can be reset by MCU reset, RIIC reset, and internal reset functions. Table 37.7 lists the reset states of registers and functions when a reset is applied or a condition is detected.

Table 37.7 Reset States of Registers and Functions When a Reset is Applied or a Condition is Detected

		MCU Reset	RIIC Reset (ICE = 0, IICRST = 1)	Internal Reset (ICE = 1, IICRST = 1)	Start Condition/ Restart Condition Detection	Stop Condition Detection
ICCR1	SDAO, SCLO	To be reset	To be reset	To be reset	Retained	Retained
	IICRST, ICE		Retained	Retained		
	Others		To be reset			
ICCR2	ST, RS	To be reset	To be reset	To be reset	To be reset	Retained
	SP					To be reset
	TRS				See note 1	
	MST				See note 1	
	BBSY			Retained	Becomes 1	
ICMR1	BC[2:0]	To be reset	To be reset	To be reset	To be reset	Retained
	Others			Retained	Retained	
ICMR2		To be reset	To be reset	Retained	Retained	Retained
ICMR3	ACKBT	To be reset	To be reset	Retained	Retained	To be reset
	Others					Retained
ICFER		To be reset	To be reset	Retained	Retained	Retained
ICSER		To be reset	To be reset	Retained	Retained	Retained
ICIER		To be reset	To be reset	Retained	Retained	Retained
ICSR1		To be reset	To be reset	To be reset	Retained	To be reset
ICSR2	START	To be reset	To be reset	To be reset	Becomes 1	To be reset
	STOP				Retained	Becomes 1
	TEND					To be reset
	TDRE				See note 1	
	Others				Retained	Retained
SARL0, SARL1, SARL2, SARU0, SARU1, SARU2		To be reset	To be reset	Retained	Retained	Retained
ICBRH, ICBRL		To be reset	To be reset	Retained	Retained	Retained
ICDRT		To be reset	To be reset	Retained	Retained	Retained
ICDRR		To be reset	To be reset	Retained	Retained	Retained
ICDRS		To be reset	To be reset	To be reset	Retained	Retained
Timeout function		To be reset	To be reset	To be reset	Operation	Operation
Bus free time measurement		To be reset	To be reset	Operation	Operation	Operation

Note 1. This bit is not reset. This bit becomes 0 or 1 in accordance with the conditions.

37.15 Event Link Function (Output)

The RIIC0 handles event output for the event link controller (ELC) corresponding to the following sources.

- Communication error/communication event
- Receive data full
- Transmit data empty
- Transmission end

37.15.1 Interrupt Handling and Event Linking

The RIIC has four types of interrupts: communication error or communication event (arbitration-lost detection, NACK detection, timeout detection, start condition detection, or stop condition detection), receive data full, transmit data empty, and transmission end. Each of these has an enable bit to control enabling and disabling of the interrupt signal. An interrupt request signal is output for the ICU when an interrupt source condition is satisfied while the setting of the corresponding enable bit is enabled.
The corresponding event signals are sent to other modules via the ELC when the interrupt source conditions are satisfied, regardless of the settings of the interrupt enable bits.
For details on interrupt sources, see Table 37.6.

37.16 Usage Notes

37.16.1 Setting Module Stop Function

Module stop state can be entered or released using module stop control register B (MSTPCRB) or module stop control register C (MSTPCRC). The initial setting is for operation of the RIIC to be stopped. RIIC register access is enabled by releasing the module stop state.
For details on module stop control registers B and C, refer to section 11, Low Power Consumption.

37.16.2 Notes on Starting Transfer

If the IR flag corresponding to the RIIC interrupt is 1 when transfer is started (ICCR1.ICE bit is 1), follow the procedure below to clear interrupts before enabling operations. Starting transfer with the IR flag set to 1 while the ICCR1.ICE bit is 1 leads to an interrupt request being internally retained after transfer starts, and this can lead to unanticipated behavior of the IR flag.

1. Confirm that the ICCR1.ICE bit is 0.
2. Set the relevant interrupt enable bits (ICIER.TIE, etc.) on the peripheral function side to 0.
3. Read the relevant interrupt enable bits (ICIER.TIE, etc.) on the peripheral function side and confirm that its value is 0.
4. Set the IR flag to 0.

38. High-Speed I2C-bus Interface (RIICHS)

This MCU has a single-channel I2C-bus interface (RIICHS0).
The RIICHS module conforms with the NXP I2C-bus (Inter-IC bus) interface and provides a subset of its functions.

38.1 Overview

Table 38.1 lists the specifications of the RIICHS and Figure 38.1 shows a block diagram of the RIICHS. Table 38.2 lists the I/O pins of the RIICHS.

Table 38.1 RIICHS Specifications (1/2)

Item	Specification
Operation mode	Four • Master transmit mode, master receive mode, slave transmit mode, and slave receive mode
Communication format Transfer rate	• I2C-bus format - Standard-mode (Sm): 0 to 100 kbps - Fast-mode (Fm): 0 to 400 kbps - Fast-mode Plus (Fm+): 0 to 1 Mbps - High-speed mode (Hs-mode): 0 to 3.4 Mbps • SMBus format: 10 to 100 kbps • Master mode or slave mode selectable • Automatic securing of the various setup times, hold times, and bus-free times for the transfer rate
Serial clock (SCL)	• For master operation, the duty cycle of the SCL is selectable in the range from 4 to 96%.
Generating and detecting conditions	START, repeated START, and STOP conditions are automatically generated. START conditions (including repeated START conditions) and STOP conditions are detectable.
Address detection	• Slave address (static address) (max 3 address) • 7-bit and 10-bit address formats are supported. • General call address • Hs-mode master code • Device ID • Host address
Acknowledgment	• For transmission, the acknowledgment bit is automatically loaded. Transfer of the next data for transmission can be automatically suspended on detection of a not-acknowledge signal. • For reception, the acknowledgment bit is automatically transmitted. If a wait between the eighth and ninth clock pulses has been selected, software control of the acknowledgment in response to the received data is possible.
Wait function (clock stretch)	• During reception, cycles of waiting by holding the SCL line low can be inserted at the following two types of timing: Waiting between the eighth and ninth clock pulses Waiting between the ninth clock pulse and the first clock pulse of the next byte
SDA output delay function	It is possible to delay the change timing of the output of the transmission data bits and the acknowledgment bit relative to the falling edge of SCL.
Arbitration	• For multi-master operation Clock synchronization for the SCL line in cases of conflict with the SCL signal from another master is possible. When generating the START condition would create conflict on the bus, loss in arbitration is detected by testing for non-matching between the internal data level and the actual level on the SDA line. During master operation, loss in arbitration is detected by testing for non-matching between the actual level on the SDA line and the internal data level. • Loss in arbitration due to detection of the START condition while the bus is busy is detectable (to prevent the generating of double START conditions). • Loss in arbitration in transfer of a not-acknowledge signal due to the internal signal (NACK) and the actual level on the SDA line not matching is detectable. • Loss in arbitration due to non-matching of internal data level and the actual level on the SDA line is detectable in slave transmission.
Timeout function	The internal timeout function is capable of detecting long-interval stop of the SCL.
Noise filter	• Analog noise filter • Digital noise filter

Table 38.1 RIICHS Specifications (2/2)

Item	Specification
Interrupt source	Four sources: • Transmit data empty (TXI) • Receive data full (RXI) • Transmission end (TEI) • Communication error or communication event (EEI) - START condition detection (including repeated START condition) - STOP condition detection - NACK detection - Arbitration lost - Timeout detection
Event link function (output)	• Transmit data empty • Receive data full • Transmission end • Communication error or communication event
Low power consumption function	Module stop state can be set.

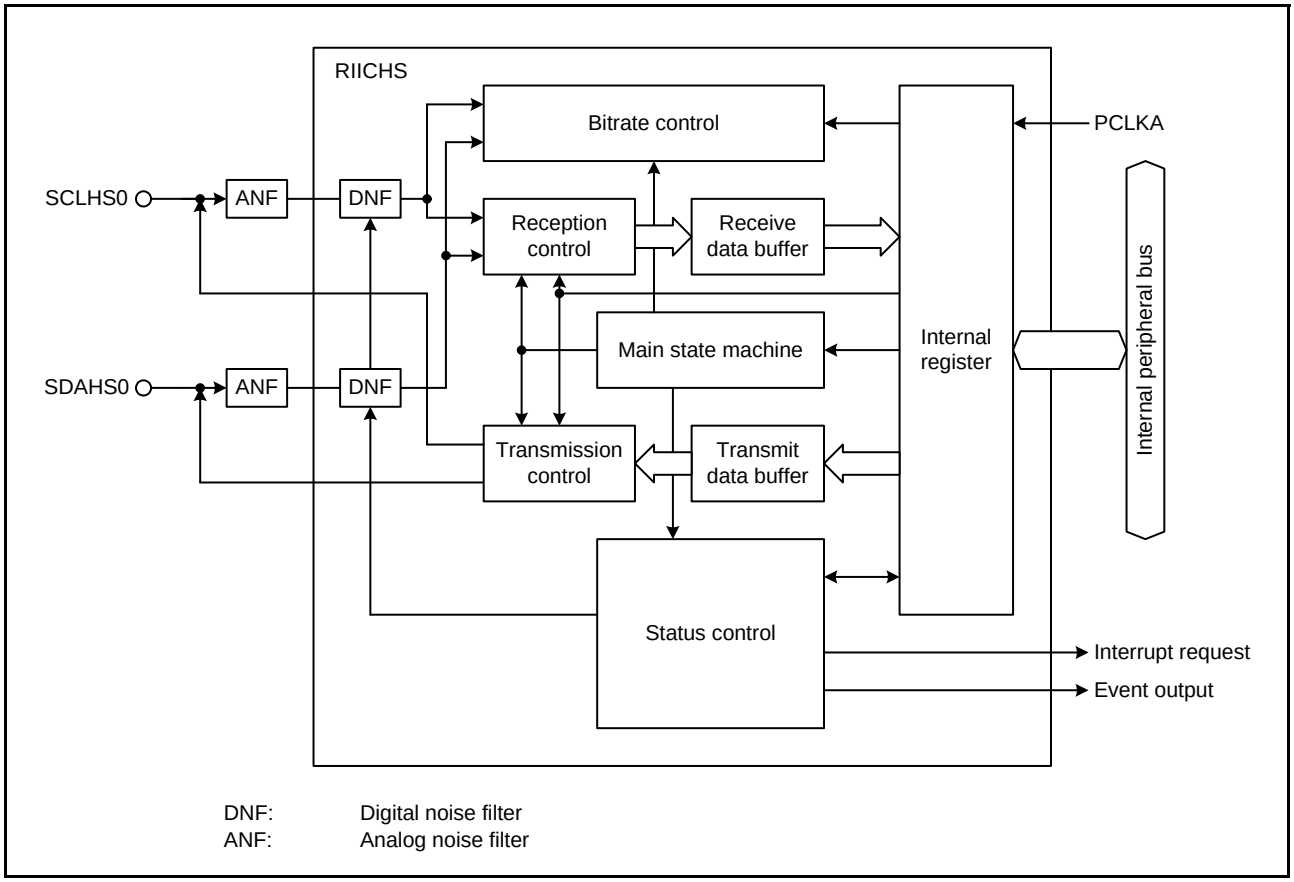


Figure 38.1 RIICHS Block Diagram

Table 38.2 RIICHS Pin Configuration

Channel	Pin Name	I/O	Function
RIICHS0	SCLHS0	I/O	Serial clock I/O pin
	SDAHS0	I/O	Serial data I/O pin

38.2 Register Descriptions

38.2.1 Control Register (ICCR)

Address(es): RIICHS0.ICCR 000E C014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ICE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Bit Name	Description	R/W
b30 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b31	ICE	Bus Interface Enable	0: The I2C bus operation is disabled. (SCLHS0 and SDAHS0 pins in inactive state) 1: The I2C bus operation is enabled. (SCLHS0 and SDAHS0 pins in active state)	R/W

ICE Bit (Bus Interface Enable)

Enables or disables the operation on the I2C-bus.

When using RIICHS, set this bit to 1. When the ICE bit is 1, the SCLHS0 and SDAHS0 pins are active.

Set this bit to 0 when RIICHS is not used. When the ICE bit is 0, the SCLHS0 and SDAHS0 pins are inactive.

When this bit is read as 0, it indicates that RIICHS bus operation is prohibited.

38.2.2 Reset Control Register (ICRCR)

Address(es): RIICHS0.ICRCR 000E C020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ISRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MRST
Value after 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	MRST	Module Reset	0: All registers and internal circuits are released from reset. 1: All registers and internal circuits are reset.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	ISRST	Internal Status Reset	0: ICBCR.BC[4:0] bits, ICSSR register, ICSR2 register, and internal circuits are released from reset. 1: ICBCR.BC[4:0] bits, ICSSR register, ICSR2 register, and internal circuits are reset.	R/W
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R

MRST Bit (Module Reset)

When setting this bit to 1, the RIICHS registers and the internal circuits are reset and the I2C bus operation is disabled. This bit is automatically set to 0 upon the RIICHS reset completion.

Note: If this bit is 1, do not rewrite the values of the other bits.

ISRST Bit (Internal Status Reset)

When this bit is set to 1, some registers and internal circuits are reset. Refer to section 38.15, Reset Description for detail on the registers.

38.2.3 Operating Mode Monitor Register (ICMMR)

Address(es): RIICHS0.ICMMR 000E C024h

	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
	—	—	—	—	—	—	—	—	MSWP	—	—	TRS	—	MST	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b2	MST	Master/Slave Mode Flag	0: Slave mode 1: Master mode	R/W*1
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b4	TRS	Transmit/Receive Mode Flag	0: Receive mode 1: Transmit mode	R
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b7	MSWP	MST Write Protect	0: Writing to the MST flag is disabled. 1: Writing to the MST flag is enabled (when writing simultaneously with the value of the target bit).	W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. When setting a value to the MST flag, set the MSWP bit to 1 at the same time.

MST Flag (Master/Slave Mode Flag)

[Setting conditions]

- When 1 is written to the MSWP bit and the MST flag at the same time.
- When a START condition is generated normally according to the START condition generation request (when a START condition is detected with the ICCGR.ST bit set to 1).

[Clearing conditions]

- When the RIICHS is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.
- When 1 is written to the MSWP bit and 0 is written to the MST flag at the same time.
- When a STOP condition is detected.
- When the ICSR2.AL flag becomes 1 (arbitration is lost).

TRS Flag (Transmit/Receive Mode Flag)

This bit indicates transmit or receive mode.

The RIICHS is in receive mode when the TRS flag is set to 0 and is in transmit mode when the bit is set to 1.

Combination of this bit and the MST flag indicates the operating mode of the RIICHS.

The value of TRS flag is automatically changed to 1 for transmit mode or 0 for receive mode by generating or detection of a START condition and setting of the R/W# bit.

[Setting conditions]

- When a START condition is generated normally according to the START condition generation request (when a START condition is detected with the ICCGR.ST bit set to 1).
- When a repeated START condition is generated normally according to the repeated START condition generation

request (when a repeated START condition is detected with the ICCGR.RS bit set to 1).

- When the R/W# bit added to the slave address is set to 0 in master mode.
- When the address received in slave mode matches the address enabled in ICSCR register, with the R/W# bit set to 1.

[Clearing conditions]

- When a STOP condition is detected.
- When the ICSR2.AL flag becomes 1 (arbitration is lost).
- 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 ICSCR register when the value of the received R/W# bit is 0 (including cases where the received address is the general call address).
- In slave mode, a repeated START condition is detected (a START condition is detected with ICBSR.BFREE flag = 0 and MST flag = 0).
- When 1 is written to ICRCR.MRST bit or ICRCR.ISRST bit to reset the RIICHS or the internal status.

MSWP Bit (MST Write Protect)

This bit is used to rewrite the MST flag.
When setting a value to the MST flag, set this bit to 1 at the same time.
This bit is automatically set to 0. The read value is 0.

38.2.4 Function Enable Register (ICFER)

Address(es): RIICHS0.ICFER 000E C060h

	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
	HSME	FMPE	—	SMBS	—	—	—	SCLE	—	—	—	—	—	SALE	NALE	MALE
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	MALE	Master Arbitration-Lost Detection Enable	0: Master arbitration-lost detection is disabled. (Disables the arbitration-lost detection function and does not clear the ICMMR.MST and ICMMR.TRS flags automatically when arbitration is lost.) 1: Master arbitration-lost detection is enabled. (Enables the arbitration-lost detection function and clears the ICMMR.MST and ICMMR.TRS flags automatically when arbitration is lost.)	R/W
b1	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
b2	SALE	Slave Arbitration-Lost Detection Enable	0: Slave arbitration-lost detection is disabled. 1: Slave arbitration-lost detection is enabled.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	SCLE	SCL Synchronization Enable	0: SCL synchronization is disabled. 1: SCL synchronization is enabled.	R/W
b11 to b9	—	Reserved	These bits are read as 0 and cannot be modified.	R
b12	SMBS	SMBus/I2C-bus Selection	0: The I2C-bus select 1: The SMBus select	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R
b14	FMPE	Fast-mode Plus Enable	0: No Fm+ slope control circuit is used for the SCLHS0 pin and SDAHS0 pin. 1: An Fm+ slope control circuit is used for the SCLHS0 pin and SDAHS0 pin.	R/W
b15	HSME	Hs-mode Enable	0: High speed mode is disabled. 1: High speed mode is enabled.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

MALE Bit (Master Arbitration-Lost Detection Enable)

This bit uses 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 uses 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 uses 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).

SCLE Bit (SCL Synchronization Enable)

This bit uses to specify whether to synchronize the SCL output with the SCL input. Normally, set this bit to 1.
When the SCLE bit set to 0 (SCL synchronization is disabled), the RIICHS does not synchronize the SCL output with the SCL input. In this setting, the RIICHS outputs the clock with the transfer rate set in ICBRH and ICBRL regardless of the SCLHS0 line state. For this reason, if the load of the I2C bus line is much larger than the specification value or if the SCL output overlaps in multiple masters, the short-cycle SCL that does not meet the specification may be output. When the SCL synchronization is not, it also affects the generation of a START condition, repeated START condition, and STOP condition, and the continuous output of additional SCL.
This bit must not be set to 0 except for checking the output of the set transfer rate.

FMPE Bit (Fast-mode Plus Enable)

This bit uses to specify whether to use a slope control circuit for Fast-mode Plus (Fm+).
When this bit 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 set 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. Set this bit to 0 when using the transmission rate at other rates (up to 100 kbps (Sm), up to 400 kbps (Fm)) or for SMBus (10 to 100 kbps). When performing Hs-mode communication, set this bit according to the mode when sending the Hs-mode master code.

HSME Bit (Hs-mode Enable)

This bit is used for communicating in Hs-mode.
When this bit is set to 1, the Hs-mode master code is recognized and Hs-mode communication is possible.
After the START condition is detected, if Hs-mode master code (0000 1XXXb) transmission is recognized, Hs-mode communication starts from repeated START condition after receiving the NACK response.
It communicates at the bit rate set in the ICFBR register until the NACK response, and automatically switches from repeated START condition issuance after receiving the NACK response to the bit rate set in the ICHBR register.
Hs-mode continues until a STOP condition is detected.
When the STOP condition is detected, the bit rate is automatically switched to the bit rate set in the ICFBR register.

Note: When this bit is set to 1, the ICSR2.NACKF bit will not be set even if a NACK response is received after sending the Hs-mode master code.

38.2.5 Slave Mode Control Register (ICSCR)

Address(es): RIICHS0.ICSCR 000E C064h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	SAR2E	SAR1E	SAR0E
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	HOAE	—	—	—	—	—	—	—	—	DIDE	HSMCE	—	—	—	—	GCAE
Value after 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	GCAE	General Call Address Detection Enable	0: General call address detection is disabled. 1: General call address detection is enabled.	R/W
b4 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b5	HSMCE	Hs-mode Master Code Detection Enable	0: Hs-mode master code detection is disabled. 1: Hs-mode master code detection is enabled.	R/W
b6	DIDE	Device-ID Address Detection Enable	0: Device-ID address detection is disabled. 1: Device-ID address detection is enabled.	R/W
b14 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R
b15	HOAE	Host Address Detection Enable	0: Host address detection is disabled. 1: Host address detection is enabled.	R/W
b16	SAR0E	Slave Address 0 Detection Enable	0: The value of the SAMR0.SVA[9:0] bits is disabled. 1: The value of the SAMR0.SVA[9:0] bits is enabled.	R/W
b17	SAR1E	Slave Address 1 Detection Enable	0: The value of the SAMR1.SVA[9:0] bits is disabled. 1: The value of the SAMR1.SVA[9:0] bits is enabled.	R/W
b18	SAR2E	Slave Address 2 Detection Enable	0: The value of the SAMR2.SVA[9:0] bits is disabled. 1: The value of the SAMR2.SVA[9:0] bits is enabled.	R/W
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R

GCAE Bit (General Call Address Detection Enable)

This bit is used to specify whether to ignore the general call address (0000 000b + 0 (write): All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the RIICHS recognizes the received slave address as the general call address independently of the slave addresses set in SAMRy.SVA[9:0] bits and performs data receive operation.

When this bit is set to 0, the received slave address is ignored even if it matches the general call address.

HSMCE Bit (Hs-mode Master Code Detection Enable)

This bit is used to specify whether to recognize and execute the Hs-mode master code (0000 1XXXb) is received in the first byte after a START condition is detected.

When this bit is set to 1, if the received first byte matches the Hs-mode master code, RIICHS recognizes that the Hs-mode master code has been received.

The first byte after repeated START condition after NACK response to Hs-mode master code is recognized as a slave address and compared with the slave address set in the SAMRy.SVA[9:0] bits (y = 0 to 2).

If the addresses match, the transmission/reception operation continues according to the R/W# bit value.

Hs-mode continues until a STOP condition is detected.

When this bit is set to 0, RIICHS will ignore the pattern until a STOP condition is detected, even if it matches the Hs-mode master code.

When this bit is set to 1, set the ICCSCR.WAITAE bit to 0 and ICCSCR.WAITRE bit to 1.

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 repeated START condition is detected.
When this bit is set to 1, if the received first frame matches the device ID, the RIICHS recognizes that the device-ID address has been received. When the following R/W# bit is 0 (write), the RIICHS recognizes the second and the following frames as slave addresses and continues the receive operation.
When this bit is set to 0, the RIICHS 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, refer to section 38.7.3, Device-ID Address Detection.

HOAE Bit (Host Address Detection Enable)

This bit uses to specify whether to ignore received host address (0001 000b) when the ICFER.SMBS bit is 1.
When this bit set to 1 while the ICFER.SMBS bit is 1, if the received slave address matches the host address, the RIICHS recognizes the received slave address as the host address independently of the slave addresses set in SAMRy.SVA[9:0] bits and performs the receive operation.
When the ICFER.SMBS bit or the HOAE bit is set to 0, the received slave address is ignored even if it matches the host address.

SARyE Bit (Slave Address y Detection Enable) (y = 0 to 2)

This bit is used to enable or disable the received slave address and the slave address set in SAMRy.SVA[9:0] bits.
When this bit is set to 1, the slave address set in SAMRy.SVA[9:0] bits is enabled and is compared with the received slave address.
When this bit is set to 0, the slave address set in SAMRy.SVA[9:0] bits is disabled and is ignored even if it matches the received slave address.

38.2.6 Reference Clock Control Register (ICRCCR)

Address(es): RIICHS0.ICRCCR 000E C070h

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	CKS[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	CKS[2:0]	Internal Reference Clock Select	Selects the internal reference clock source (IICφ) for the RIICHS. b2 b0 0 0 0: PCLKA/1 0 0 1: PCLKA/2 0 1 0: PCLKA/4 0 1 1: PCLKA/8 1 0 0: PCLKA/16 1 0 1: PCLKA/32 1 1 0: PCLKA/64 1 1 1: PCLKA/128	R/W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

38.2.7 F/S-Mode Bit Rate Register (ICFBR)

Address(es): RIICHS0.ICFBR 000E C074h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DBL	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	LOW[7:0]	Low Width Setting	Count value for the low period of SCL.	R/W
b15 to b8	HIGH[7:0]	High Width Setting	Count value for the high period of SCL.	R/W
b30 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R
b31	DBL	High/Low Width Doubling	0: Set the high and low widths to the set value. 1: Double the high and low widths of the set values.	R/W

This register is used to specify the bit rate other than Hs-mode.

LOW[7:0] Bits (Low Width Setting)

LOW[7:0] bits are used to set the low period of SCL.

The RIICHS counts the low period with the internal reference clock source (IICφ) specified by the ICRCCR.CKS[2:0] bits.

It also works to generate the data setup time for automatic SCL low-hold operation (refer to section 38.8, Automatic Low-Hold Function for SCL); when the RIICHS is used in slave mode, these bits need to be set to a value longer than the data setup time*1.

If the digital noise filter is enabled (the ICICR.NFE bit is 1), set the LOW[7:0] bits to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICICR.NF[3:0] bits.

Note 1. Data setup time (tSU;DAT)
250 ns (up to 100 kbps: Standard-mode (Sm))
100 ns (up to 400 kbps: Fast-mode (Fm))
50 ns (up to 1 Mbps: Fast-mode Plus (Fm+))

HIGH[7:0] Bits (High Width Setting)

HIGH[7:0] bits use to set the high period of SCL.

HIGH[7:0] bits are valid in master mode. If the RIICHS is used only in I2C slave mode, these bits need not to set the high period.

The RIICHS counts the high period with the internal reference clock source (IICφ) specified by the ICRCCR.CKS[2:0] bits.

If the digital noise filter is enabled (the ICICR.NFE bit is 1), set the HIGH[7:0] bits to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICICR.NF[3:0] bits.

The transfer rate and the duty cycle are calculated using the following expression.

tLOW = (LOW[7:0] + NF[3:0] +1) / IICφ
tHIGH = (HIGH[7:0] + NF[3:0] +1) / IICφ
Transfer rate = 1 / (tLOW + tHIGH + tr + tf)
Duty cycle = (tr + tHIGH) / (tLOW + tHIGH + tr + tf)

tLOW: Low period of the SCL
tHIGH: high period of the SCL
tr: Rise time of the SCL signal*2
tf: Fall time of the SCL signal*2

Note 2. The SCL line rise time [tr] and SCL line fall time [tf] depend on the total bus line capacitance [Cb] and the pull-up resistor [Rp]. For details, refer to the I2C-bus standard from NXP Semiconductors.

Table 38.3 Example of the ICFBR Register Settings for Transfer Rate (PCLKA = 120 MHz)

Transfer Rate (kbps)	CKS[2:0] Bits	HIGH[7:0] Bits	LOW[7:0] Bits	NFE Bit	NF[3:0] Bits
10	101b	122 (7Ah)	247 (F7h)	0	—
	101b	121 (79h)	246 (F6h)	1	0001b
50	011b	91 (5Bh)	188 (BCh)	0	—
	011b	90 (5Ah)	187 (BBh)	1	0001b
100	010b	108 (6Ch)	151 (97h)	0	—
	010b	107 (6Bh)	150 (96h)	1	0001b
400	000b	68 (44h)	158 (9Eh)	0	—
	000b	67 (43h)	157 (9Dh)	1	0001b
	001b	34 (22h)	78 (4Eh)	0	—
	001b	33 (21h)	77 (4Dh)	1	0001b
1000	000b	27 (1Bh)	63 (3Fh)	0	—
	000b	26 (1Ah)	62 (3Eh)	1	0001b
	001b	13 (0Dh)	31 (1Fh)	0	—
	001b	12 (0Ch)	30 (1Eh)	1	0001b

Rise time of SCL line (tr)
- 100 kbps or less (Sm): 1000 ns
- 400 kbps (Fm): 300 ns
- 1 Mbps (Fm+): 120 ns

Fall time of SCL line (tf)
- 400 kbps (Sm/Fm): 300 ns
- 1 Mbps (Fm+): 120 ns

38.2.8 Hs-Mode Bit Rate Register (ICHBR)

Address(es): RIICHS0.ICHBR 000E C078h

	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
	HIGH[7:0]								LOW[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	LOW[7:0]	Low Width Setting	Count value for the low period of SCL.	R/W
b15 to b8	HIGH[7:0]	High Width Setting	Count value for the high period of SCL.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

This register is used to specify the bit rate in Hs-mode.

LOW[7:0] Bits (Low Width Setting)

LOW[7:0] bits are used to set the low period of SCL.

The RIICHS counts the low period with the internal reference clock source (IICφ) specified by the ICRCCR.CKS[2:0] bits.

It also works to generate the data setup time for automatic SCL low-hold operation (refer to section 36.8, Automatic Low-Hold Function for SCL); when the RIICHS is used in slave mode, these bits need to be set to a value longer than the data setup time*1.

If the digital noise filter is enabled (the ICICR.NFE bit is 1), set the LOW[7:0] bits to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICICR.NF[3:0] bits.

Note 1. Data setup time (tSU;DAT)
10 ns (up to 3.4 Mbps: Hs-mode (Hs))

HIGH[7:0] Bits (High Width Setting)

HIGH[7:0] bits use to set the high period of SCL.

HIGH[7:0] bits are valid in master mode. If the RIICHS is used only in I2C slave mode, these bits need not to set the high period.

The RIICHS counts the high period with the internal reference clock source (IICφ) specified by the ICRCCR.CKS[2:0] bits.

If the digital noise filter is enabled (the ICICR.NFE bit is 1), set the HIGH[7:0] bits to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICICR.NF[3:0] bits.

The transfer rate and the duty cycle are calculated using the following expression.

tLOW = (LOW[7:0] + NF[3:2] +1) / IICφ
tHIGH = (HIGH[7:0] + NF[3:2] +1) / IICφ
Transfer rate = 1 / (tLOW + tHIGH + tr + tf)
Duty cycle = (tr + tHIGH) / (tLOW + tHIGH + tr + tf)

tLOW: Low period of the SCL
tHIGH: high period of the SCL
tr: Rise time of the SCL signal*2
tf: Fall time of the SCL signal*2

Note 2. The SCL line rise time [tr] and SCL line fall time [tf] depend on the total bus line capacitance [Cb] and the pull-up resistor [Rp]. For details, refer to the I2C-bus standard from NXP Semiconductors.

38.2.9 Bus Free Time Setting Register (ICBFTR)

Address(es): RIICHS0.ICBFTR 000E C07Ch

	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
b8 to b0	—	—	The count value is a period for detecting the bus free condition.	R/W
b31 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R

This register is used to specify the time from when the STOP condition is detected until the ICBSR.BFREE flag is set to 1. The number set in this register is counted using IICφ as the count source.
Set a value that satisfies the t_{BUF} (bus free time between a STOP and START condition) specified in the I2C-bus specification.

38.2.10 Output Signal Control Register (ICOCR)

Address(es): RIICHS0.ICOCR 000E C088h

	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
	DLCS	—	—	—	—	SDDL[2:0]			—	—	—	CLO	—	SOWP	SCLO	SDAO
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	SDAO	SDA Output Control	0: The RIICHS drives the SDAHS0 pin low. 1: The RIICHS releases the SDAHS0 pin.	R/W
b1	SCLO	SCL Output Control	0: The RIICHS drives the SCLHS0 pin low. 1: The RIICHS releases the SCLHS0 pin. (High level output is achieved through an external pull-up resistor.)	R/W
b2	SOWP	SCLO/SDAO Write Protect	0: Writing to the SCLO and SDAO bits is disabled. 1: Writing to the SCLO and SDAO bits is enabled (when writing simultaneously with the value of the target bit). This bit is read as 0.	W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R
b4	CLO	Additional SCL Output	0: Does not output an additional SCL (default). 1: Outputs an additional SCL. (The CLO bit is cleared automatically after one clock pulse is output.)	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b10 to b8	SDDL[2:0]	SDA Output Delay Setting	• When ICOCR.DLCS bit = 0 (IICφ) b10 b8 0 0 0: No output delay 0 0 1: 1 IICφ cycle 0 1 0: 2 IICφ cycles 0 1 1: 3 IICφ cycles 1 0 0: 4 IICφ cycles 1 0 1: 5 IICφ cycles 1 1 0: 6 IICφ cycles 1 1 1: 7 IICφ cycles • When ICOCR.DLCS bit = 1 (IICφ/2) b10 b8 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
b14 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R
b15	DLCS	SDA Output Delay Clock Source Select	0: The internal reference clock (IICφ) is selected as the clock source of the SDA output delay counter. 1: The internal reference clock divided by 2 (IICφ/2) is selected as the clock source of the SDA output delay counter.*1	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. The setting DLCS = 1 (IICφ/2) only becomes valid when SCL is at the low level. When SCL is at the high level, the setting DLCS = 1 becomes invalid and the clock source becomes the internal reference clock (IICφ).

SDAO Bit (SDA Output Control) and SCLO Bit (SCL Output Control)

These bits are used to directly control the SDAHS0 and SCLHS0 signals output from the RIICHS.
When writing to these bits, also write 1 to the SOWP bit at the same time.
The result of setting these bits is input to the RIICHS via the input buffer. When slave mode is selected, a START condition may be detected and the bus may be released depending on the bit settings.
Do not rewrite these bits during a START condition, STOP condition, repeated START condition, or during transmission or reception. Operation after rewriting under the above conditions is not guaranteed.

CLO Bit (Additional SCL Output)

This bit is used to output an additional SCL for debugging or error processing.
Normally, set the bit to 0. Setting the bit to 1 in a normal communication state causes a communication error.
For details on this function, refer to section 38.11.2, Additional SCL Output Function.

38.2.11 Input Signal Control Register (ICICR)

Address(es): RIICHS0.ICICR 000E C08Ch

	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
	—	—	—	—	—	—	—	—	—	—	—	NFE	NF[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	NF[3:0]	Noise Filter Stage Select	Except in Hs-mode b3 b0 0 0 0 0: Noise of up to 1 IICφ cycle is filtered out (single-stage filter). 0 0 0 1: Noise of up to 2 IICφ cycles is filtered out (2-stage filter). 0 0 1 0: Noise of up to 3 IICφ cycles is filtered out (3-stage filter). : : 1 1 1 1: Noise of up to 16 IICφ cycles is filtered out (16-stage filter). In Hs-mode b3 b2 0 0: Noise of up to 1 IICφ cycle is filtered out (single-stage filter). 0 1: Noise of up to 2 IICφ cycles is filtered out (2-stage filter). 1 0: Noise of up to 3 IICφ cycles is filtered out (3-stage filter). 1 1: Noise of up to 4 IICφ cycles is filtered out (4-stage filter). The lower 2 bits (b1 and b0) are ignored.	R/W
b4	NFE	Digital Noise Filter Enable	0: Digital noise filter circuit is not used. 1: Digital noise filter circuit is used.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R
b7, b6	—	Reserved	These bits are read as 1. The write value should be 1.	R
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

NF[3:0] Bits (Noise Filter Stage Select)

These bits are used to select the number of stages in the digital noise filter.

For details on the digital noise filter function, refer to section 38.6, Digital Noise Filters.

Note: Set the noise range to be filtered out by the noise filter within a range less than the SCLHS0 line high period or low 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 serial clock is regarded as noise by the noise filter function of the RIICHS, which may prevent the RIICHS from operating normally.

38.2.12 Timeout Control Register (ICTOR)

Address(es): RIICHS0.ICTOR 000E C090h

	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
	—	—	—	—	—	—	—	—	TMOM[1:0]		TMOH	TMOL	—	—	TMOS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TMOS[1:0]	Timeout Detection Time Select	b1 b0 0 0: Timeout counter is 16 bits (up to 65536 counts) 0 1: Timeout counter is 14 bits (up to 16384 counts) 1 0: Timeout counter is 8 bits (up to 256 counts) 1 1: Timeout counter is 6 bits (up to 64 counts)	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	TMOL	Timeout L Count Control	0: Count is disabled while the SCLHS0 line is low. 1: Count is enabled while the SCLHS0 line is low.	R/W
b5	TMOH	Timeout H Count Control	0: Count is disabled while the SCLHS0 line is high. 1: Count is enabled while the SCLHS0 line is high.	R/W
b7, b6	TMOM[1:0]	Timeout Detection Mode Select	b7 b6 0 0: Time-out function is enabled in any of the following conditions. • ICBSR.BFREE flag = 0 and master mode • ICBSR.BFREE flag = 0 and slave mode and own slave address match • ICBSR.BFREE flag = 1 and ICCGR.ST = 1 0 1: Time-out function is enabled during ICBSR.BFREE flag = 0. 1 0: Time-out function is enabled during ICBSR.BFREE flag = 1. 1 1: Setting prohibited	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R

TMOS[1:0] Bits (Timeout Detection Time Select)

These bits are used to select for the timeout detection time when the timeout function is enabled (ICSER.TMOE bit = 1).
When these bits are set to 00b, the timeout detection internal counter functions as a 16-bit counter.
When these bits are set to 01b, the counter functions as a 14-bit counter.
When these bits are set to 10b, the counter functions as a 8-bit counter.
When these bits are set to 11b, the counter functions as a 6-bit counter.
While the SCLHS0 line is in the state that enables this counter as specified by bits TMOH and TMOL, the counter counts up in synchronization with the internal reference clock (IICφ) as a count source.
For details on the timeout function, refer to section 38.11.1, 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 SCLHS0 line is held low when the timeout function is enabled (ICSER.TMOE bit = 1).

TMOH Bit (Timeout H Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCLHS0 line is held high when the timeout function is enabled (ICSER.TMOE bit = 1).

TMOM[1:0] Bits (Timeout Detection Mode Select)

If these bits set to 00b, time-out function is enabled in any of the following three status.

- ICBSR.BFREE flag = 0 and master mode
- ICBSR.BFREE flag = 0 and slave mode and own slave address match
- ICBSR.BFREE flag = 1 and ST bit = 1

If these bits set to 01b, time-out function is enabled during ICBSR.BFREE flag = 0.

If these bits set to 10b or 11b, time-out function is enabled during ICBSR.BFREE flag = 1.

38.2.13 Acknowledgment Bit Control Register (ICACKR)

Address(es): RIICHS0.ICACKR 000E C0A0h

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	ACKWP	ACKBT	ACKBR
Value after 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	ACKBR	Received Acknowledge	0: 0 is received as the acknowledgment bit (ACK reception). 1: 1 is received as the acknowledgment bit (NACK reception).	R
b1	ACKBT	Transmit Acknowledge	0: 0 is to be sent as the acknowledgment bit (ACK transmission). 1: 1 is to be sent as the acknowledgment bit (NACK transmission).	R/W*1
b2	ACKWP	ACKBT Write Protect	0: Writing to the ACKBT bit is disabled. 1: Writing to the ACKBT bit is enabled (when writing simultaneously with the value of the target bit).	W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. When setting a value to the ACKBT bit, set the ACKWP bit to 1 at the same time

ACKBR Bit (Received Acknowledge)

This bit is used to store the acknowledgment bit received from the receiver in transmit mode.

[Setting condition]

- When 1 is received as the acknowledgment bit with the ICMMR.TRS flag set to 1.

[Clearing conditions]

- When 0 is received as the acknowledgment bit with the ICMMR.TRS flag set to 1.
- When the RIICHS is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

ACKBT Bit (Transmit Acknowledge)

[Setting condition]

- When 1 is written to the ACKWP and ACKBT bits at the same time.

[Clearing conditions]

- When 1 is written to the ACKWP bit and 0 is written to the ACKBT bit at the same time.
- When a STOP condition generation is detected. (When a STOP condition is detected with the ICCGR.SP bit set to 1.)

ACKWP Bit (ACKBT Write Protect)

This bit is used to control the modification of the ACKBT bit.

When setting a value to the ACKBT bit, set this bit to 1 at the same time.

This bit is automatically set to 0. The read value is 0.

38.2.14 Clock Stretch Control Register (ICCSCR)

Address(es): RIICHS0.ICCSCR 000E C0A4h

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WAITRE	WAITAE
Value after 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	WAITAE	Acknowledgment Generation Wait Enable	0: The ICCSR.RDRF flag becomes 1 at the rising edge of the ninth SCL. (The SCLHS0 line is not held low at the falling edge of the eighth clock pulse.) 1: The ICCSR.RDRF flag becomes 1 at the rising edge of the eighth SCL. (The SCLHS0 line is held low at the falling edge of the eighth clock pulse.) Low-hold is released by writing a value to the ICACKR.ACKBT bit.	R/W
b1	WAITRE	Receive Data Read Wait Enable	0: No WAIT (The period between ninth clock pulse and first clock pulse is not held low.) 1: WAIT (The period between ninth clock pulse and first clock pulse is held low.) Low-hold is released by reading ICDR register.	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

WAITAE Bit (Acknowledgment Generation Wait Enable)

This bit is used to select the ICCSR.RDRF flag set timing in receive mode and also to select whether to hold the SCLHS0 line low at the falling edge of the eighth SCL.

When the WAITAE bit is 0, the SCLHS0 line is not held low at the falling edge of the eighth SCL, and the ICCSR.RDRF flag is set to 1 at the rising edge of the ninth SCL.

When the WAITAE bit is 1, the ICCSR.RDRF flag is set to 1 at the rising edge of the eighth SCL and the SCLHS0 line is held low at the falling edge of the eighth SCL. The low-hold of the SCLHS0 line is released by writing a value to the ICACKR.ACKBT bit. After data is received with this setting, the SCLHS0 line is automatically held low before the acknowledgment bit is sent. This enables processing to send ACK (ICACKR.ACKBT bit = 0) or NACK (ICACKR.ACKBT bit = 1) according to receive data.

WAITRE Bit (Receive Data Read Wait Enable)

This bit is used to control whether to hold the period between the ninth SCL and the first SCL low until the receive data buffer (ICDR register) is completely read each time single-byte data is received in receive mode.

When the WAITRE bit is 0, the receive operation is continued without holding the period between the ninth and the first SCL low. When both the WAITAE and WAITRE bits are 0, continuous receive operation is enabled with the double buffer.

When the WAITRE bit is 1, the SCLHS0 line is held low from the falling edge of the ninth SCL until the ICDR register value is read each time single-byte data is received. This enables receive operation in byte units.

Note: Read the ICDR register before setting the WAITRE bit to 0.

38.2.15 Condition Generation Request Register (ICCGR)

Address(es): RIICHS0.ICCGR 000E C140h

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	SP	RS	ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ST	START Condition Generation Request	0: Does not request to generate a START condition. 1: Requests to generate a START condition.	R/W
b1	RS	Repeated START Condition Generation Request	0: Does not request to generate a repeated START condition. 1: Requests to generate a repeated START condition.	R/W
b2	SP	STOP Condition Generation Request	0: Does not request to generate a STOP condition. 1: Requests to generate a STOP condition.	R/W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

ST Bit (START Condition Generation Request)

This bit is used to request transition to master mode and generation of a START condition.

For details on the START condition generation, refer to section 38.10, START Condition/Repeated START Condition/STOP Condition Generating 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 generated (a START condition is detected).
- When the ICSR2.AL flag becomes 1 (arbitration is lost).
- When 1 is written to ICRCR.MRST bit or ICRCR.ISRST bit to apply a RIICHS reset or an internal reset.

Note: Set the ST bit to 1 (requests to generate a START condition) when the ICBSR.BFREE flag is set to 1 (bus free state).
Note that arbitration may be lost if the ST bit is set to 1 (requests to generate a START condition) when the ICBSR.BFREE flag is set to 0 (bus busy state).

RS Bit (Repeated START Condition Generation Request)

This bit is used to request that a repeated START condition be generated in master mode.

When this bit is set to 1 to request to generate a repeated START condition, a repeated START condition is generated when the BFREE flag is set to 0 (bus busy state) and the ICMMR.MST flag is set to 1 (master mode).

For details on the repeated START condition generation, refer to section 38.10, START Condition/Repeated START Condition/STOP Condition Generating Function.

[Setting condition]

- When 1 is written to the RS bit with the ICBSR.BFREE flag set to 0.

[Clearing conditions]

- When 0 is written to the RS bit.
- When a repeated START condition has been generated. (a repeated START condition is detected.)

- When the ICSR2.AL flag becomes 1 (arbitration is lost).
- When 1 is written to ICRCR.MRST bit or ICRCR.ISRST bit to apply a RIICHS reset or an internal reset.

Note: Do not set the RS bit to 1 while generating a STOP condition.
Note: If 1 (requests to generate a repeated START condition) is written to the RS bit in slave mode, the repeated START condition is not generated but the RS bit remains set to 1.
If the operating mode changes to master mode with the bit not being cleared, note that the repeated START condition may be generated.

SP Bit (STOP Condition Generation Request)

This bit is used to request that a STOP condition be generated in master mode.
When this bit is set to 1 to request to generate a STOP condition, a STOP condition is generated when the ICBSR.BFREE flag is set to 0 (bus busy state) and the ICMMR.MST flag is set to 1 (master mode).
For details on the STOP condition generation, refer to section 38.10, START Condition/Repeated START Condition/STOP Condition Generating Function.
[Setting condition]

- When 1 is written to the SP bit with the ICBSR.BFREE flag set to 0 and the ICMMR.MST flag set to 1.

[Clearing conditions]

- When 0 is written to the SP bit.
- When a STOP condition has been generated. (a STOP condition is detected.)
- When the ICSR2.AL flag becomes 1 (arbitration is lost).
- When a START condition and a repeated START condition are detected.
- When 1 is written to ICRCR.MRST bit or ICRCR.ISRST bit to apply an RIICHS reset or an internal reset.

Note: When the ICBSR.BFREE flag is 1 (bus free state), writing 1 to the SP bit is ignored.
Note: Do not set the SP bit to 1 while a repeated START condition is being generated.

38.2.16 Transmit/Receive Data Register (ICDR)

Address(es): RIICHS0.ICDR 000E C158h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

The ICDR register is used both to read from the receive data buffer and to write to the transmit data buffer.

For Read Operation:
When 1 byte of data has been received, the received data is transferred from the internal shift register to the ICDR register to enable the next data to be received.
The double-buffer structure of the shift register and the ICDR register allows continuous receive operation if the received data has been read from the ICDR register while the shift register is receiving data.
Read data from the ICDR register once when a receive data full interrupt (RXI) request is generated.
If the ICDR register receives the next receive data before the current data is read from the ICDR register (while the ICCSR.RDRF flag is 1), the RIICHS automatically holds the SCLHS0 line low one cycle before the RDRF flag is set to 1 next.

For Write Operation:
When the ICDR register detects a space in the internal shift register, it transfers the transmit data that has been written to the ICDR register to the shift register and starts transmitting data in transmit mode.
The double-buffer structure of the ICDR register and the shift register allows continuous transmit operation if the next transmit data has been written to the ICDR register while the shift register data is being transmitted.
Write transmit data to the ICDR register once when a transmit data empty interrupt (TXI) request is generated.

38.2.17 Status Register 2 (ICSR2)

Address(es): RIICHS0.ICSR2 000E C1D0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	TMOF	—	—	—	AL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TEND	—	—	—	NACKF	—	—	STOP	START
Value after 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	START	START Condition Detection Flag	0: START condition is not detected. 1: START condition is detected.	R/(W) *1
b1	STOP	STOP Condition Detection Flag	0: STOP condition is not detected. 1: STOP condition is detected.	R/(W) *1
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	NACKF	NACK Detection Flag	0: NACK is not detected 1: NACK is detected.	R/(W) *1
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	TEND	Transmission End Flag	0: Data is being transmitted. 1: Data has been transmitted.	R/(W) *1
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	AL	Arbitration Lost Flag	0: Arbitration is not lost. 1: Arbitration is lost.	R/(W) *1
b19 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	TMOF	Timeout Detection Flag	0: Timeout is not detected. 1: Timeout is detected.	R/(W) *1
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 0 can be written to this bit.

START Flag (START Condition Detection Flag)

[Setting condition]

- The following 1 and 2 are satisfied.
  1. The ICSER.STDE bit is 1.
  2. When a START condition (or a repeated START condition) is detected.

[Clearing condition]

- When 0 is written to the START flag after reading START flag = 1.
- When a STOP condition is detected.
- When the RIICHS is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

STOP Flag (STOP Condition Detection Flag)

[Setting condition]

- The following 1 and 2 are satisfied.
  1. The ICSER.SPDE bit is 1.
  2. When a STOP condition is detected.

[Clearing condition]

- When 0 is written to the STOP flag after reading STOP flag = 1.
- When the RIICHS is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

NACKF Flag (NACK Detection Flag)

[Setting condition]

- All of the following are satisfied.
  1. The ICSER.NAKDE bit is 1 (NACK detection is enabled).
  2. When ACK is not received (NACK is received) from the receiver in transmit mode.

[Clearing condition]

- When 0 is written to the NACKF flag after reading NACKF flag = 1.
- When the RIICHS is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

TEND Flag (Transmission End Flag)

[Setting condition]

- All of the following are satisfied.
  1. The ICSER.TEDE bit is 1 (transmission end detection is enabled).
  2. At the rising edge of the ninth SCL while the ICCSR.TDRE flag is 1. Excluding when sending an address.

[Clearing condition]

- When 0 is written to the TEND flag after reading TEND flag = 1.
- When data is written to the ICDR register.
- When a STOP condition is detected.
- When the RIICHS is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

AL Flag (Arbitration Lost Flag)

[Setting condition]

When master arbitration-lost detection is enabled: ICSER.ALE bit = 1, ICFER.MALE bit = 1.

- When the internal SDA output state does not match the SDAHS0 line level at the rising edge of SCL except for the ACK period during data (including slave address) transmission in master transmit mode (when the SDAHS0 line is driven low while the internal SDA output is high (the SDAHS0 pin is in the high-impedance state)).
- The following 1 and 2 are satisfied.
  1. When the START condition is detected while the ICCGR.ST bit is 1.
  2. When the internal SDA output state does not match the SDAHS0 line level.
- When the ICCGR.ST bit is set to 1 (requests to generate a START condition) while the ICBSR.BFREE flag is 0.

When NACK transmission arbitration-lost detection is enabled: ICSER.ALE bit = 1, ICFER.NALE bit = 1.

- When the internal SDA output state does not match the SDAHS0 line level at the rising edge of SCL in the ACK period during NACK transmission in receive mode.

When slave arbitration-lost detection is enabled: ICSER.ALE bit = 1, ICFER.SALE bit = 1.

- When the internal SDA output state does not match the SDAHS0 line level at the rising edge of SCL except for the ACK period during data transmission in slave transmit mode.

[Clearing condition]

- When 0 is written to the AL flag after reading AL flag = 1.
- When the RIICHS is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

TMOF Flag (Timeout Detection Flag)

[Setting condition]

- All of the following 1 to 3 are satisfied.
  1. The ICSER.TMOE bit is 1 (timeout detection is enabled).
  2. When the master mode or the received slave address matches the slave address n in slave mode.
  3. When the SCLHS0 line state remains unchanged for the period specified by ICTOR register.

[Clearing condition]

- When 0 is written to the TMOF flag after reading TMOF flag = 1.
- When the RIICHS is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

38.2.18 Status Detection Enable Register (ICSER)

Address(es): RIICHS0.ICSER 000E C1D4h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	TMOE	—	—	—	ALE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TEDE	—	—	—	NAKDE	—	—	SPDE	STDE
Value after 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	STDE	START Condition Detection Enable	0: START condition detection is disabled. 1: START condition detection is enabled.	R/W
b1	SPDE	STOP Condition Detection Enable	0: STOP condition detection is disabled. 1: STOP condition detection is enabled.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	NAKDE	NACK Detection Enable	0: NACK detection is disabled. 1: NACK detection is enabled.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	TEDE	Transmission End Detection Enable	0: Transmission end detection is disabled. 1: Transmission end detection is enabled.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	ALE	Arbitration Lost Detection Enable	0: Arbitration lost detection is disabled. 1: Arbitration lost detection is enabled.	R/W
b19 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	TMOE	Timeout Detection Enable	0: Timeout detection is disabled. 1: Timeout detection is enabled.	R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R

This register is used to enable or disable detection of each status. When the status enabled by this register is detected, the corresponding flag in the ICSR2 register is set to 1.

38.2.19 Status Interrupt Enable Register (ICSIER)

Address(es): RIICHS0.ICSIER 000E C1D8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	TMOIE	—	—	—	ALIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TEIE	—	—	—	NAKIE	—	—	SPIE	STIE
Value after 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	STIE	START Condition Detection Interrupt Enable	0: START condition detection interrupt is disabled. 1: START condition detection interrupt is enabled.	R/W
b1	SPIE	STOP Condition Detection Interrupt Enable	0: STOP condition detection interrupt is disabled. 1: STOP condition detection interrupt is enabled.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	NAKIE	NACK Detection Interrupt Enable	0: NACK detection interrupt is disabled. 1: NACK detection interrupt is enabled.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b8	TEIE	Transmission End Interrupt Enable	0: Transmission end interrupt is disabled. 1: Transmission end interrupt is enabled.	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R
b16	ALIE	Arbitration Lost Detection Interrupt Enable	0: Arbitration lost interrupt is disabled. 1: Arbitration lost interrupt is enabled.	R/W
b19 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R
b20	TMOIE	Timeout Detection Interrupt Enable	0: Timeout detection interrupt is disabled. 1: Timeout detection interrupt is enabled.	R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R

This register is used to enable or disable interrupts for each status. When a flag in the ICSR2 register becomes 1 and the corresponding interrupt enable bit is 1, an interrupt request is output.

38.2.20 Communication Status Register (ICCSR)

Address(es): RIICHS0.ICCSR 000E C1E0h

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDRF	TDRE
Value after 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	TDRE	Transmit Data Empty Flag	0: Transmit data buffer contains transmit data. 1: Transmit data buffer contains no transmit data.	R
b1	RDRF	Receive Data Full Flag	0: Receive data buffer contains no receive data. 1: Receive data buffer contains receive data.	R
b31 to b2	—	Reserved	These bits are read as 0 and cannot be modified.	R

TDRE Flag (Transmit Data Empty Flag)

[Setting condition]

- When data is transferred from transmit data buffer to shift register and transmit data buffer becomes empty.
- When the ICMMR.TRS flag is set to 1.
- When the received slave address matches while the TRS flag is 1.

[Clearing condition]

- When data is written to ICDR register.
- When the ICMMR.TRS flag is set to 0.
- When the RIICHS is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

Note: When the ICSR2.NACKF flag is set to 1 while the ICSER.NAKDE bit is 1, RIICHS aborts data transmission/reception. Here, if the TDRE flag is 0 (next transmit data has been written), data is transferred to the shift register and the transmit data buffer becomes empty at the rising edge of the ninth clock cycle, but the TDRE flag is not set to 1.

RDRF Flag (Receive Data Full Flag)

[Setting condition]

- When receive data is transferred from shift register to receive data buffer.
  The RDRF flag is set to 1 on the rising edge of the eighth or ninth SCL (selected in the ICCSCR.WAITAE bit).
- When the received slave address matches after a START (or repeated START) condition is detected with the ICMMR.TRS flag set to 0.

[Clearing condition]

- When data is read from ICDR register.
- When the RIICHS is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

38.2.21 Communication Status Detection Enable Register (ICCSER)

Address(es): RIICHS0.ICCSER 000E C1E4h

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDE	TDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TDE	Transmit Data Empty Detection Enable	0: Transmit data empty detection is disabled. 1: Transmit data empty detection is enabled.	R/W
b1	RDE	Receive Data Full Detection Enable	0: Receive data full detection is disabled. 1: Receive data full detection is enabled.	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

This register is used to enable or disable detection of each status. When the status enabled by this register is detected, the corresponding flag in the ICCSR register is set to 1.

38.2.22 Communication Status Interrupt Enable Register (ICCSIER)

Address(es): RIICHS0.ICCSIER 000E C1E8h

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RIE	TIE
Value after 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	TIE	Transmit Data Empty Interrupt Enable	0: Transmit data empty interrupt is disabled. 1: Transmit data empty interrupt is enabled.	R/W
b1	RIE	Receive Data Full Interrupt Enable	0: Receive data full interrupt is disabled. 1: Receive data full interrupt is enabled.	R/W
b31 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

This register is used to enable or disable interrupts for each status. When a flag in the ICCSR register becomes 1 and the corresponding interrupt enable bit is 1, an interrupt request is output.

38.2.23 Bus Status Register (ICBSR)

Address(es): RIICHS0.ICBSR 000E C210h

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BFREE
Value after 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	BFREE	Bus Free Flag	0: Have not detected bus free. 1: Have detected bus free.	R
b31 to b1	—	Reserved	These bits are read as 0.	R

BFREE Flag (Bus Free Flag)

This flag indicates whether the I2C-bus is occupied (bus busy) or released (bus free).

[Setting condition]

- When the number of cycles (IICφ) set in the ICBFTR register has elapsed when the SCLHS0 and SDAHS0 lines are high after a STOP condition is detected.
- When the number of cycles (IICφ) set in the ICBFTR register has elapsed when the SCLHS0 and SDAHS0 lines are high after setting the ICCR.ICE bit to 1.

[Clearing condition]

- When a START condition is detected.
- When the ICCR.ICE bit is set to 0.
- When the RIICHS is reset by setting the ICRCR.MRST bit to 1.
- When the internal status is reset by setting the ICRCR.ISRST bit to 1.

38.2.24 Slave Mode Status Register (ICSSR)

Address(es): RIICHS0.ICSSR 000E C214h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	AAS2	AAS1	AAS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	HOA	—	—	—	—	—	—	—	—	DID	HSMC	—	—	—	—	GCA
Value after 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	GCA	General Call Address Detection Flag	0: General call address is not detected. 1: General call address is detected.	R/(W) *1
b4 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b5	HSMC	Hs-mode Master Code Detection Flag	0: Hs-mode master code is not detected. 1: Hs-mode master code is detected.	R/(W) *1
b6	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 (write)).	R/(W) *1
b14 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R
b15	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
b16	AAS0	Slave Address 0 Detection Flag	0: Slave address 0 is not detected. 1: Slave address 0 is detected.	R/(W) *1
b17	AAS1	Slave Address 1 Detection Flag	0: Slave address 1 is not detected. 1: Slave address 1 is detected.	R/(W) *1
b18	AAS2	Slave Address 2 Detection Flag	0: Slave address 2 is not detected. 1: Slave address 2 is detected.	R/(W) *1
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Only 0 can be written to this bit.

GCA Flag (General Call Address Detection Flag)

[Setting condition]

- This flag is set to 1 at the rising edge of the ninth SCL in the first byte when the following 1 and 2 are satisfied.
  1. The ICSCR.GCAE bit is 1 (general call address detection is enabled).
  2. When the received slave address matches the general call address (0000 000b + 0 (write)).

[Clearing condition]

- When 0 is written to the GCA flag after reading GCA flag to be 1.
- When a STOP condition is detected.
- When a repeated START condition is detected.

HSMC Flag (Hs-mode Master Code Detection Flag)

[Setting condition]

- This flag is set to 1 at the rising edge of the ninth SCL in the first byte when the following 1 and 2 are satisfied.
  1. The ICSCR.HSMCE bit is 1 (Hs-mode master code detection is enabled).
  2. When the first byte received immediately after a START condition is detected matches a value of Hs-mode master code (0000 1XXXb) + 1 (NACK).

[Clearing condition]

- When 0 is written to the HSMC flag after reading HSMC flag to be 1.
- When a STOP condition is detected.

DID Flag (Device-ID Address Detection Flag)

[Setting condition]

- This flag is set to 1 at the rising edge of the ninth SCL in the first byte when the following 1 and 2 are satisfied.
  1. The ICSCR.DIDE bit is 1 (device-ID address detection is enabled).
  2. When the first byte received immediately after a START condition or repeated START condition is detected matches a value of device ID (1111 100b) + 0 (write).

[Clearing condition]

- When 0 is written to the DID flag after reading DID flag to be 1.
- When a STOP condition is detected.
- This flag is set to 0 at the rising edge of the ninth SCL in the first byte when the following 1 and 2 or 1 and 3 are satisfied.
  1. The ICSCR.DIDE bit is 1 (device-ID address detection is enabled).
  2. When the first byte received immediately after a START condition or repeated START condition is detected does not match a value of device ID (1111 100b).
  3. When the first byte received immediately after a START condition or repeated START condition is detected matches a value of (device ID (1111 100b) + 0 (write)) and the second byte does not match any of slave addresses 0 to 2.

HOA Flag (Host Address Detection Flag)

[Setting condition]

- This flag is set to 1 at the rising edge of the ninth SCL in the first byte when the following 1 and 2 are satisfied.
  1. The ICSCR.HOAE bit is 1 (host address detection is enabled).
  2. When the received slave address matches the host address (0001 000b).

[Clearing condition]

- When 0 is written to the HOA flag after reading HOA flag to be 1.
- When a STOP condition is detected.
- When a repeated START condition is detected.

AASy Flag (Slave Address y Detection Flag) (y = 0 to 2)

[Setting condition]

For 7-bit address format: SAMRy.FS flag = 0.

- This flag is set to 1 at the rising edge of the ninth SCL in the first byte when the following 1 and 2 are satisfied.
  1. The ICSCR.SARyE bit is 1 (slave y is enabled).
  2. When the received slave address matches the SAMRy.SVA[6:0] bits value.

For 10-bit address format: SAMRy.FS flag = 1.

- This flag is set to 1 at the rising edge of the ninth SCL in the second byte when the following 1 and 2 are satisfied.
  1. The ICSCR.SARyE bit is 1 (slave y is enabled).
  2. When the received slave address matches a value of 11110b + SAMRy.SVA[9:8] bits and the following address matches the SAMRy.SVA[7:0] bits value.

[Clearing condition]

- When 0 is written to the AASy flag after reading AASy flag to be 1.
- When a STOP condition is detected.

For 7-bit address format: SAMRy.FS flag = 0.

- This flag is set to 0 at the rising edge of the ninth SCL in the first byte when the following 1 and 2 are satisfied.
  1. The ICSCR.SARyE bit is 1 (slave y is enabled).
  2. When the received slave address does not match SAMRy.SVA[6:0] bits value.

For 10-bit address format: SAMRy.FS flag = 1.

- This flag is set to 0 at the rising edge of the ninth SCL in the first byte when the following 1 and 2 are satisfied.
  1. The ICSCR.SARyE bit is 1 (slave y is enabled).
  2. When the received slave address does not match a value of 11110b + SAMRy.SVA[9:8] bits.
- This flag is set to 0 at the rising edge of the ninth SCL in the second byte when the following 1 and 2 are satisfied.
  1. The ICSCR.SARyE bit is 1 (slave y is enabled).
  2. When the received slave address matches a value of 11110b + SAMRy.SVA[9:8] bits and the following address does not match the SAMRy.SVA[7:0] bits value.

38.2.25 Slave Address Register y (SARy) (y = 0 to 2)

Address(es): RIICHS0.SAR0 000E C2B0h, RIICHS0.SAR1 000E C2B4h, RIICHS0.SAR2 000E C2B8h

	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
	—	—	—	—	—	FS	SVA[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	SVA[9:0]	Slave Address	Set a slave address	R/W
b10	FS	Address Format Select	0: 7-bit address format 1: 10-bit address format	R/W
b31 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R

FS Bit (Address Format Select)

This bit is used to select 7-bit address or 10-bit address for slave address (SVA[9:0] bits).

When this bit is 0, the 7-bit address format is selected for slave address, the SVA[6:0] bit setting is valid, and the SVA[9:7] bit setting is ignored.

When this bit is 1, the 10-bit address format is selected for slave address and the SVA[9:0] bit setting is valid.

When the ICSCR.SARyE bit is 0, the setting of this bit is ignored.

38.2.26 Slave Address Monitor Register y (SAMRy) (y = 0 to 2)

Address(es): RIICHS0.SAMR0 000E C330h, RIICHS0.SAMR1 000E C334h, RIICHS0.SAMR2 000E C338h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	SAV	—	—	FS	—	SVA[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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are read as 0.	R
b25 to b16	SVA[9:0]	Slave Address	Slave address is set.	R
b26	—	Reserved	This bit is read as 0.	R
b27	FS	Address Format Flag	0: The 7-bit address format is selected. 1: The 10-bit address format is selected.	R
b29, b28	—	Reserved	These bits are read as 0.	R
b30	SAV	Static Address Valid Flag	0: Slave address is disabled. 1: Slave address is enabled.	R
b31	—	Reserved	This bit is read as 0.	R

SVA[9:0] Bits (Slave Address)

The SVA[9:0] bits indicate a valid slave address.

[The SAMRy.SAV flag is 1 and the SAMRy.FS flag is 0.]

- The SVA[9:7] bits are 000b.
- The SVA[6:0] bits are the SARy.SVA[6:0] bits.

[The SAMRy.SAV flag is 1 and the SAMRy.FS flag is 1.]

- The SVA[9:0] bits are the SARy.SVA[9:0] bits.

FS Flag (Address Format Flag)

[Setting condition]

- All of the following are satisfied.
  1. The ICSCR.SARyE bit is 1 (slave y is enabled).
  2. The SARy.FS bit is 1 (the address length is 10 bits).

[Clearing condition]

- [Setting condition] is not satisfied.

SAV Flag (Static Address Valid Flag)

[Setting condition]

- All of the following are satisfied.
  1. The ICSCR.SARyE bit is 1 (slave y is enabled).
  2. If the SAMRy.FS flag is 0, the SARy.SVA[6:0] bits are not 0000000b.
     If the SAMRy.FS flag is 1, the SARy.SVA[9:0] bits are not 0000000000b.

[Clearing condition]

- [Setting condition] is not satisfied.

38.2.27 Bit Count Register (ICBCR)

Address(es): RIICHS0.ICBCR 000E C380h

	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
	—	—	—	—	—	—	—	—	—	—	—	BC[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	BC[4:0]	Bit Counter	These bits indicate the number of bits remaining to be transferred. Refer to Table 38.4 for details.	R
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R

BC[4:0] 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 SCLHS0 line.

Table 38.4 Counter Value and Number of Remaining Bits

	Master		Slave	
BC[4:0]	Address phase	Data phase	Address phase	Data phase
00h	2 or 1 bits	2 or 1 bits	3 to 1 bits	2 or 1 bits
01h	3 bits	3 bits	4 bits	3 bits
02h	4 bits	4 bits	5 bits	4 bits
03h	5 bits	5 bits	6 bits	5 bits
04h	6 bits	6 bits	7 bits	6 bits
05h	7 bits	7 bits	8 bits	7 bits
06h	8 bits	8 bits	9 bits	8 bits
07h	9 bits	9 bits	—	9 bits

38.2.28 Internal Status Monitor Register (ICIMR)

Address(es): RIICHS0.ICIMR 000E C3CCh

	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
	—	—	—	—	—	—	—	—	—	—	—	—	SDAO	SCLO	SDAI	SCLI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	SCLI	SCL Line Monitor Flag	0: SCLHS0 line is low. 1: SCLHS0 line is high.	R
b1	SDAI	SDA Line Monitor Flag	0: SDAHS0 line is low. 1: SDAHS0 line is high.	R
b2	SCLO	SCL Output Monitor Flag	0: The RIICHS has driven the SCLHS0 pin low. 1: The RIICHS has released the SCLHS0 pin.	R
b3	SDAO	SDA Output Monitor Flag	0: The RIICHS has driven the SDAHS0 pin low. 1: The RIICHS has released the SDAHS0 pin.	R
b31 to b4	—	Reserved	These bits are read as 0.	R

SCLO Flag (SCL Output Monitor Flag) and SDAO Flag (SDA Output Monitor Flag)

When reading these bits, the state of signals output from the RIICHS can be read.

38.3 Operation

38.3.1 Communication Data Format

The I2C-bus format consists of 8-bit data and 1-bit acknowledgment. The frame following a START condition or repeated START 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 generated. Figure 38.2 shows the I2C-bus format, and Figure 38.3 shows the I2C-bus timing.

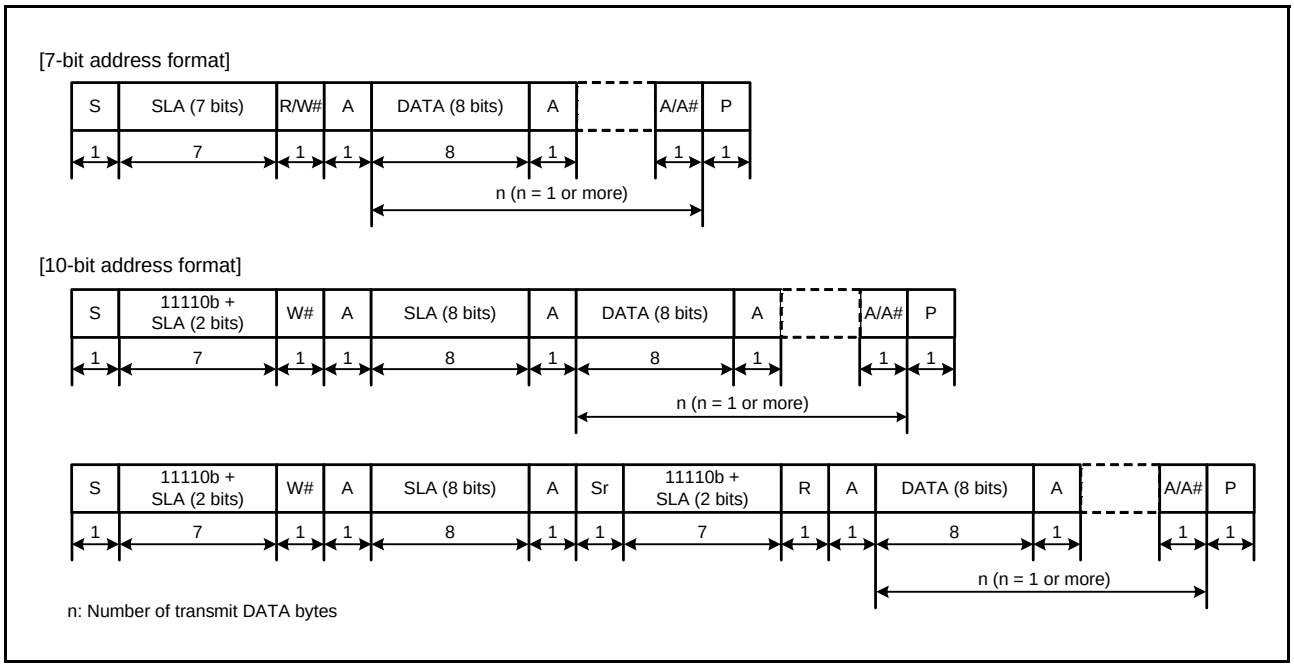


Figure 38.2 I2C-bus Format

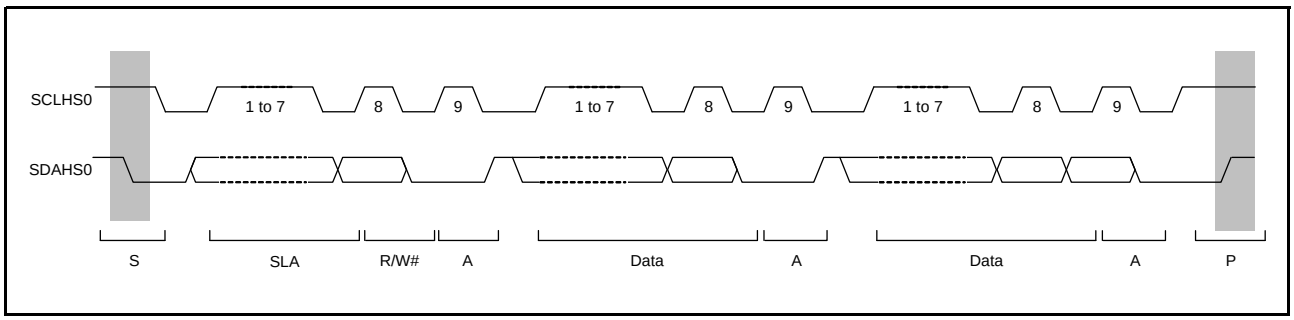


Figure 38.3 I2C-bus Timing (SLA = 7 Bits)

S: START condition. The master device drives the SDAHS0 line low from high while the SCLHS0 line is high.

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 receiver drives the SDAHS0 line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)

A#: Not Acknowledge. The receiver drives the SDAHS0 line high.

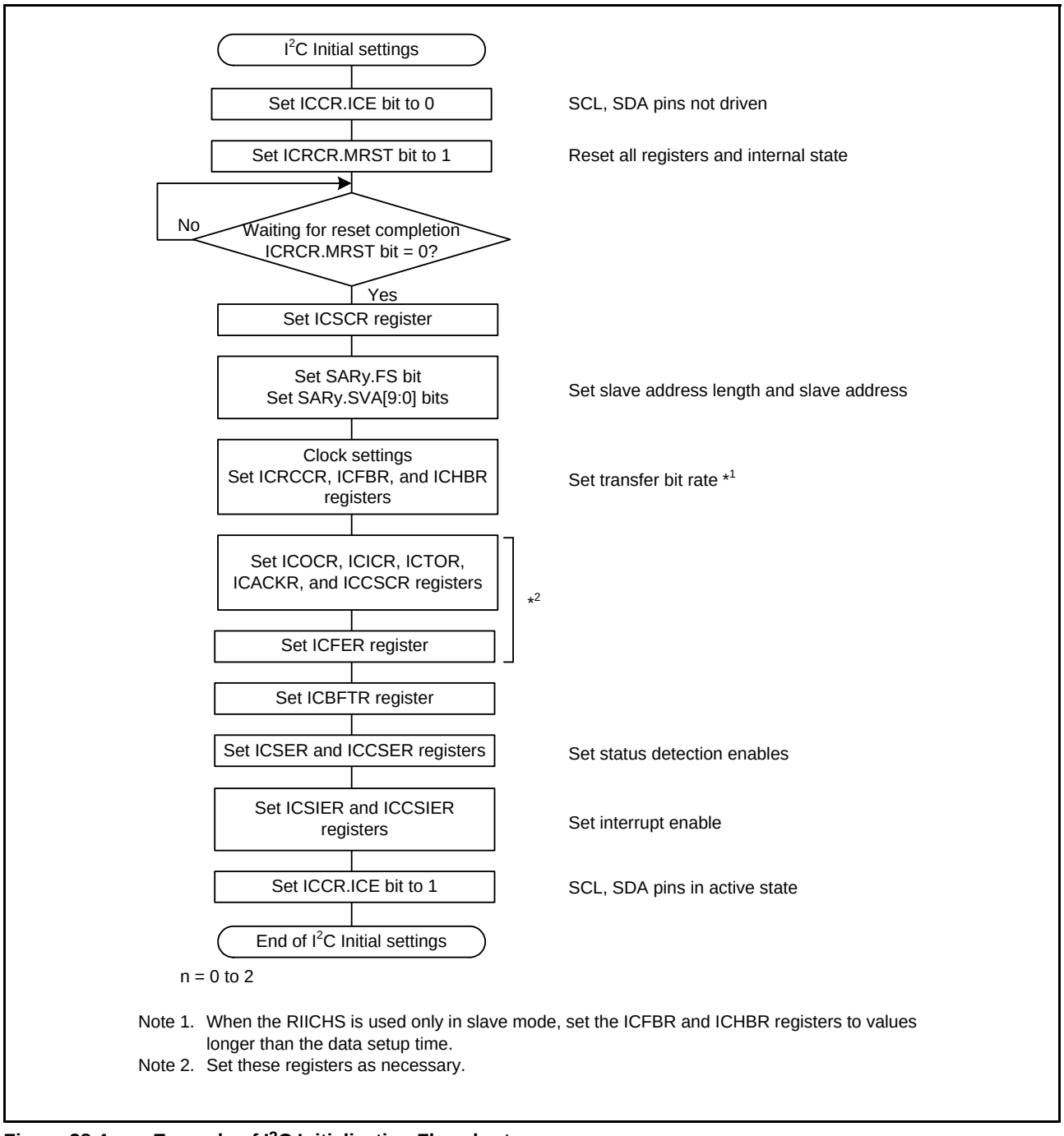
Sr: Repeated START condition. The master device drives the SDAHS0 line low from high after the setup time has elapsed with the SCLHS0 line high.

DATA: Transmitted or received data

P: STOP condition. The master device drives the SDAHS0 line high from low while the SCLHS0 line is high.

38.3.2 Initial Settings

Before starting data transmission and reception, initialize the RIICHS according to the procedure in Figure 38.4.
First, Set the ICCR.ICE bit to 0 (SCLHS0, SDAHS0 pins not driven).
Next, Set the ICRCR.MRST bit to 1 (RIICHS reset). This initialize the all registers and internal state. Refer to Table 38.6, Reset control of Registers. Then, it waits for MRST to become 0.
After that, set SARy.FS bit (y = 0 to 2), SARy.SVA[9:0] bits, ICFBR register, ICICR register, ICOCR register, ICTOR register, ICCSCR register, ICACKR register, ICFER register, and set the other registers as necessary (for initial settings of the RIICHS, see Figure 38.4).



Note 1. When the RIICHS is used only in slave mode, set the ICFBR and ICHBR registers to values longer than the data setup time.
Note 2. Set these registers as necessary.

Figure 38.4 Example of I2C Initialization Flowchart

38.3.3 Master Transmit Operation

In master transmit operation, the RIICHS outputs the SCL and transmitted data signals as the master device, and the slave device returns acknowledgments. Figure 38.5 shows an example of usage of master transmission and Figure 38.6 to Figure 38.8 show the timing of operations in master transmission.
The following describes the procedure and operations for master transmission.

(1) Initial settings. For details, refer to section 38.3.2, Initial Settings.
(2) Read the ICBSR.BFREE flag to check that the bus is open, and then set the ICCGR.ST bit to 1 (requests to generate a START condition). Upon receiving the request, the RIICHS generates a START condition. At the same time, the BFREE flag is automatically set to 0, the ICSR2.START flag is automatically set to 1 and the ICCGR.ST bit is automatically set to 0. At this time, if the START condition is detected and the internal levels for the SDA output state and the levels on the SDAHS0 line have matched while the ICCGR.ST bit is 1, the RIICHS recognizes that generating of the START condition as requested by the ICCGR.ST bit has been successfully completed, and the ICMMR.MST and ICMMR.TRS flags is automatically set to 1, placing the RIICHS in master transmit mode. The ICCSR.TDRE flag is also automatically set to 1 in response to setting of the TRS flag to 1.
(3) Check that the ICCSR.TDRE flag is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDR register. Once the data for transmission are written to ICDR register, the ICCSR.TDRE flag is automatically set to 0, and the ICCSR.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 flag is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the RIICHS 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 ICCGR.SP bit to generate a STOP condition.
For data transmission with an address in the 10-bit format, start by writing 1111 0b, the 2 higher-order bits of the slave address, and W to ICDR register as the first address transmission. Then, as the second address transmission, write the 8 lower-order bits of the slave address to ICDR register.
(4) After confirming that the ICCSR.TDRE flag is 1, write the data for transmission to the ICDR register. The RIICHS automatically holds the SCLHS0 line low until the data for transmission are ready or a STOP condition is generated
(5) After all bytes of data for transmission have been written to the ICDR register, wait until the value of the ICSR2.TEND flag returns to 1, and then set the ICCGR.SP bit to 1 (requests to generate a STOP condition). Upon receiving a STOP condition generation request, the RIICHS generates the STOP condition.
(6) Upon detecting the STOP condition, the RIICHS automatically sets the ICMMR.MST and ICMMR.TRS flags to 00b and enters slave receive mode. Furthermore, it automatically sets the ICCSR.TDRE flag to 0, ICSR2.TEND flag to 0 and sets the ICSR2.SPDF flag to 1.
(7) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

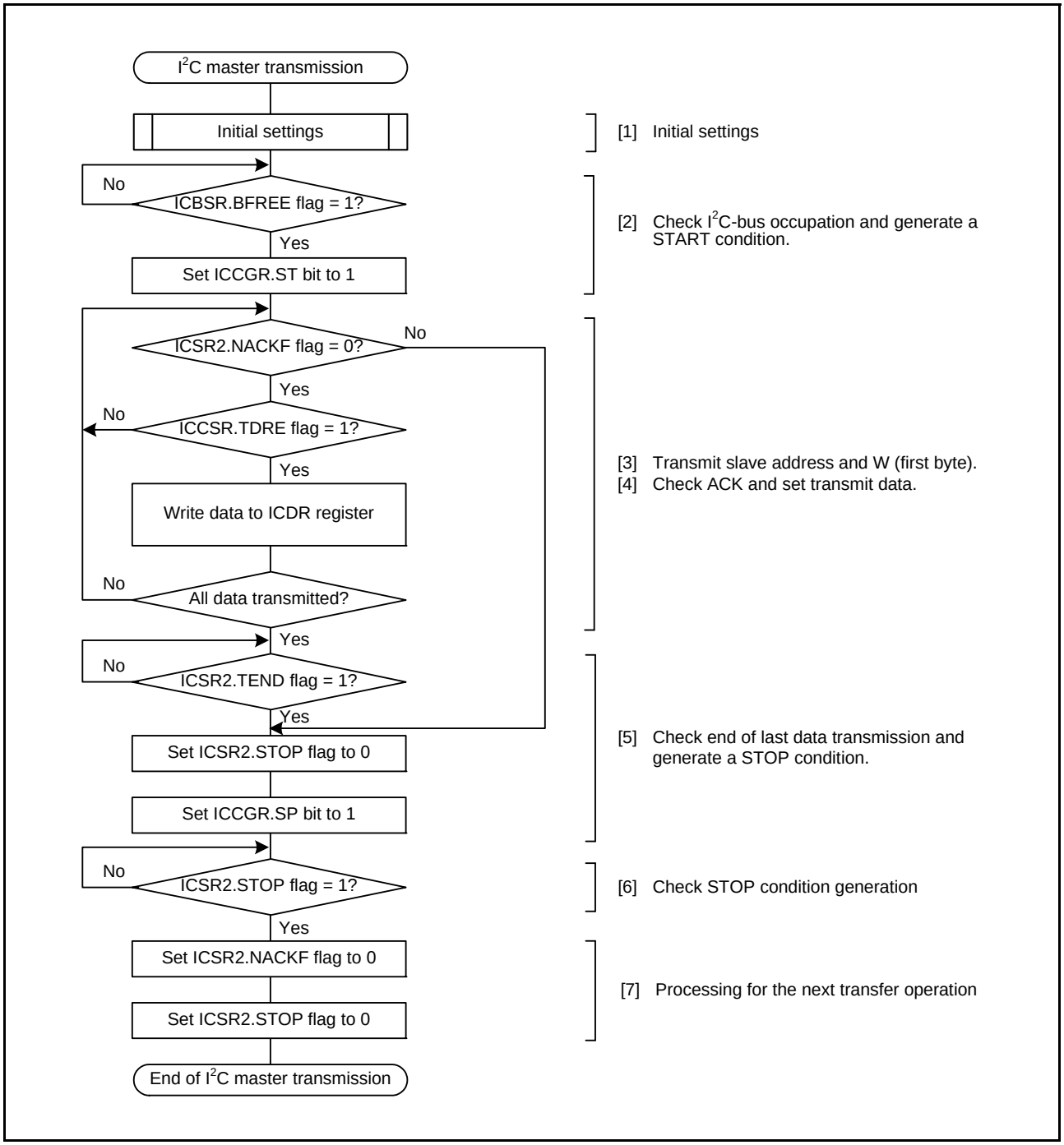


Figure 38.5 Example of I2C Master Transmission Flowchart

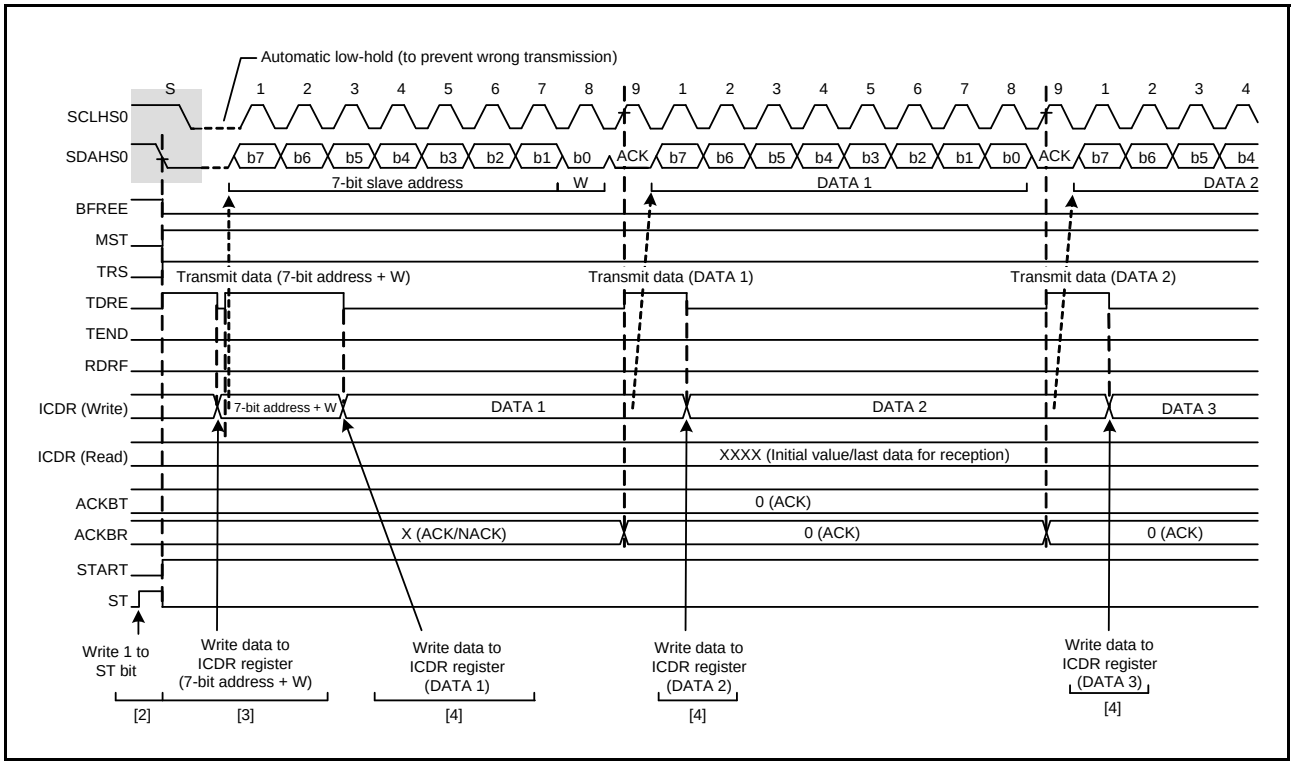


Figure 38.6 Master Transmit Operation Timing (1) (7-Bit Address Format)

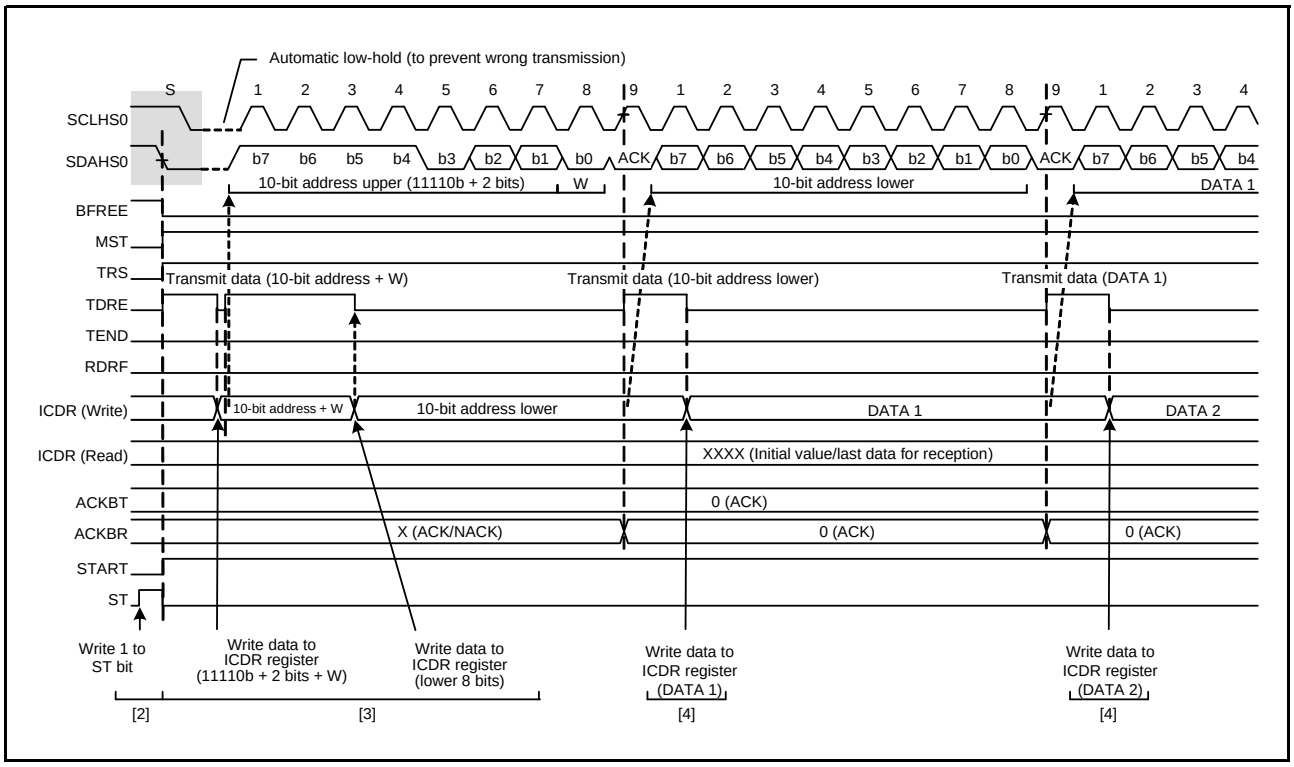


Figure 38.7 Master Transmit Operation Timing (2) (10-Bit Address Format)

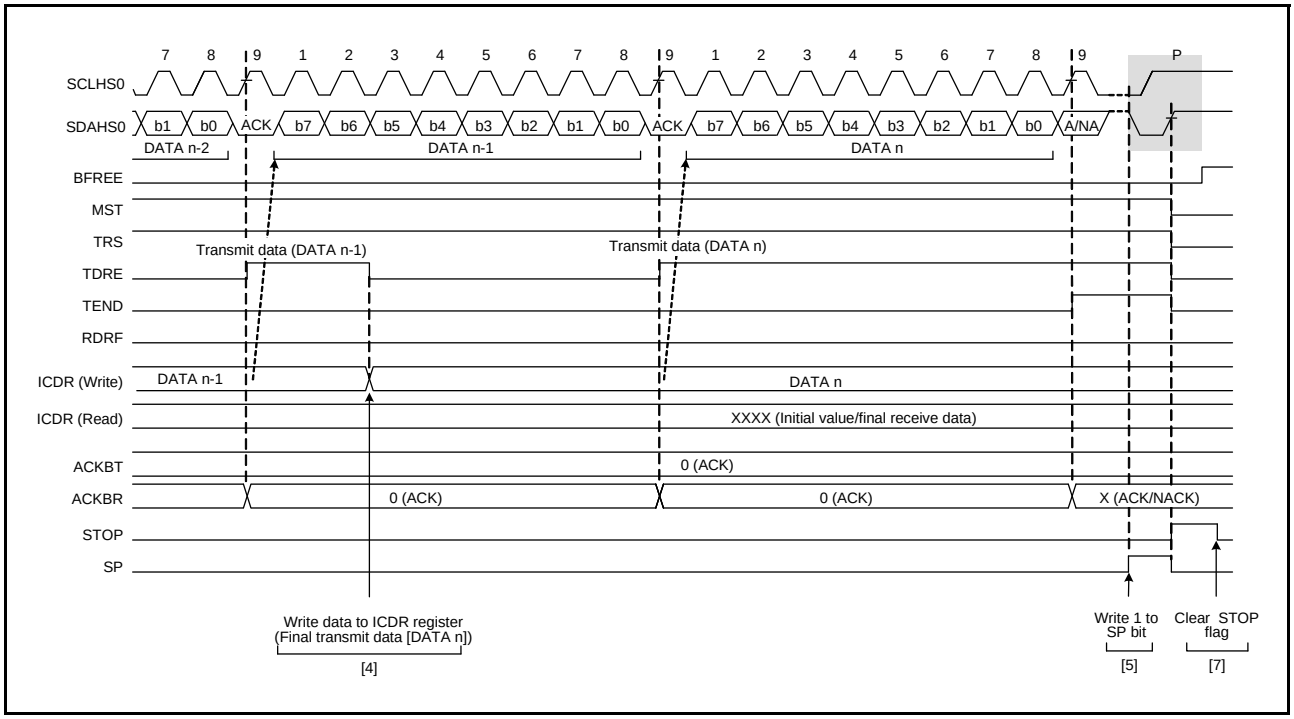


Figure 38.8 Master Transmit Operation Timing (3)

38.3.4 Master Receive Operation

In master receive operation, the RIICHS as a master device outputs the SCL, receives data from the slave device, and returns acknowledgments. Since the RIICHS 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 38.9 and Figure 38.10 show examples of usage of master reception and Figure 38.11 to Figure 38.13 show the timing of operations in master reception.
The following describes the procedure and operations for master reception.

(1) Initial settings. For details, refer to section 38.3.2, Initial Settings.
(2) Read the ICBSR.BFREE flag to check that the bus is open, and then set the ICCGR.ST bit to 1 (requests to generate a START condition). Upon receiving the request, the RIICHS generates a START condition. When the RIICHS detects the START condition, the ICBSR.BFREE flag is automatically set to 0 and the ICSR2.START flag is automatically set to 1 and the ICCGR.ST bit is automatically set to 0. At this time, if the START condition is detected and the levels for the SDA output and the levels on the SDAHS0 line have matched while the ICCGR.ST bit is 1, the RIICHS recognizes that generating of the START condition as requested by the ICCGR.ST bit has been successfully completed, and the ICMMR.MST and ICMMR.TRS flags is automatically set to 1, placing the RIICHS in master transmit mode. The ICCSR.TDRE flag is also automatically set to 1 in response to setting of the TRS flag to 1.
(3) Check that the ICCSR.TDRE flag is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to ICDR register. Once the data for transmission are written to ICDR register, the ICCSR.TDRE flag is automatically set to 0, and the ICCSR.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 ICMMR.TRS flag 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 flag is set to 0 on the rising edge of the ninth SCL, placing the RIICHS in master receive mode. At this time, the ICCSR.TDRE flag is set to 0 and if the slave device acknowledges the ICCSR.RDRF flag is automatically set to 1.
If no slave device recognized the address or there was an error in communications, the ICSR2.NACKF flag becomes 1. In this case, write 1 to the ICCGR.SP bit to generate a STOP condition. For master reception from a device with a 10-bit address, start by using master transmission to send the 10-bit address, and then generate a repeated START condition. After that, transmitting 1111 0b, the two higher-order bits of the slave address, and the R bit places the RIICHS in master receive mode.
(4) Dummy read ICDR register after confirming that the ICCSR.RDRF flag is 1; this makes the RIICHS start output of the SCL and start data reception.
(5) After 1 byte of data has been received, the ICCSR.RDRF flag is set to 1 on the rising edge of the eighth or ninth SCL (the clock signal) as selected by the ICCSCR.WAITAE bit. Reading out ICDR register at this time will produce the received data, and the RDRF flag is automatically set to 0 at the same time. Furthermore, the value of the acknowledgment bit received during the ninth SCL is returned as the value set in the ICACKR.ACKBT bit. Furthermore, if the next byte to be received is the next to last byte, set the ICCSCR.WAITRE bit to 1 (for wait insertion) before reading the ICDR register (containing the second byte from last). As well as enabling NACK output even in the case of delays in processing to set the ICACKR.ACKBT bit to 1 (NACK) in step (6), due to other interrupts, etc., this fixes the SCLHS0 line to low on the falling edge of the ninth clock pulse in reception of the last byte, so the state is such that generating a STOP condition is possible.
(6) When the ICCSCR.WAITAE 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 ICACKR.ACKBT bit to 1 (NACK).
(7) After reading out the byte before last from the ICDR register, if the value of the ICCSR.RDRF flag is confirmed to be 1, write 1 to the ICCGR.SP bit (requests to generate a STOP condition) and then read the last byte from ICDR register. When the ICCGR.SP bit is set to 1, the RIICHS is released from the wait state and generates the STOP condition after low-level output in the ninth clock pulse is completed or the SCLHS0 line is released from the low-hold state.

(8) Upon detecting the STOP condition, the RIICHS automatically sets the ICMMR.MST and ICMMR.TRS flags to 00b and enters slave receive mode. Furthermore, detection of the STOP condition leads to setting of the ICSR2.STOP flag to 1.

(9) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

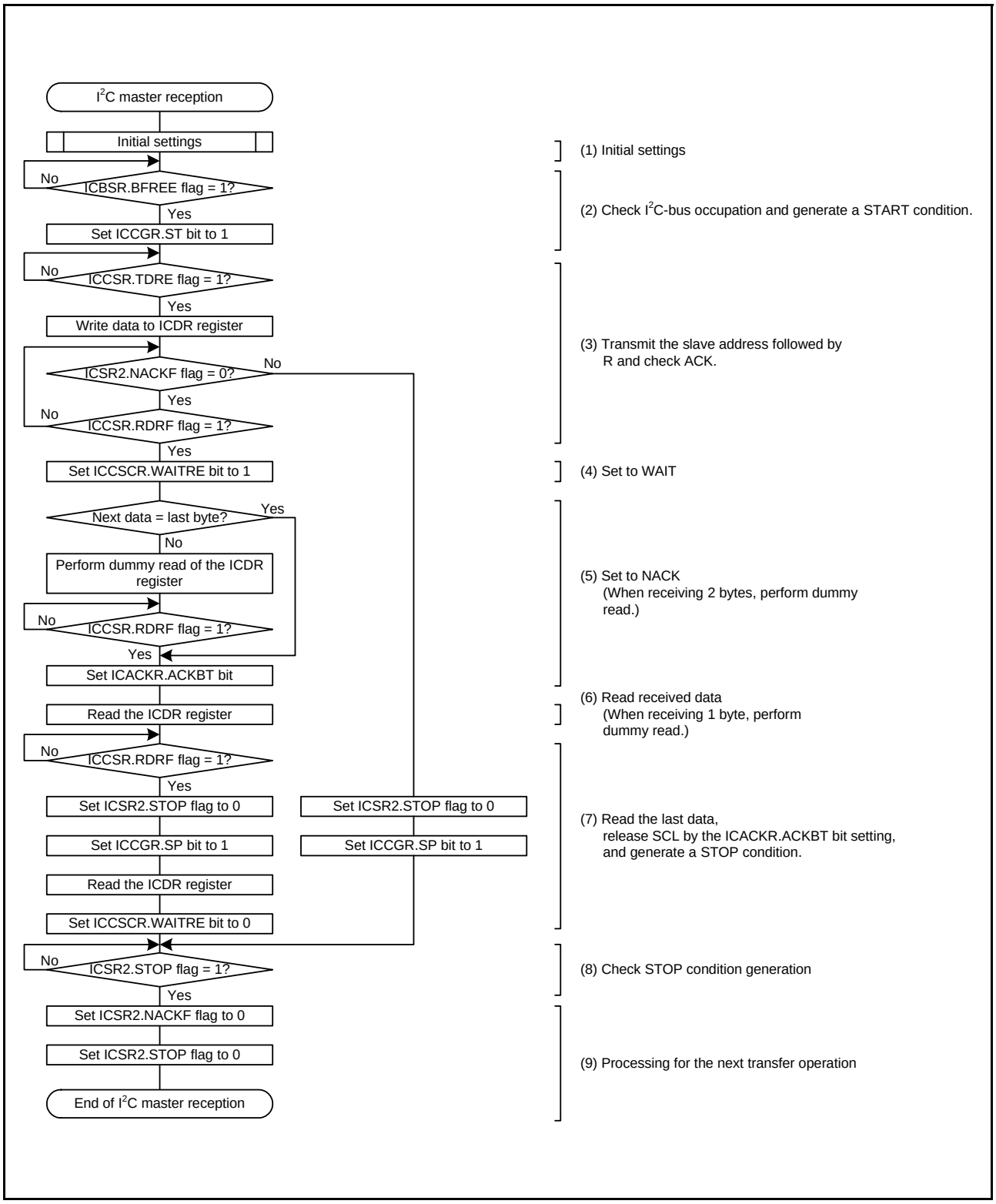


Figure 38.9 Example of I2C Master Reception Flowchart (7-Bit Address Format, 1 or 2 bytes)

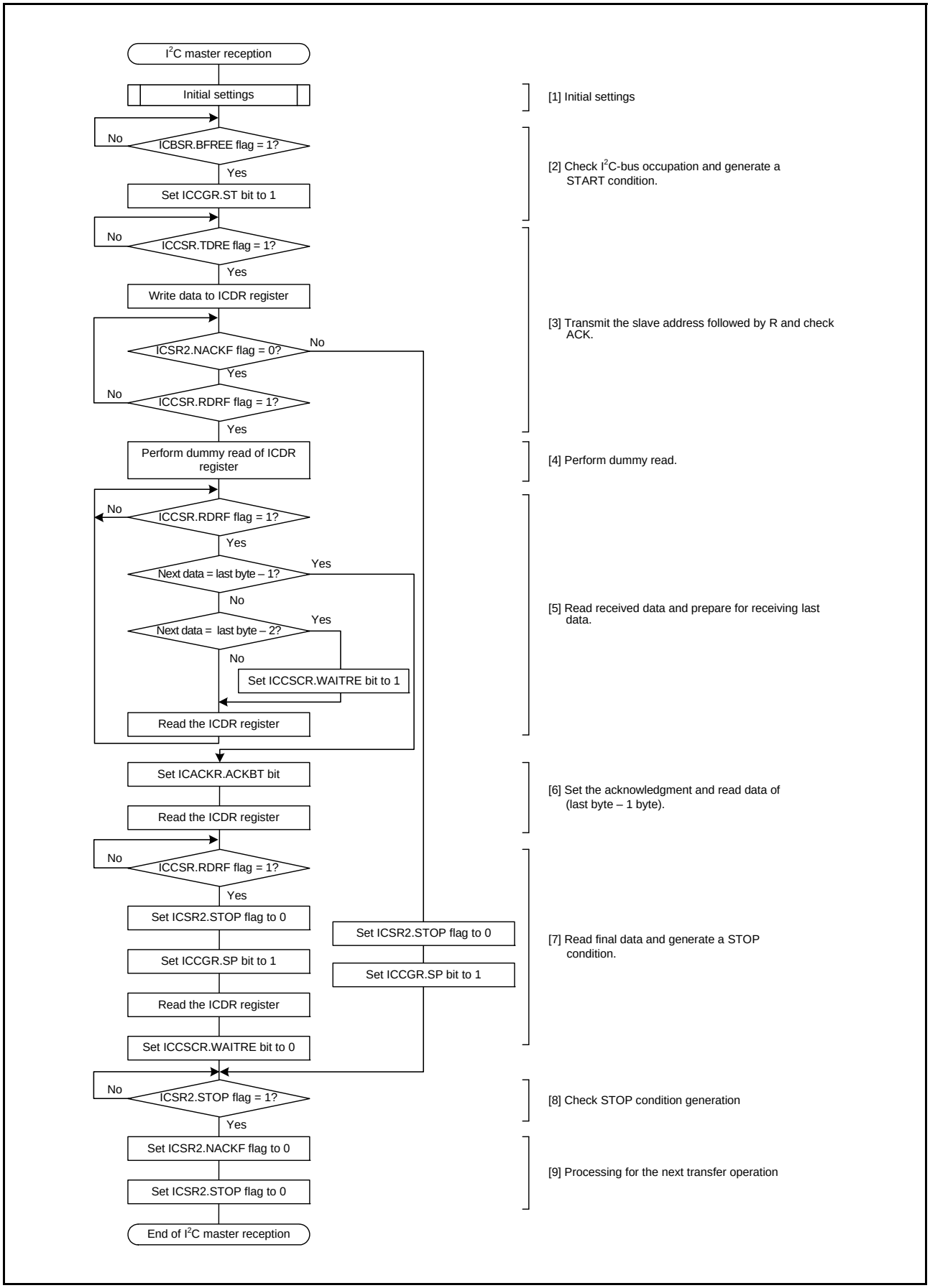


Figure 38.10 Example of I2C Master Reception Flowchart (7-Bit Address Format, 3 Bytes or More)

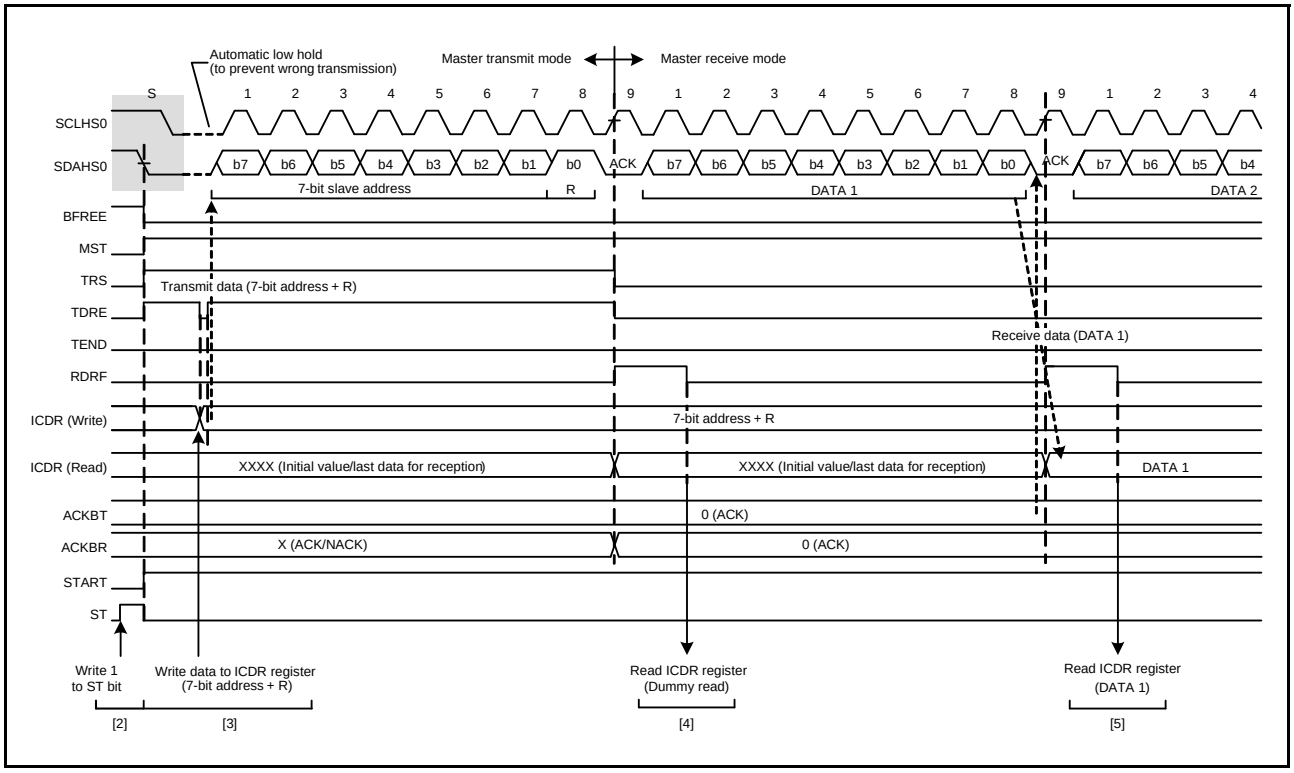


Figure 38.11 Master Receive Operation Timing (1) (7-Bit Address Format, When WAITAE = 0)

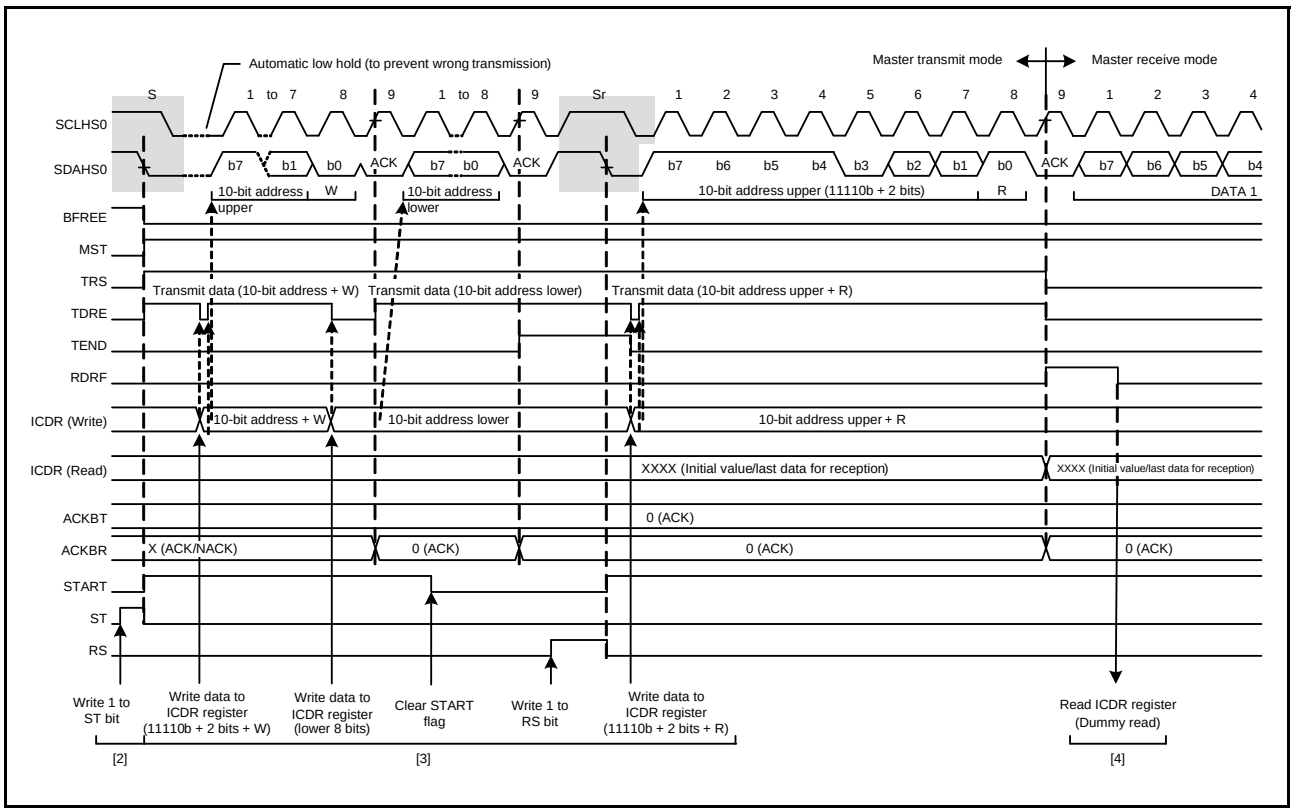


Figure 38.12 Master Receive Operation Timing (2) (10-Bit Address Format, When WAITAE = 0)

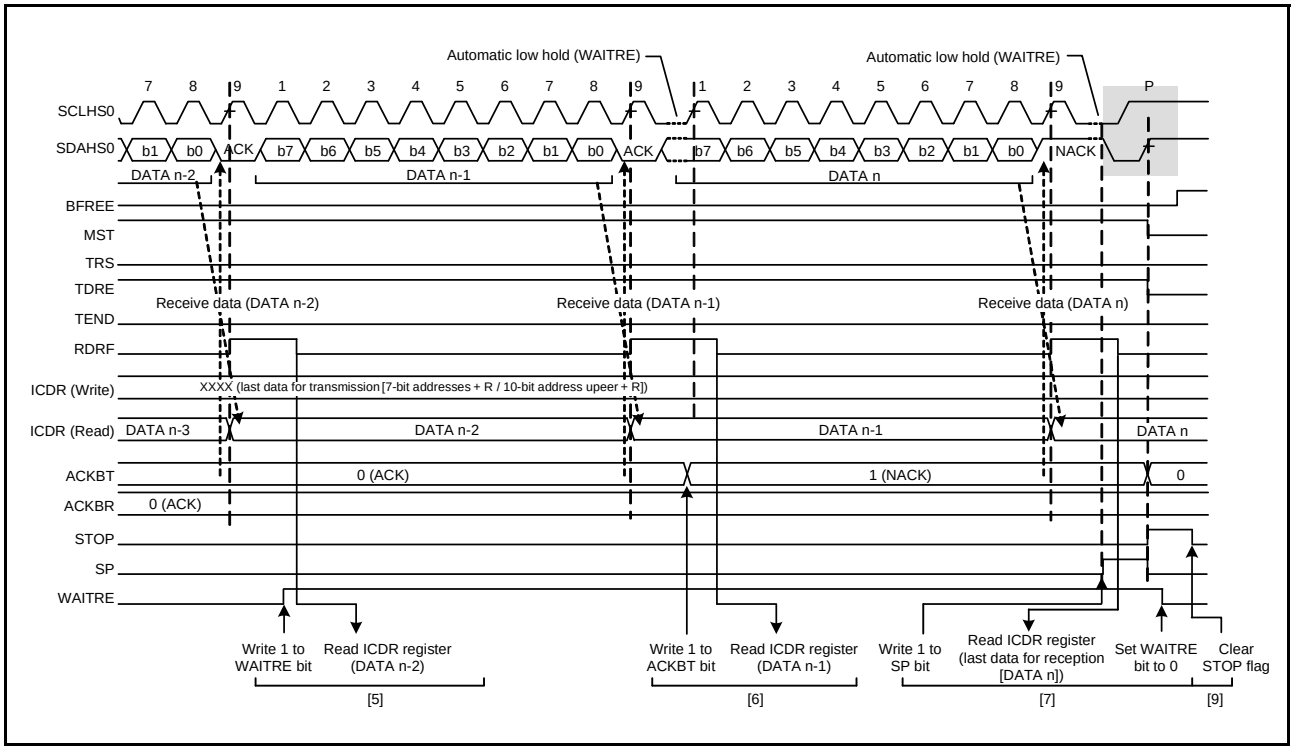


Figure 38.13 Master Receive Operation Timing (3) (When WAITAE = 0)

38.3.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL, the RIICHS transmits data as a slave device, and the master device returns acknowledgments.

Figure 38.14 shows an example of usage of slave transmission and Figure 38.15 and Figure 38.16 show the timing of operations in slave transmission.

The following describes the procedure and operations for slave transmission.

(1) Initial settings. For details, refer to section 38.3.2, Initial Settings.
After initial settings, the RIICHS will stay in the standby state until it receives a slave address that it matches.

(2) After receiving a matching slave address, the RIICHS sets one of the corresponding flags ICSSR.HOA, ICSSR.GCA, and ICSSR.AASy (y = 0 to 2) to 1 on the rising edge of the ninth SCL (the clock signal) and outputs the value set in the ICACKR.ACKBT bit to the acknowledgment bit on the ninth SCL. Set the ACKBT bit to 0 in advance. If the value of the R/W# bit that was also received at this time is 1, the RIICHS automatically places itself in slave transmit mode by setting both the ICMMR.TRS flag and the ICCSR.TDRE flag to 1.

(3) After the ICCSR.TDRE flag is confirmed to be 1, write the data for transmission to the ICDR register. At this time, if the RIICHS receives no acknowledgment from the master device (receives an NACK signal) while the ICSER.NAKDE bit is 1, the RIICHS suspends transfer of the next data.

(4) Wait until the ICSR2.TEND flag is set to 1 while the ICCSR.TDRE flag is 1, after the ICSR2.NACKF flag is set to 1 or the last byte for transmission is written to the ICDR register. When the ICSR2.NACKF flag or the TEND flag is 1, the RIICHS drives the SCLHS0 line low on the falling edge of ninth SCL.

(5) When the ICSR2.NACKF flag or the ICSR2.TEND flag is 1, dummy read ICDR register to complete the processing. This releases the SCLHS0 line.

(6) Upon detecting the STOP condition, the RIICHS automatically sets flags ICSSR.HOA, ICSSR.GCA, and ICSSR.AASy (y = 0 to 2), flags ICCSR.TDRE and ICSR2.TEND, and the ICMMR.TRS flag to 0, and enters slave receive mode.

(7) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and ICSR2.STOP flags to 0 for the next transfer operation.

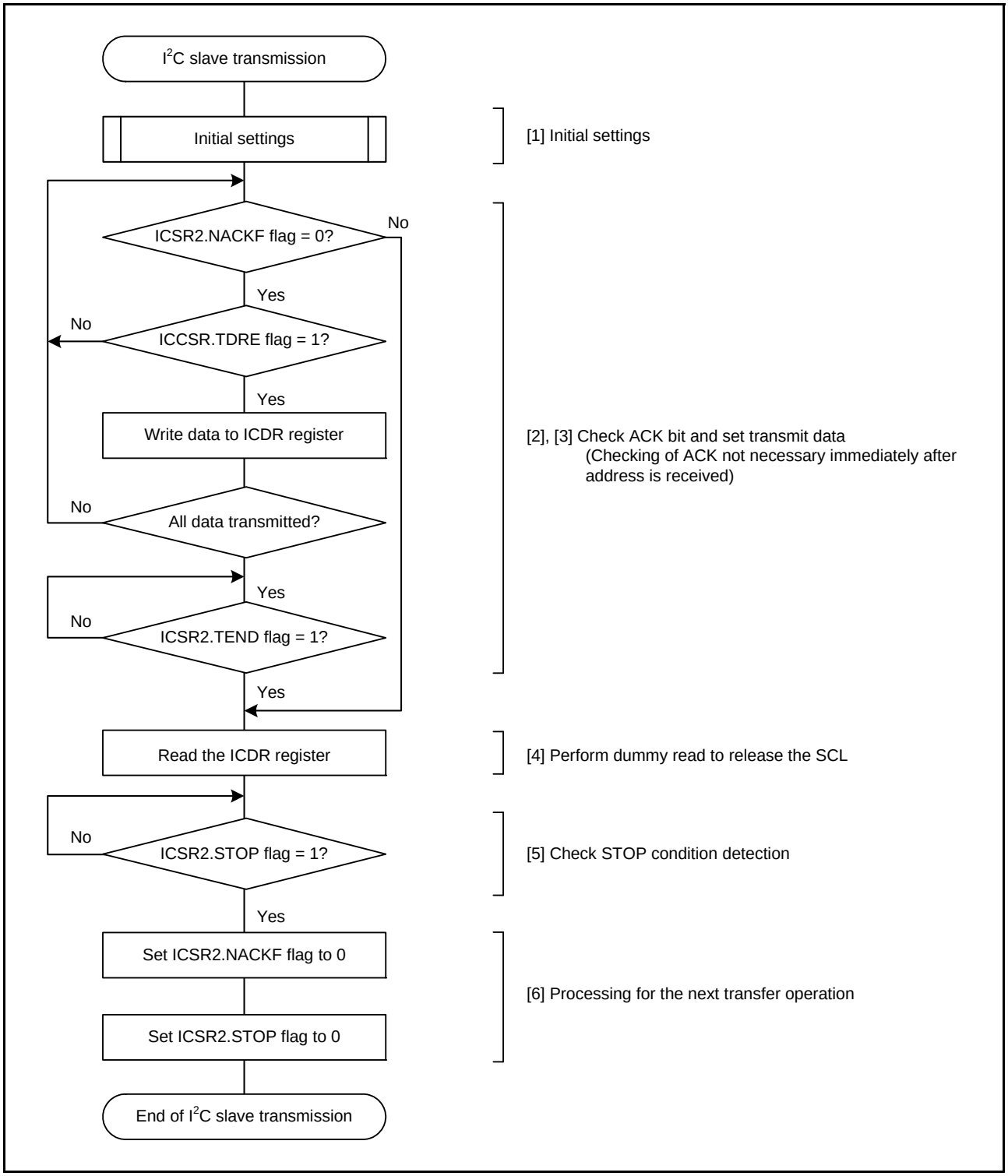


Figure 38.14 Example of I2C Slave Transmission Flowchart

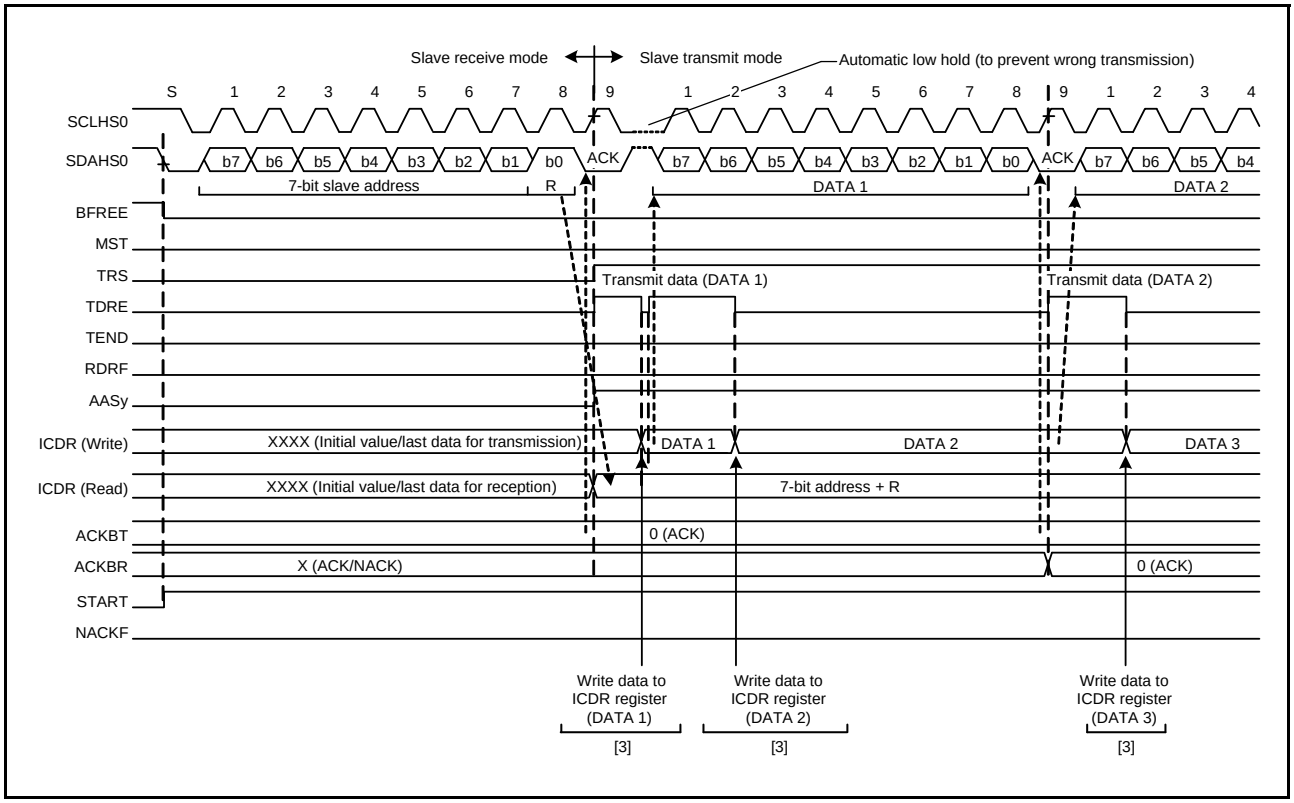


Figure 38.15 Slave Transmit Operation Timing (1) (7-Bit Address Format)

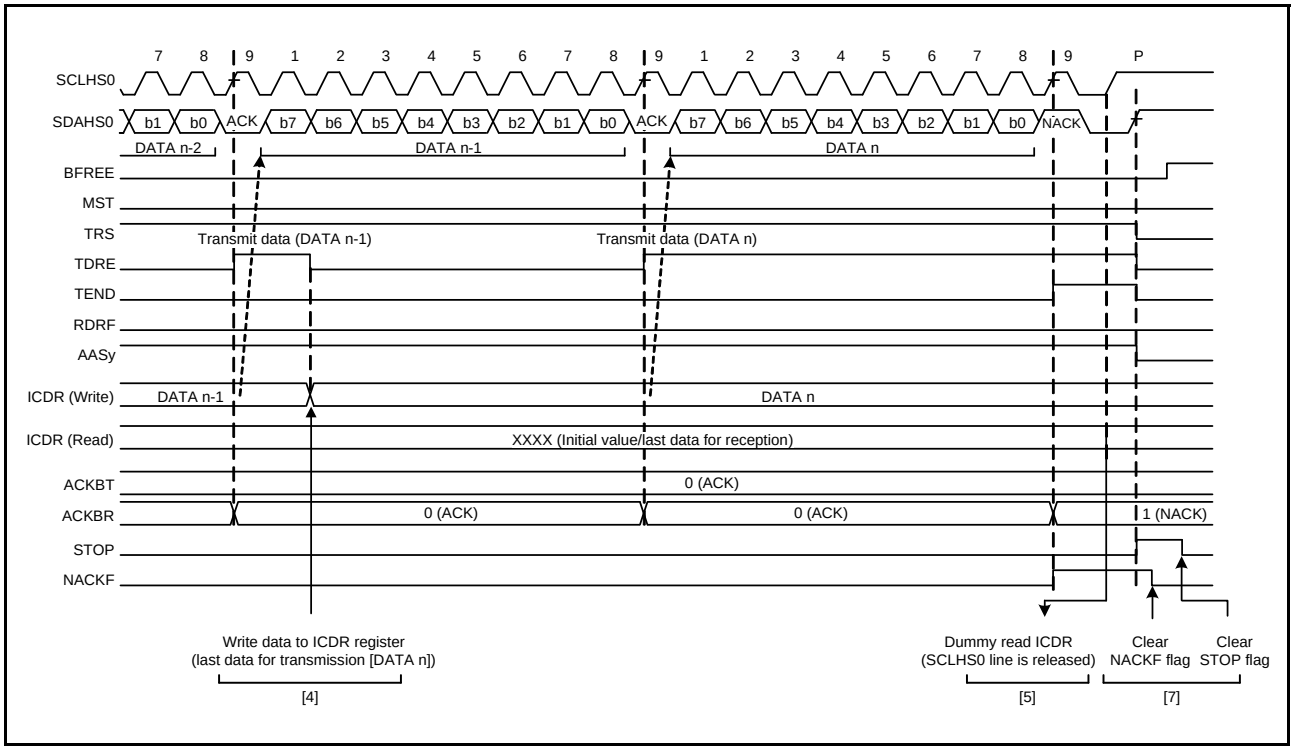


Figure 38.16 Slave Transmit Operation Timing (2)

38.3.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL and transmit data, and the RIICHS returns acknowledgments as a slave device.

Figure 38.17 shows an example of usage of slave reception and Figure 38.18 and Figure 38.19 show the timing of operations in slave reception.

The following describes the procedure and operations for slave reception.

(1) Initial settings. For details, refer to section 38.3.2, Initial Settings. After initial settings, the RIICHS will stay in the standby state until it receives a slave address that it matches.

(2) After receiving a matching slave address, the RIICHS sets one of the corresponding flags ICSSR.HOA, ICSSR.GCA, and ICSSR.AASy (y = 0 to 2) to 1 on the rising edge of the ninth SCL (the clock signal) and outputs the value set in the ICACKR.ACKBT bit to the acknowledgment bit on the ninth SCL. Set the ACKBT bit to 0 in advance. If the value of the R/W# bit that was also received at this time is 0, the RIICHS continues to place itself in slave receive mode and sets the ICCSR.RDRF flag to 1.

(3) After the ICSR2.STOP flag is confirmed to be 0 and the ICCSR.RDRF flag to be 1, dummy read ICDR register (the dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower 8-bits when the 10-bit address format is selected).

(4) When ICDR register is read, the RIICHS automatically sets the ICCSR.RDRF flag to 0. If reading of ICDR register is delayed and a next byte is received while the RDRF flag is still set to 1, the RIICHS holds the SCLHS0 line low from one SCL cycle before the timing with which RDRF should be set. In this case, reading ICDR register releases the SCLHS0 line from being held low.
When the ICSR2.STOP flag is 1 and the ICCSR.RDRF flag is also 1 or when all the data is completely received, read the ICDR register.

(5) Upon detecting the STOP condition, the RIICHS automatically sets flags ICSSR.HOA, ICSSR.GCA, and ICSSR.AASy (y = 0 to 2) to 0.

(6) After checking that the ICSR2.STOP flag is 1, set the ICSR2.STOP flag to 0 for the next transfer operation.

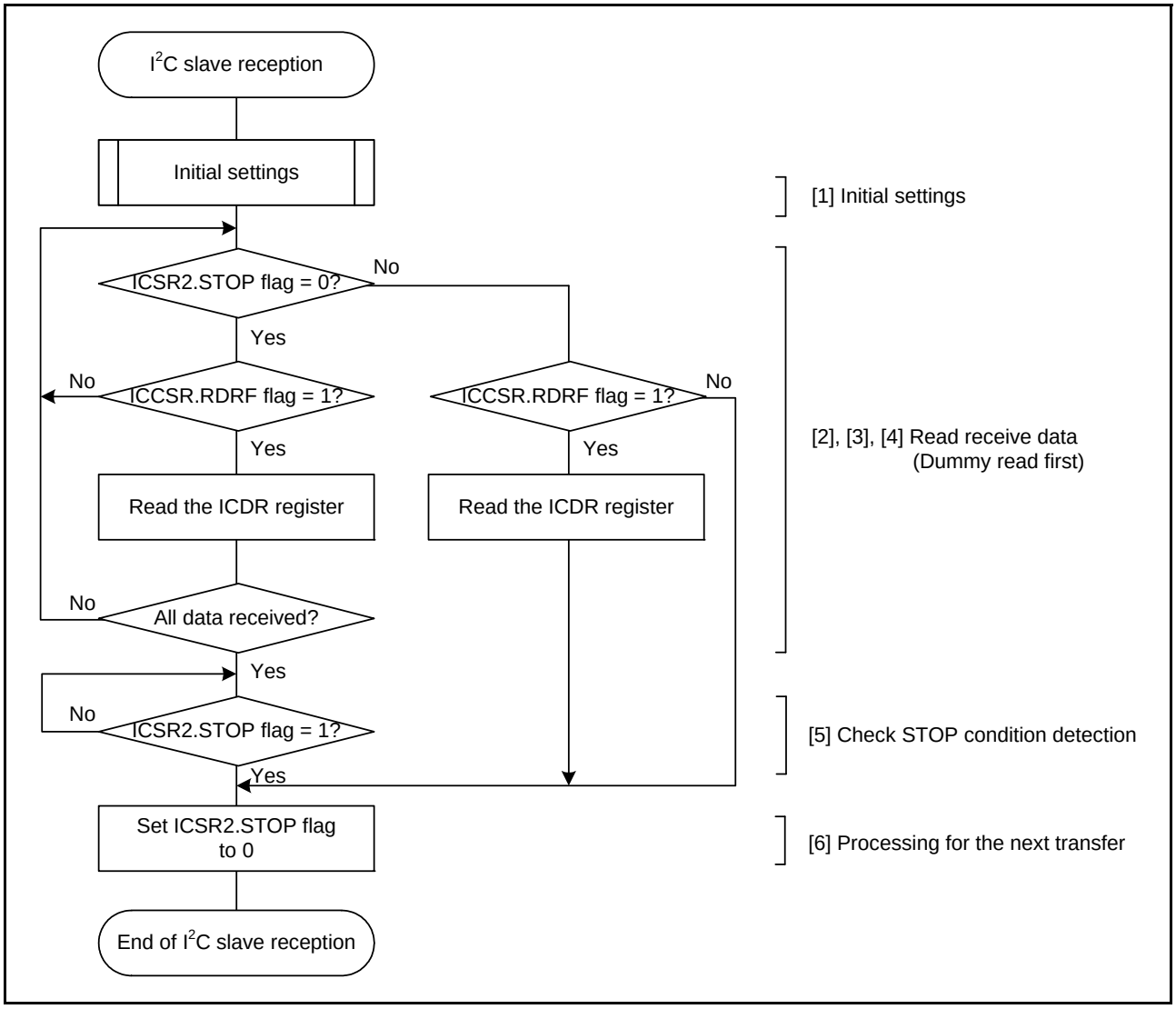


Figure 38.17 Example of I2C Slave Reception Flowchart

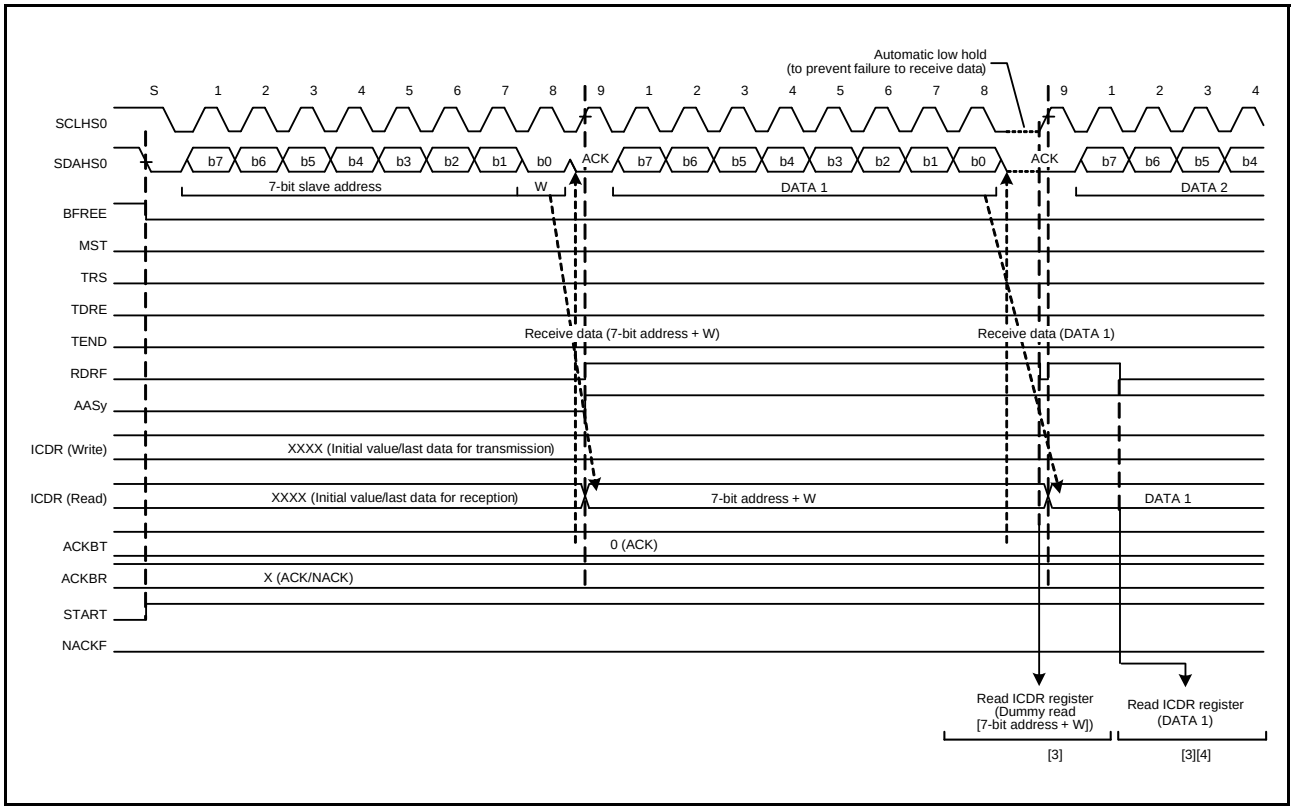


Figure 38.18 Slave Receive Operation Timing (1) (7-Bit Address Format, when WAITAE = 0)

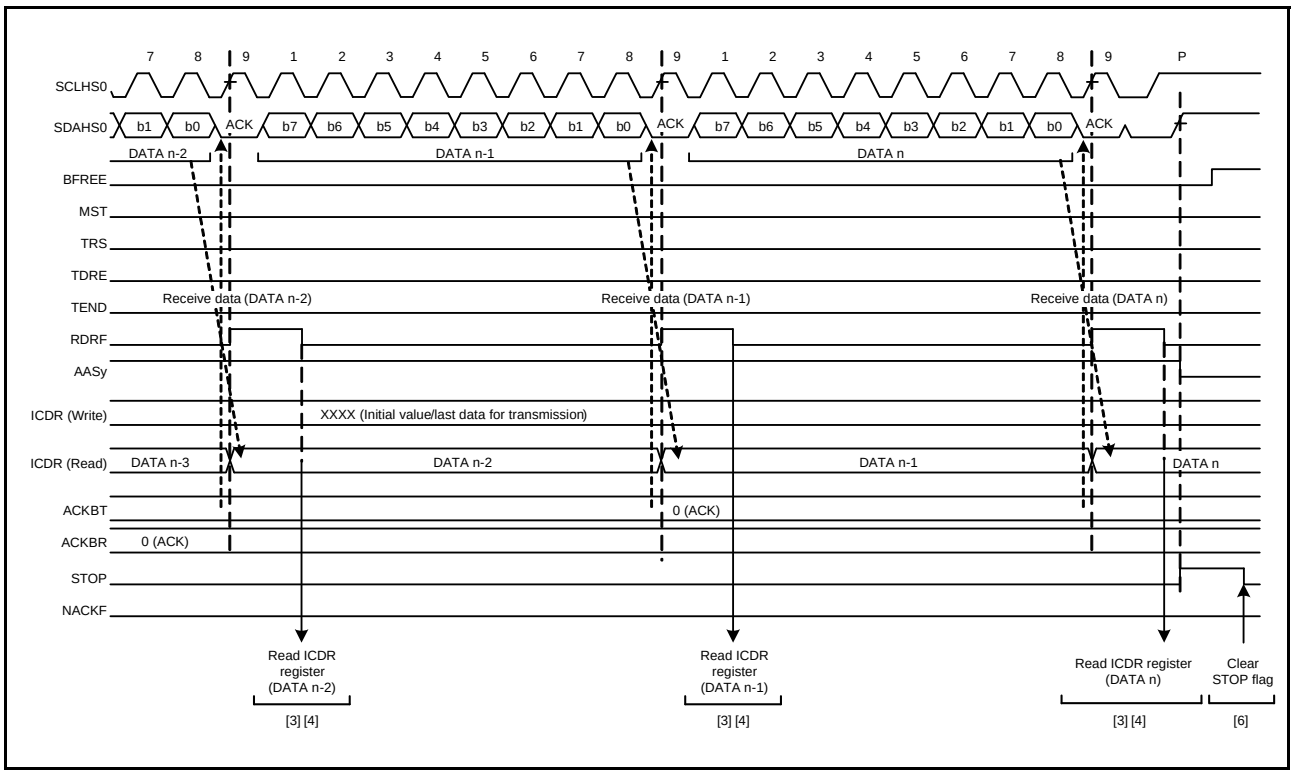


Figure 38.19 Slave Receive Operation Timing (2) (when WAITAE = 0)

38.4 SCL Synchronization Circuit

In generation of the SCL, the RIICHS starts counting out the value for width at high level specified in ICFBR.HIGH[7:0] bits when it detects a rising edge on the SCLHS0 line and drives the SCLHS0 line low once counting of the width at high level is complete. When the RIICHS detects the falling edge of the SCLHS0 line, it starts counting out the width at low period specified in ICFBR.LOW[7:0] bits, and then stops driving the SCLHS0 line (releases the line) once counting of the width at low level is complete. The SCL 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 RIICHS is equipped with a function (the SCL synchronization circuit) to obtain bit-by-bit synchronization of the SCL signals by monitoring the SCLHS0 line while in master mode. When the RIICHS has detected a rising edge on the SCLHS0 line and thus started counting out the width at high level specified in ICFBR.HIGH[7:0] bits, and the level on the SCLHS0 line falls because an SCL signal is being generated by another master device, the RIICHS stops counting when it detects the falling edge, drives the level on the SCLHS0 line low, and starts counting out the width at low level specified in ICFBR.LOW[7:0] bits. When the RIICHS finishes counting out the width at low level, it stops driving the SCLHS0 line to low (i.e. releases the line). At this time, if the width at low level of the SCL signal from the other master device is longer than the width at low level set in the RIICHS, 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 SCLHS0 line has been released. When the RIICHS finishes outputting the low period of the SCL, the SCLHS0 line is released and the SCL rises. That is, in cases of contention of SCL signals from more than one master, the width at high level of the SCL signal is synchronized with that of the clock having the narrower width, and the width at low level of the SCL signal is synchronized with that of the clock having the broader width. However, such synchronization of the SCL signal is only enabled when the ICFER.SCLE bit is set to 1.

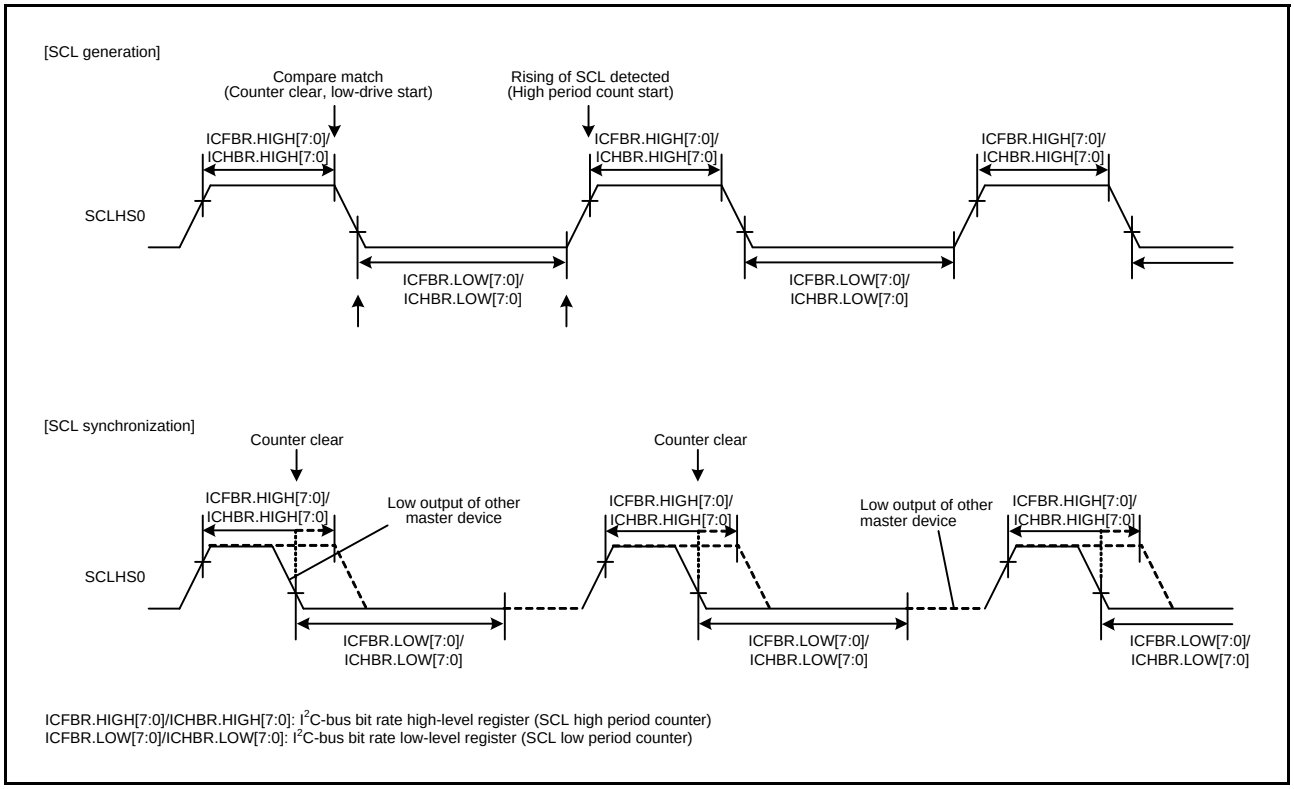


Figure 38.20 Generation and Synchronization of the SCL Signal

38.5 SDA Output Delay Function

The RIICHS module incorporates a facility for delaying output on the SDA line. The delay can be applied to all output (generating of the START, repeated START, and STOP conditions, data, and the ACK and NACK signals) on the SDA line.

The SDA output delay function is used to delay the SDA output timing from a falling edge of SCL to ensure that the SDA signal changes while the SCL is low and can be used to prevent erroneous operation of communications devices. This function is also used to satisfy the 300 ns (min.) of data hold time specified by the SMBus specification.

The output delay facility is enabled by setting the ICOCR.SDDL[2:0] bits to any value other than 000b, and disabled by setting the same bits to 000b.

When the SDA output delay function is enabled (the SDDL[2:0] bits are not 000b), the SDA output delay counter counts the number of cycles set in the SDDL[2:0] bits with the count source selected by the ICOCR.DLCS bit (internal reference clock (IICφ) or internal reference clock divided by 2 (IICφ/2)). When the counting of the delay cycles is completed, the RIICHS output the SDA signal (generating of the START, repeated START, or STOP condition, data, or an ACK or NACK signal).

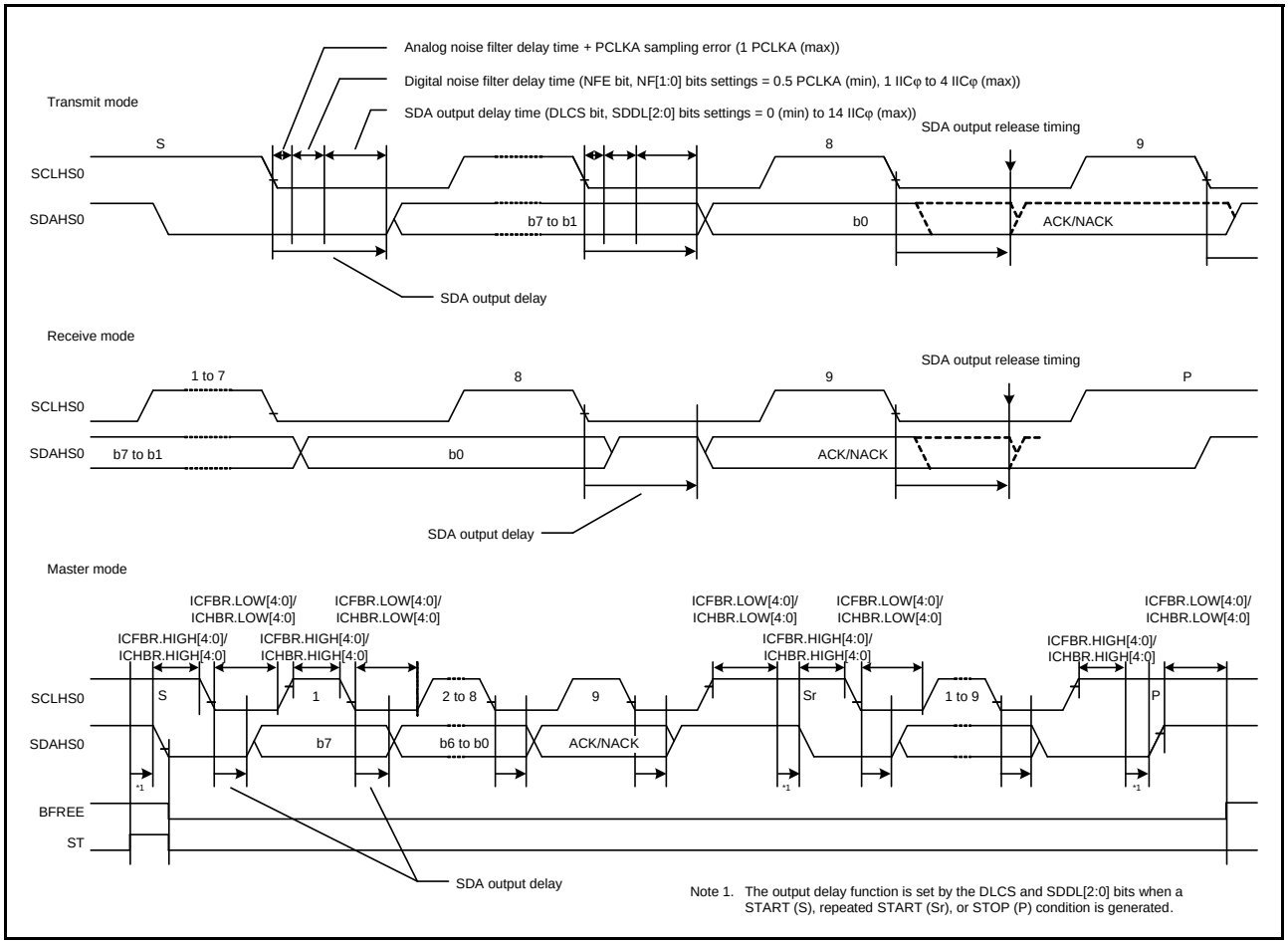


Figure 38.21 SDA Output Delay Function

38.6 Digital Noise Filters

The states of the SCLHS0 and SDAHS0 pins are conveyed to the internal circuitry through analog noise filter and digital noise filter. Figure 38.22 is a block diagram of the digital noise filter.

The on-chip digital noise filter of the RIICHS consists of 16 flip-flop circuit stages connected in series and a match-detection circuit. When Hs-mode is selected, only the first four flip-flops are enabled.

The number of effective stages in the digital noise filter is selected by the ICICR.NF[3:0] bits (ICICR.NF[3:2] bits in Hs-mode). The selected number of effective stages determines the noise-filtering capability as a period from 1 to 16 IICφ cycles (one to four IICφ cycles in Hs-mode).

The input signal from the SCLHS0 pin (also the SDAHS0 pin) is sampled on rising 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 ICICR.NF[3:0] bits, the signal level is conveyed to the subsequent stage. If the signal levels do not match, the previous value is retained.

If the ratio between the frequency of the internal operating clock (PCLKA) and the transfer rate is small (e.g. data transfer at 400 kbps with PCLKA = 4 MHz), the characteristics of the digital noise filter may lead to the elimination of needed signals as noise. In such cases, disable the digital noise filter (by setting the ICICR.NFE bit to 0) and use only the analog noise filter.

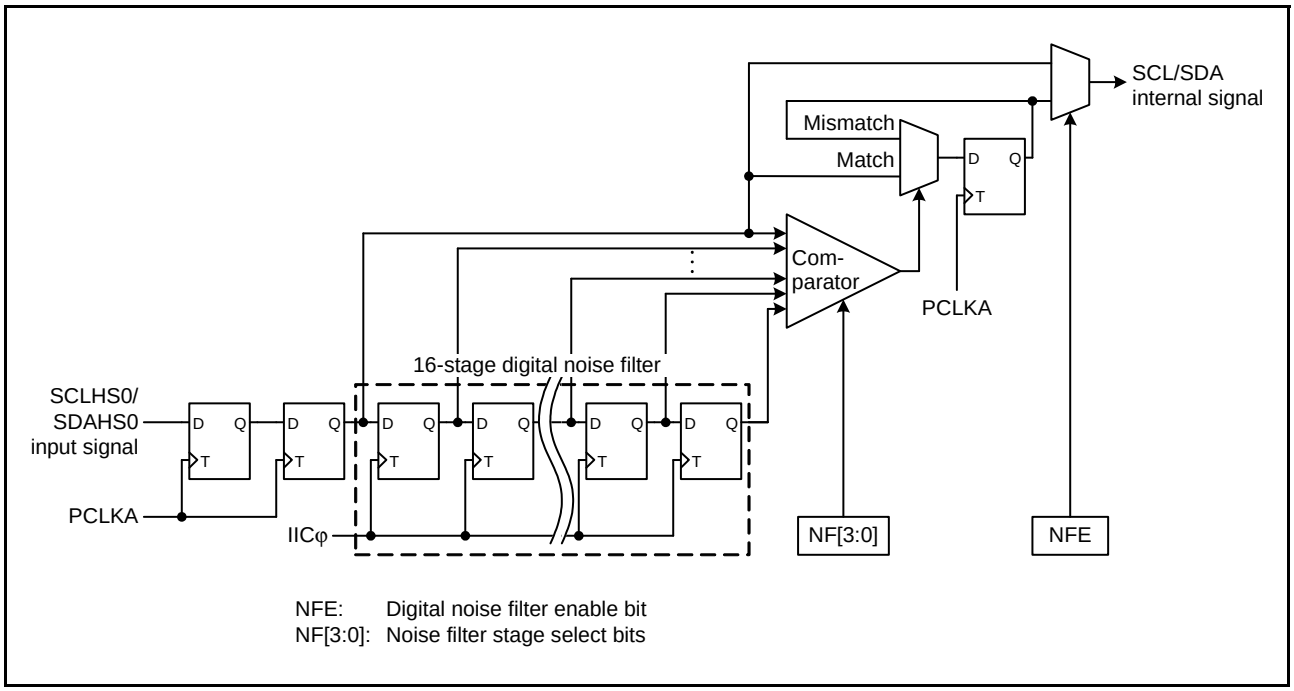


Figure 38.22 Block Diagram of Digital Noise Filter

38.7 Address Match Detection

The RIICHS 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.

38.7.1 Slave-Address Match Detection

The RIICHS can set three unique slave addresses, and has a slave address detection function for each unique slave address.

When the ICSCR.SARyE bit (y = 0 to 2) is set to 1, the slave addresses set in SAMRy register (y = 0 to 2) can be detected.

When the RIICHS detects a match of the set slave address, the corresponding ICSSR.AASy flag (y = 0 to 2) is set to 1 at the rising edge of the ninth SCL, and the ICCSR.RDRF flag or the ICCSR.TDRE flag is set to 1 according to the R/W# bit. This causes a receive data full interrupt (RXI) or transmit data empty interrupt (TXI) to be generated. The AASy flag is used to identify which slave address has been specified.

Figure 38.23 to Figure 38.25 show the AASy flag set timing in three cases.

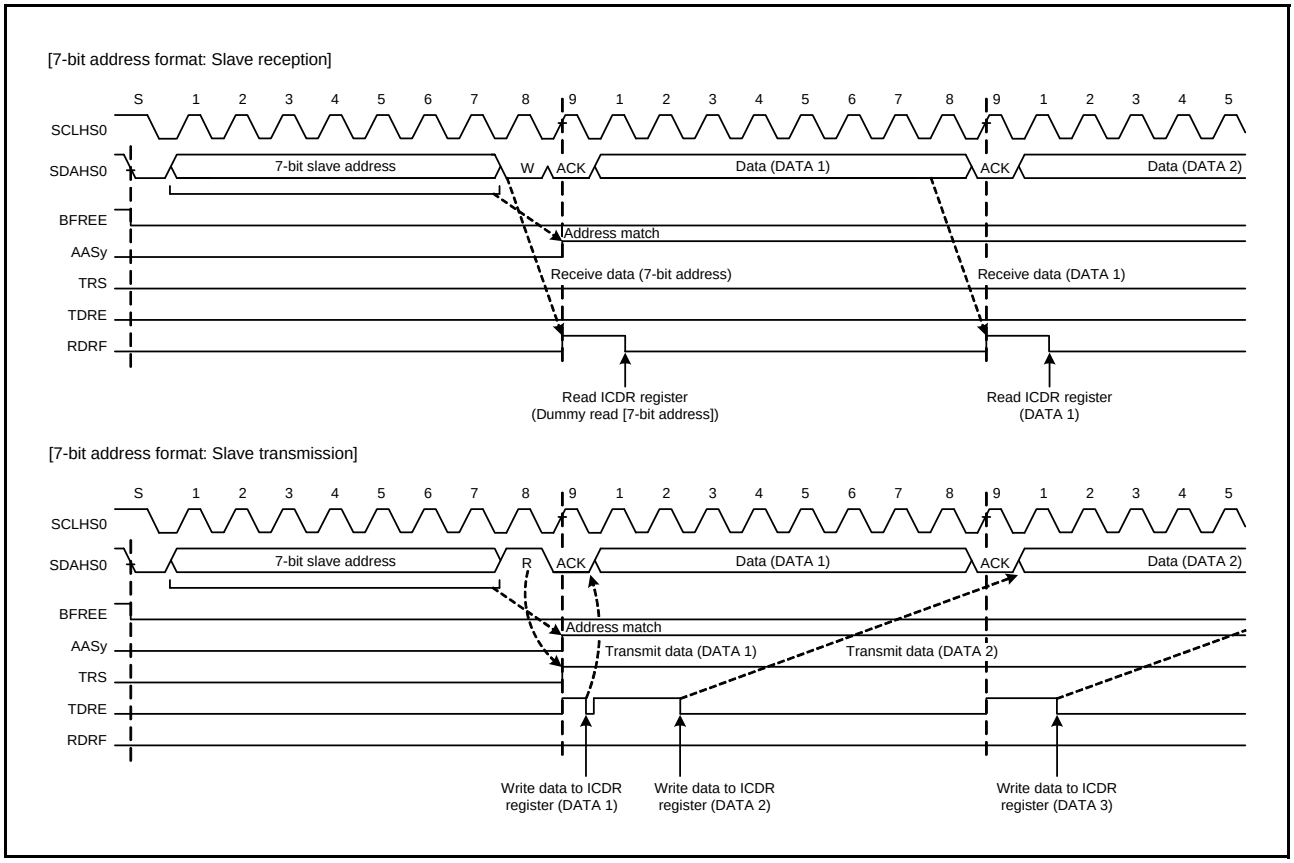


Figure 38.23 AASy Flag Set Timing with 7-Bit Address Format Selected

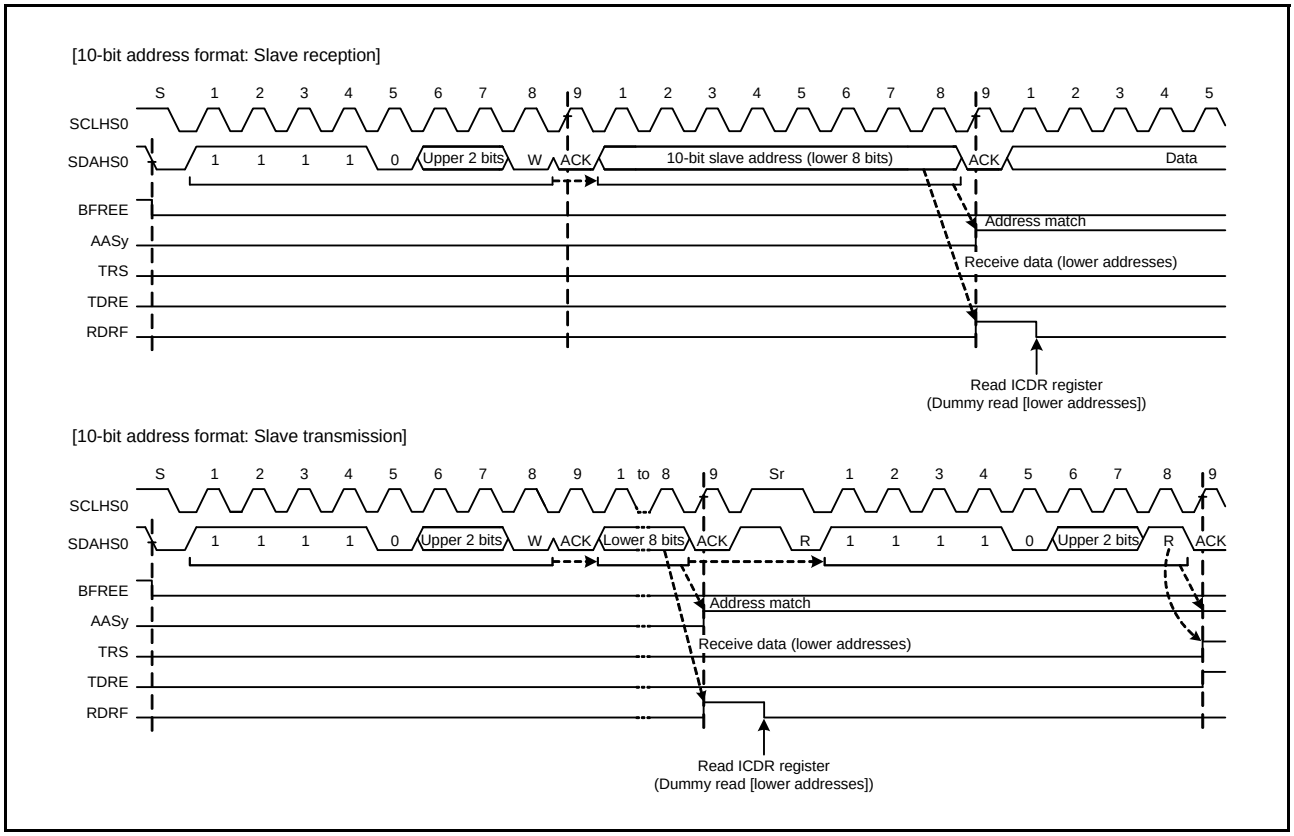


Figure 38.24 AASy Flag Set Timing with 10-Bit Address Format Selected

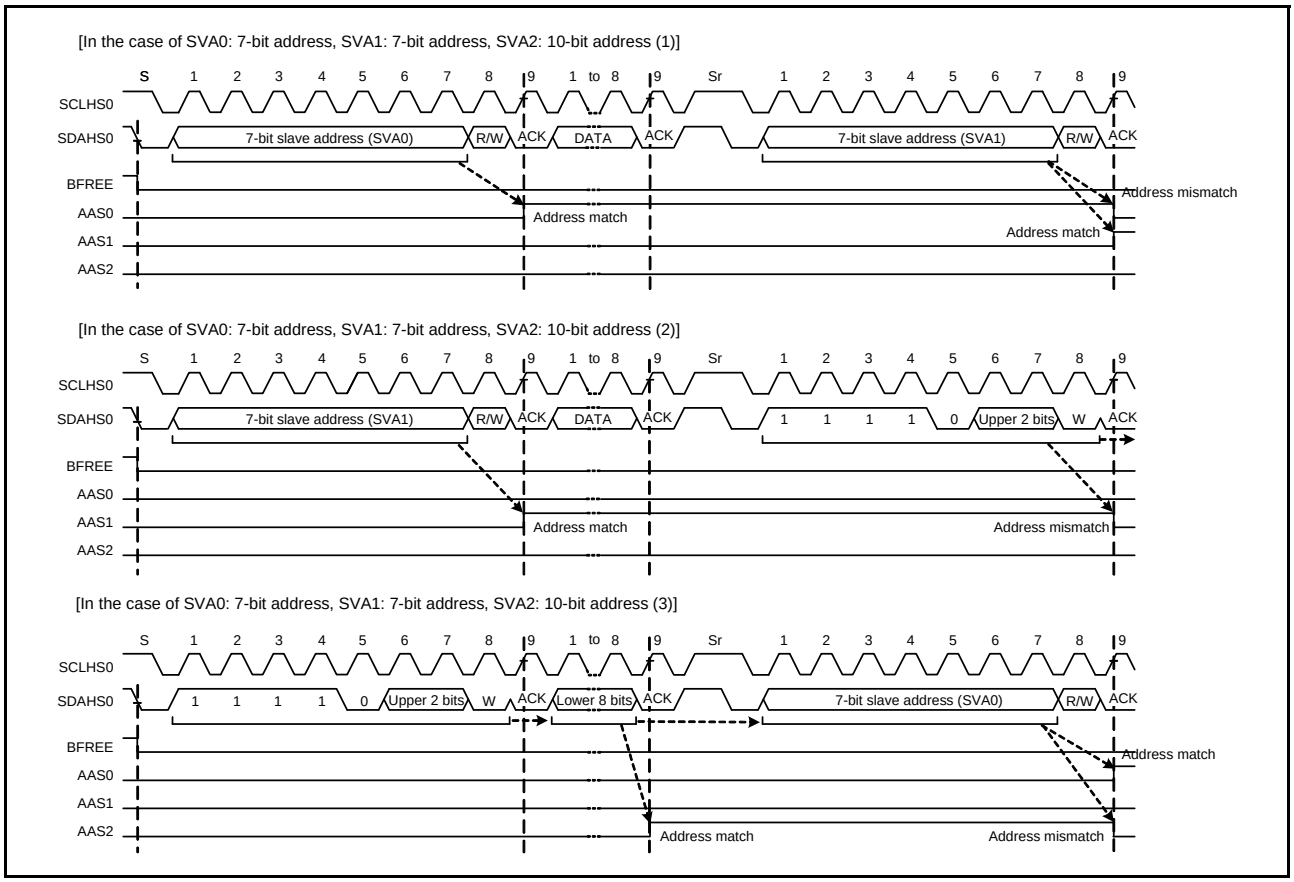


Figure 38.25 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed

38.7.2 General Call Address Detection

The RIICHS has a function for detecting the general call address (0000 000b + 0 (write)). This is enabled by setting the ICSCR.GCAE bit to 1.

If the address following a START or repeated START condition is 0000 000b + 1 (read) (start byte), the RIICHS recognizes this as the address of a slave device with an “all-zero” address but not as the general call address.

When the RIICHS detects the general call address, both the ICSSR.GCA flag and the ICCSR.RDRF flag are set to 1 on the rising edge of the ninth SCL. This leads to the generation of a receive data full interrupt (RXI). The value of the GCA flag can be confirmed to recognize that the general call address has been transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

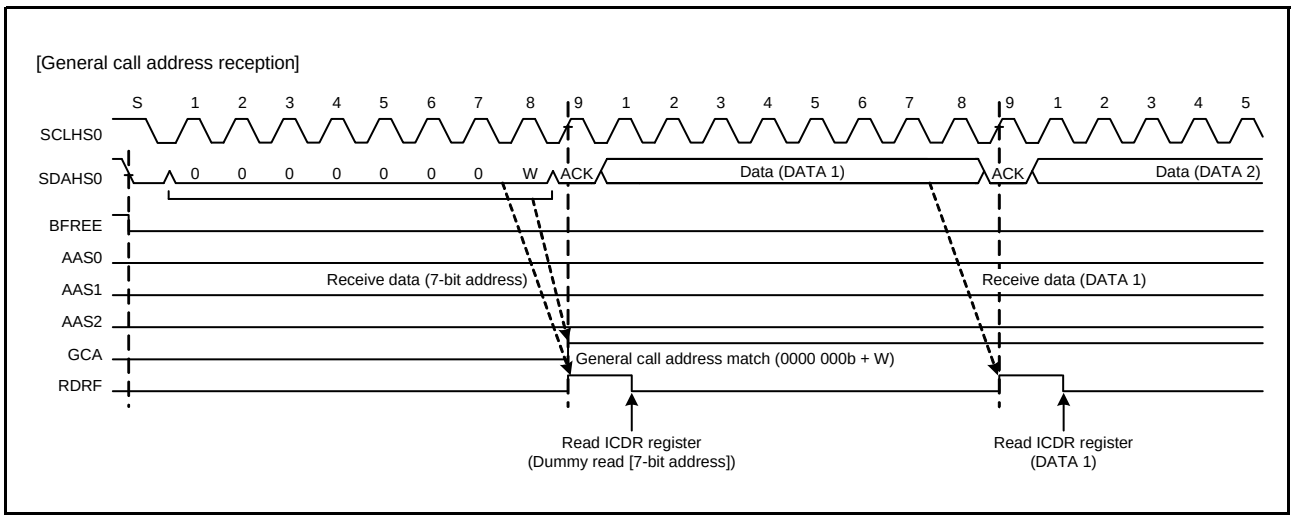


Figure 38.26 Timing of GCA Flag Setting during Reception of General Call Address

38.7.3 Device-ID Address Detection

The RIICHS module has a function for detecting device-ID addresses conformant with the I2C-bus specification.
When the RIICHS receives 1111 100b as the first seven bits of the first byte following a START condition or repeated START condition while the ICSCR.DIDE bit is 1, the RIICHS recognizes the address as a device ID address, sets the ICSSR.DID flag to 1 on the rising edge of the ninth SCL when the following R/W# bit is 0, and then compares the second and following bytes with its own slave address. If the received address matches the value in the slave address register, the RIICHS sets the corresponding ICSSR.AASy flag (y = 0 to 2) to 1.
After that, when the first byte received after a START or repeated START condition is generated matches the device ID address (1111 100b) again and the following R/W# bit is 1, the RIICHS does not compare the second and subsequent bytes and sets the ICCSR.TDRE flag to 1.
In the device-ID address detection function, the RIICHS sets the DID flag to 0 if a match with the RIICHS's own slave address is not obtained or a match with the device ID address is not obtained after a match with the RIICHS's own slave address and the detection of a repeated START condition. If the first byte after detection of a START or repeated START condition matches the device ID address (1111 100b) and the R/W# bit is 0, the RIICHS sets the DID flag to 1 and compares the second and subsequent bytes with the RIICHS's slave address. If the R/W# bit is 1, the DID flag holds the previous value and the RIICHS 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 ICCSR.TDRE flag = 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.

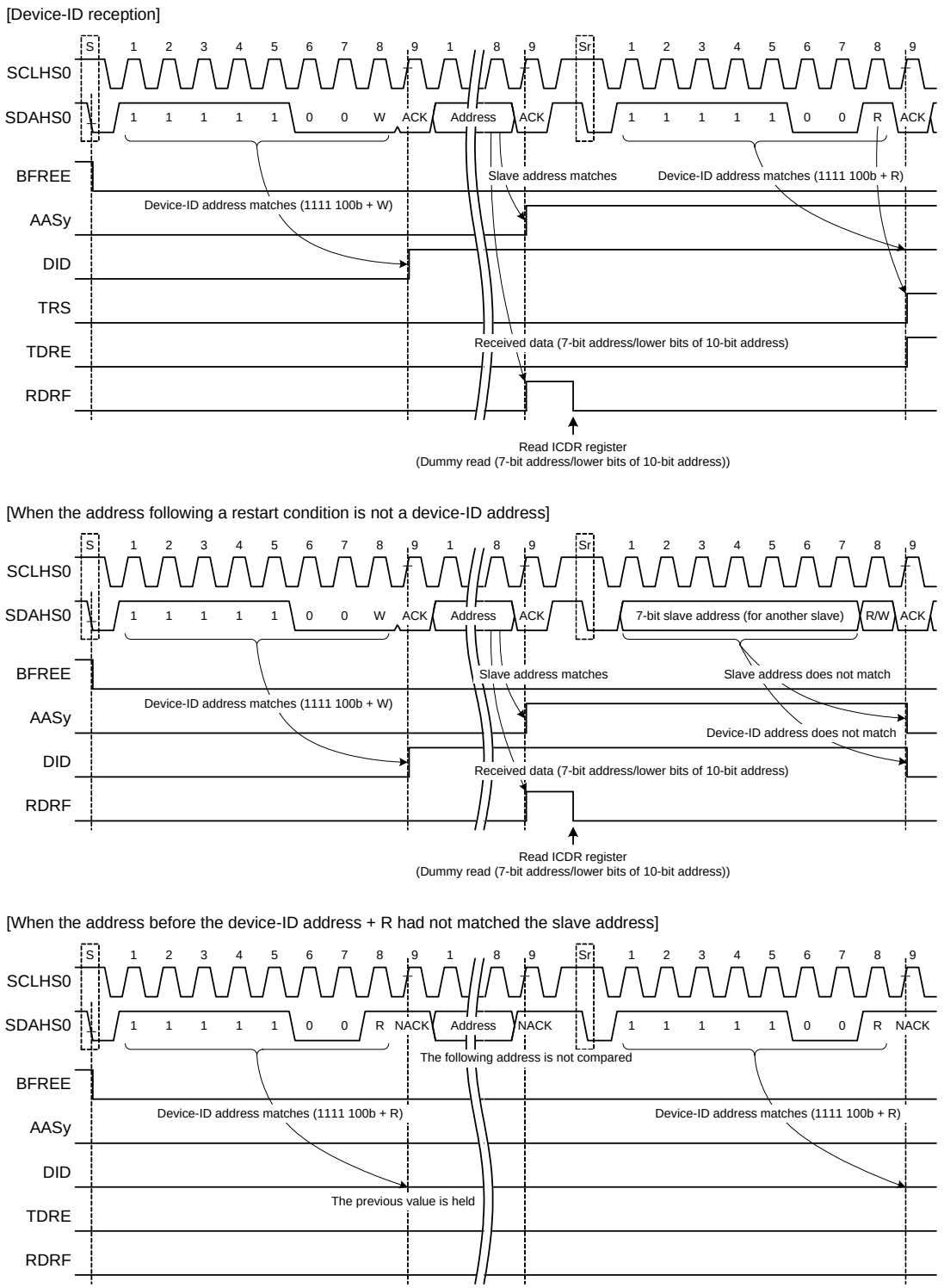


Figure 38.27 Set/Clear Timing of the AASy and DID Flags during Reception of Device-ID Address

38.7.4 Hs-Mode Master Code Detection

The RIICHS has a function to detect the Hs-mode master code (0000 1XXXb).
When the RIICHS receives 0000 1XXXb as the first byte following a START condition while the ICSCR.HSMCE bit is 1, the RIICHS recognizes it as Hs-mode master code and set the ICSSR.HSMC flag to 1 at the rising edge of the ninth SCL. The RIICHS does not acknowledge the master code regardless of the value of the ICACKR.ACKBT bit (the SDAHS0 pin remains high).
The first byte following a repeated START condition after the NACK bit is recognized as the slave address and compared with the SAMRy.SVA[9:0] bits (y = 0 to 2). When they match, the corresponding ICSSR.AASy flag is set to 1 at the rising edge of the ninth SCL. Also, the ICCSR.RDRF or TDRE flag is set to 1 depending on the value of the R/W# bit.
When a STOP condition is detected, the ICSSR.HSMC flag is set to 0.

Note: If the RIICHS receives Hs-mode master code while the ICSCR.HSMCE bit is 0, it ignores all data patterns until a STOP condition is detected.

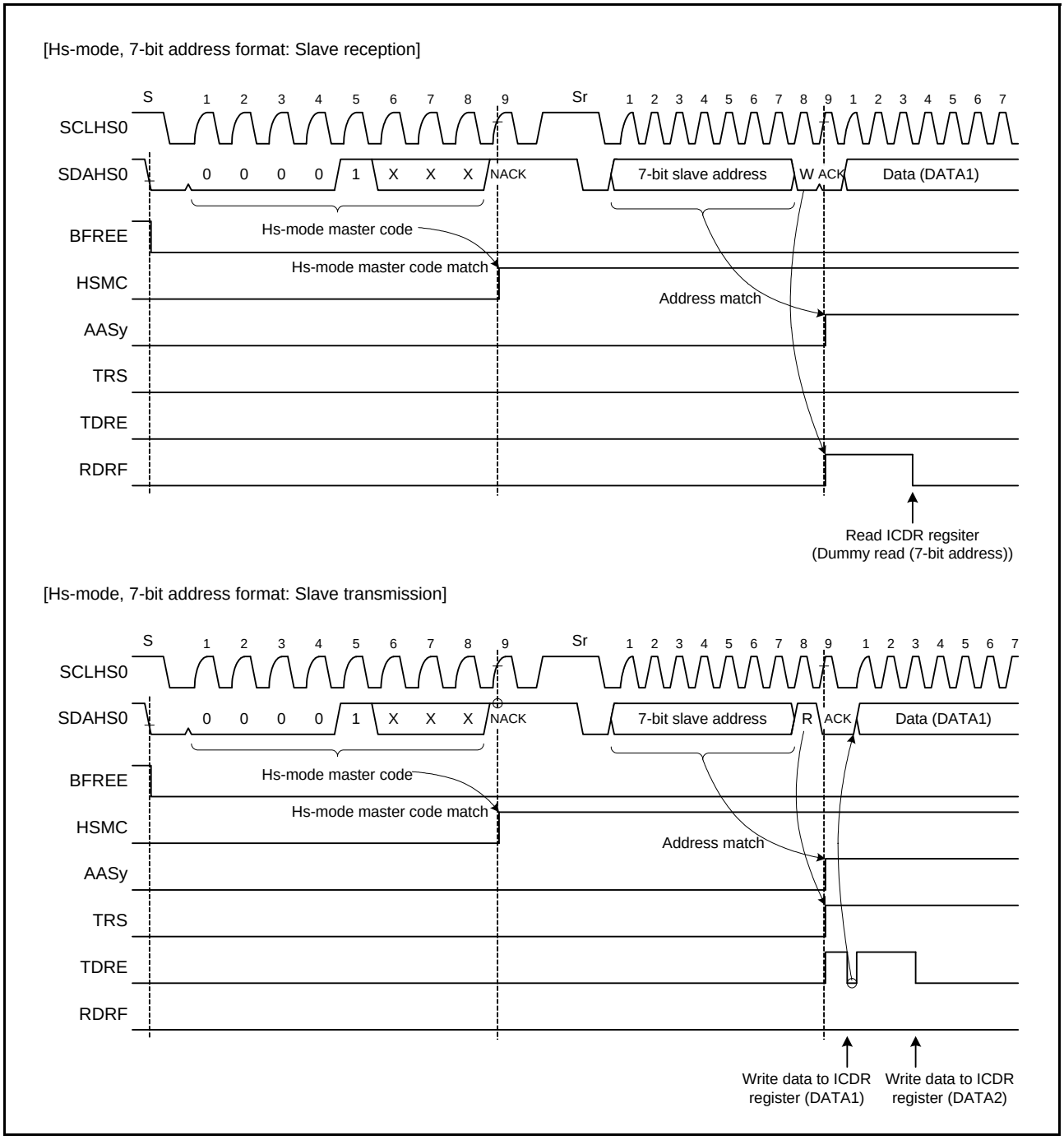


Figure 38.28 Set Timing of the HSMC and AASy Flags during Reception of Hs-Mode Master Code

38.7.5 Host Address Detection

The RIICHS has a function to detect the host address while the SMBus is operating. When the ICSCR.HOAE bit is set to 1 while the ICFER.SMBS bit is 1, the RIICHS can detect the host address (0001 000b) in slave receive mode (ICMMR.MST and ICMMR.TRS flags = 00b).

When the RIICHS detects the host address, the ICSSR.HOA flag is set to 1 at the rising edge of the ninth SCL, and at the same time, the ICCSR.RDRF flag is set to 1 when the R/W# bit is 0 (write bit). This causes a receive data full interrupt (RXI) to be generated. The HOA flag is used to recognize that the host address was sent from the smart battery or other devices.

If the bit following the host address (0001 000b) is an Rd bit (R/W# bit = 1), the RIICHS can also detect the host address. After the host address is detected, the RIICHS operates in the same manner as normal slave operation.

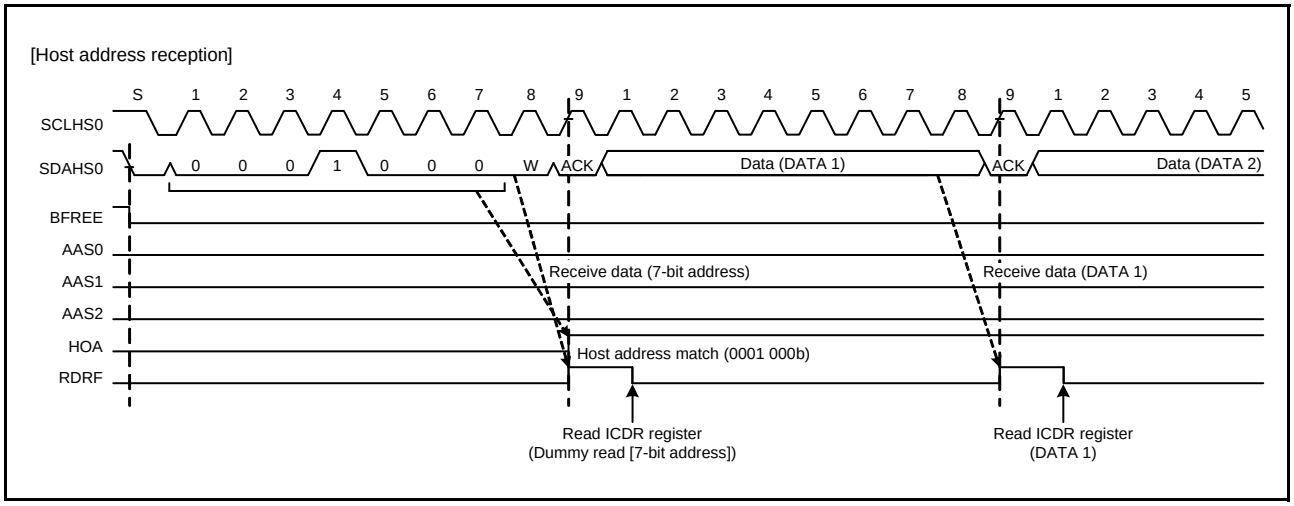


Figure 38.29 HOA Flag Set Timing during Reception of Host Address

38.8 Automatic Low-Hold Function for SCL

38.8.1 Function to Prevent Wrong Transmission of Transmit Data

When data have not been written to the transmit data register (ICDR register) with the RIICHS in transmission mode (ICMMR.TRS flag = 1), the SCLHS0 line is automatically held low over the periods shown below. This low period is extended until data for transmission have been written, which prevents the unintended transmission of erroneous data.

Master transmit mode

- Low period after a START condition or repeated START condition is generated
- Low period between the ninth clock pulse of one transfer and the first clock pulse of the next

Slave transmit mode

- Low period between the ninth clock pulse of one transfer and the first clock pulse of the next

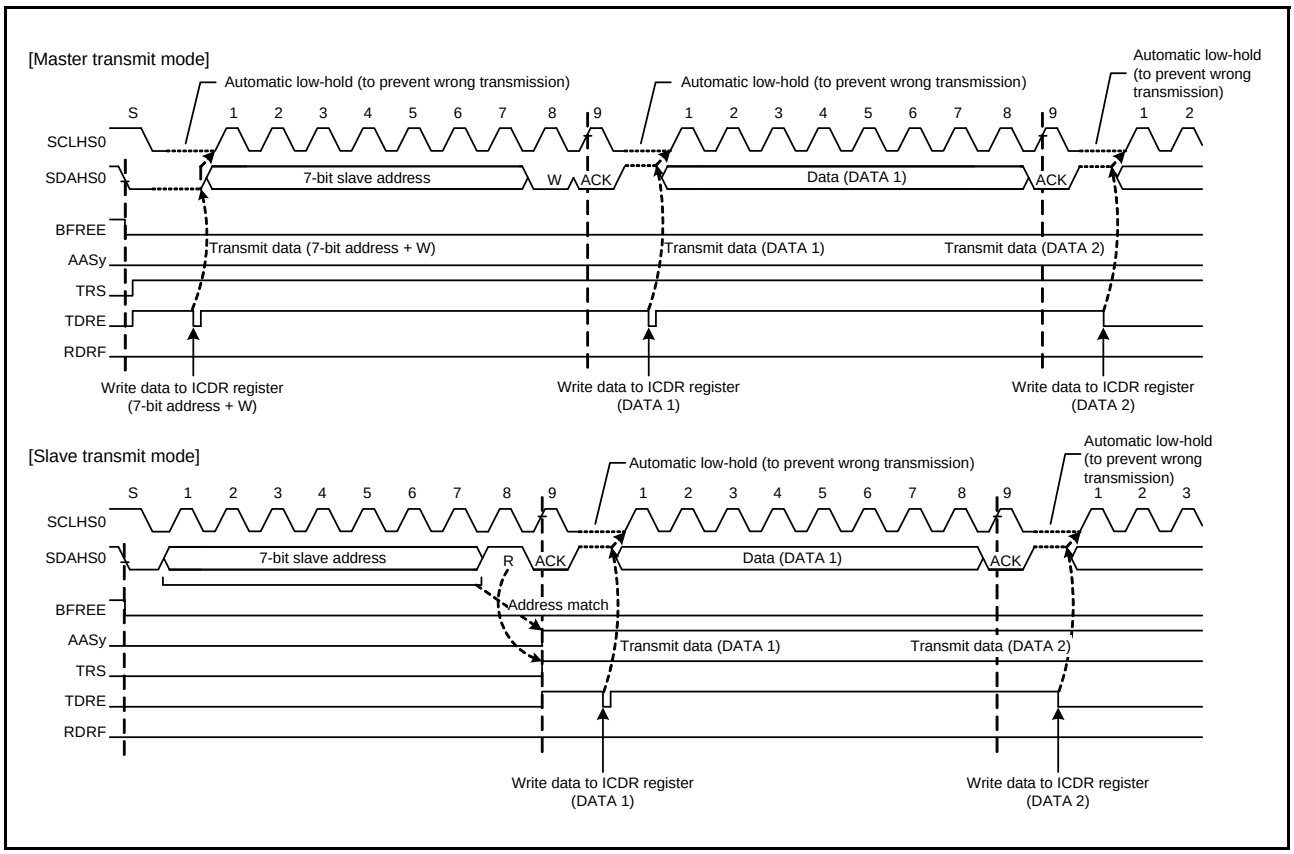


Figure 38.30 Automatic Low-Hold Operation in Transmit Mode

38.8.2 NACK Reception Transfer Suspension Function

The RIICHS has a function to suspend transfer operation when NACK is received in transmit mode (ICMMR.TRS flag = 1).

This function is enabled when the ICSER.NAKDE bit is set to 1 (transfer suspension enabled). If the next transmit data has already been written (ICCSR.TDRE flag = 0) when NACK is received, next data transmission at the falling edge of the ninth SCL is automatically suspended. This prevents the SDAHS0 line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is suspended by this function (ICSR2.NACKF flag = 1), transmit operation and receive operation are discontinued. To restore transmit/receive operation, be sure to set the NACKF flag to 0. In master transmit mode, issue a repeated START or STOP condition, set the NACKF flag to 0, and then issue a START condition again.

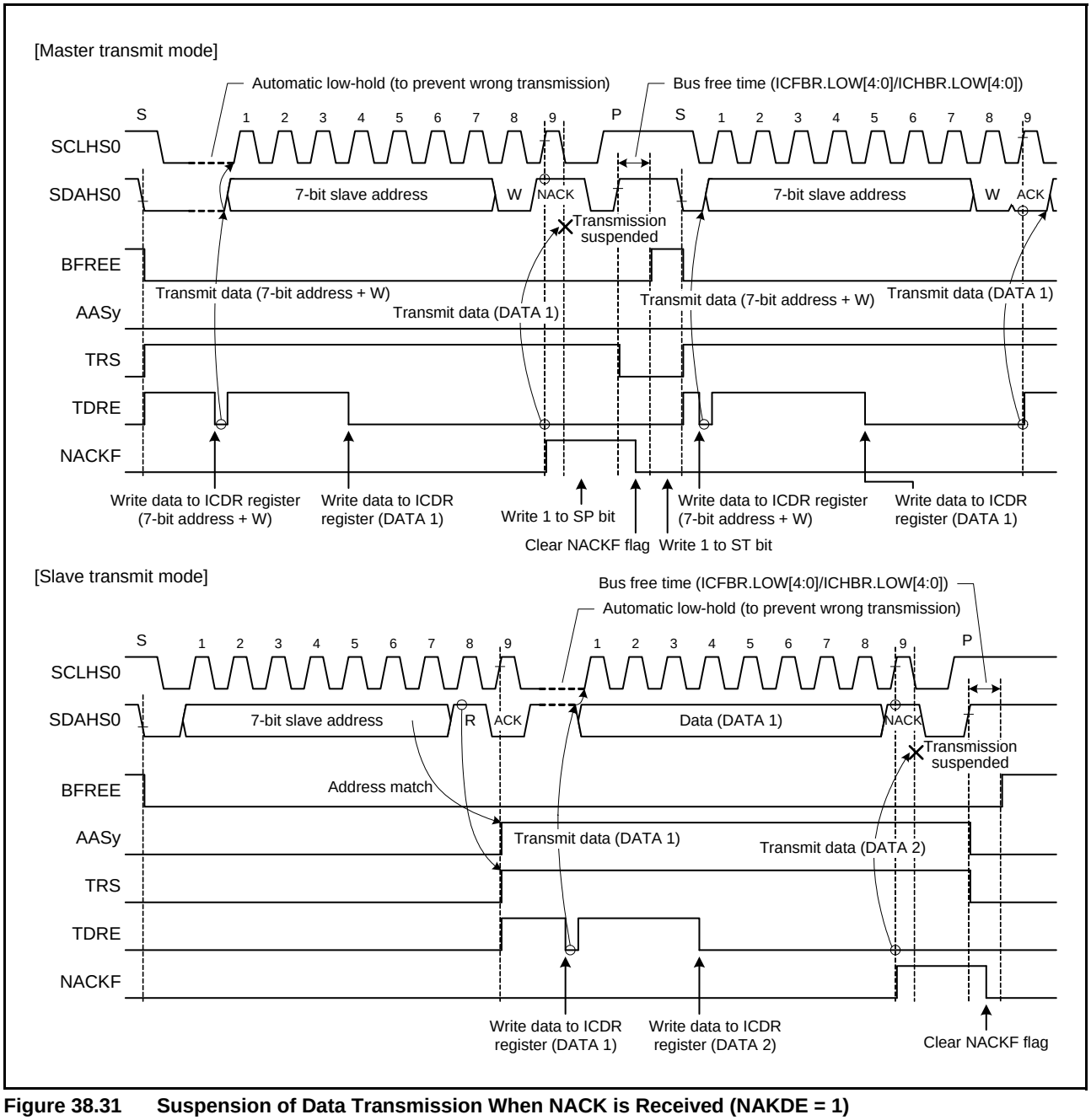


Figure 38.31 Suspension of Data Transmission When NACK is Received (NAKDE = 1)

38.8.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (ICDR register) read is delayed for a period of one transfer frame or more with receive data full (ICCSR.RDRF flag = 1) in receive mode (ICMMR.TRS flag = 0), the RIICHS holds the SCLHS0 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 RIICHS's own slave address or a slave address for the other device is designated after a STOP condition is generated.
Sections in which the SCLHS0 line is held low can be selected with a combination of the ICCSCR.WAITRE and ICCSCR.WAITAE bits.

(1) 1-Byte Receive Operation and Automatic Low-Hold Function Using the WAITRE Bit

When the ICCSCR.WAITRE bit is set to 1, the RIICHS performs 1-byte receive operation using the WAITRE bit function.
Furthermore, when the ICCSCR.WAITAE bit is 0, the RIICHS automatically sends the ACKBT bit value in ICACKR register for the acknowledgment bit in the period from the falling edge of the eighth SCL to the falling edge of the ninth SCL, and automatically holds the SCLHS0 line low at the falling edge of the ninth SCL using the WAITRE bit function. This low-hold is released by reading data from ICDR register, which enables bytewise receive operation.
The WAITRE bit function is enabled for receive frames after a match with the RIICHS's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

(2) 1-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the WAITAE Bit

When the ICCSCR.WAITAE bit is set to 1, the RIICHS performs 1-byte receive operation using the WAITAE bit function.
When the WAITAE bit is set to 1, the ICCSR.RDRF flag (receive data full) is set to 1 at the rising edge of the eighth SCL, and the SCLHS0 line is automatically held low at the falling edge of the eighth SCL. This low-hold is released by writing a value to the ICACKR.ACKBT bit, but cannot be released by reading data from ICDR register, which enables receive operation by the ACK/NACK transmission control according to the data received in byte units.
The WAITAE bit function is enabled for receive frames after a match with the RIICHS's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

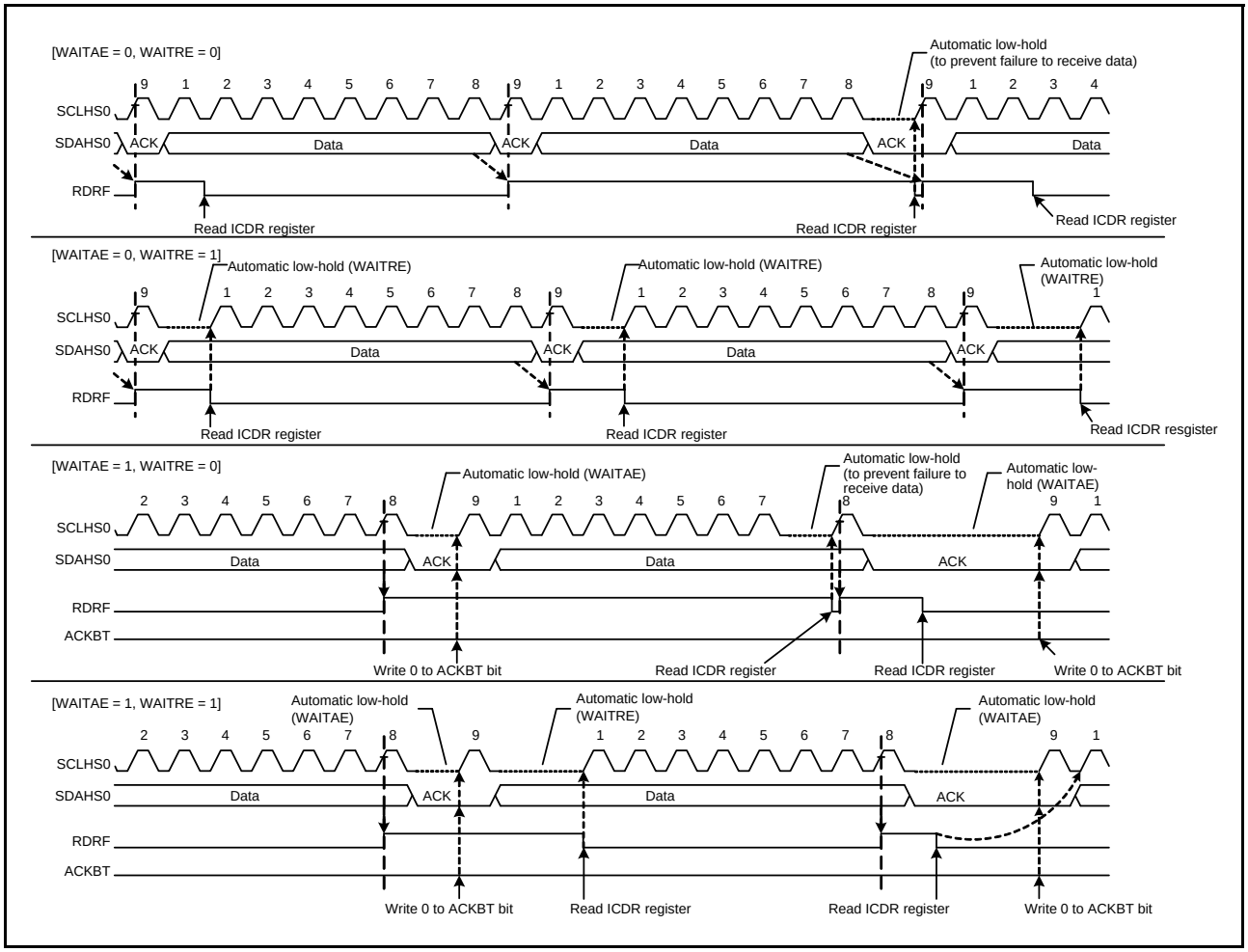


Figure 38.32 Automatic Low-Hold Operation in Receive Mode (Using WAITAE and WAITRE Bits)

38.9 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function specified in the I2C-bus specification, the RIICHS also has functions to prevent double generation of START conditions, to detect arbitration-lost during NACK transmission, and to detect arbitration-lost during slave transmission.

38.9.1 Master Arbitration-Lost Detection Function (MALE Bit)

The RIICHS drives the SDAHS0 line low to generate a START condition. However, if the SDAHS0 line is already driven low by another master device that generates a START condition, the RIICHS generates an arbitration-lost and gives a priority to the transfer by the other master device. Similarly, if the ICCGR.ST bit is set to 1 while the ICBSR.BFREE flag is 0 (bus busy), the RIICHS generates an arbitration-lost and gives a priority to the transfer by the other master device. No START condition is generated in this case.
After a START condition is generated successfully, if the level of the transmit data including the address bits (i.e. the internal SDA output level) and the level on the SDAHS0 line do not match (the internal SDA output is high (the SDAHS0 pin is high-impedance) but low is detected on the SDAHS0 line), the RIICHS generates an arbitration-lost.
If a master arbitration-lost is generated, the RIICHS immediately transitions to slave receive mode. If a slave address including the general call address matches its own address at this time, the RIICHS continues the slave operation.
The RIICHS detects master arbitration-lost when one of the following conditions is met while the ICSER.ALE and ICFER.MALE bits are 1 (master arbitration-lost detection enabled).

Conditions for detecting master arbitration-lost

- When the internal SDA output level and the level on the SDAHS0 line do not match (START condition generation error) when a START condition is generated by setting the ICCGR.ST bit to 1 while the ICBSR.BFREE flag is 1.
- When the ICCGR.ST bit is set to 1 while the BFREE flag is 0 (START condition double generation error).
- When the transmit data excluding acknowledgment bit (internal SDA signal output level) and the level on the SDAHS0 line do not match in master transmit mode (the ICMMR.MST and ICMMR.TRS flags are 1).

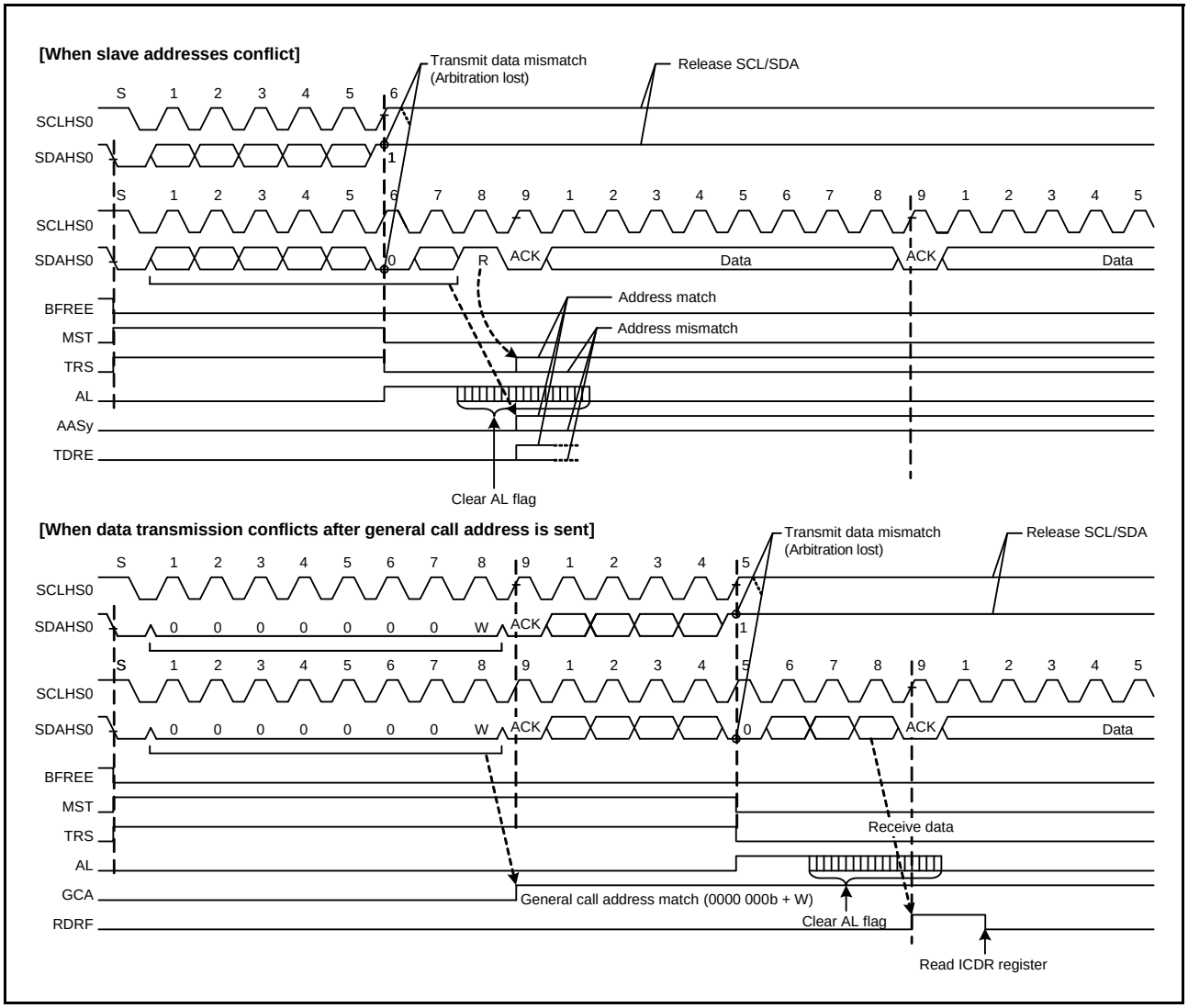


Figure 38.33 Examples of Master Arbitration-Lost Detection (MALE = 1)

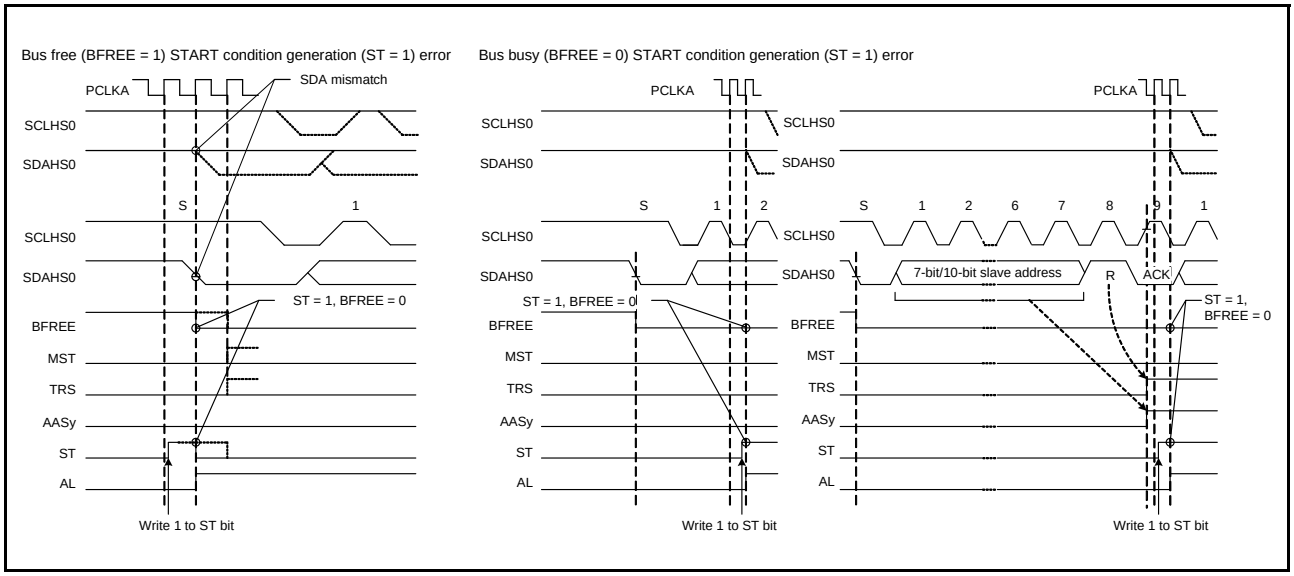


Figure 38.34 Arbitration-Lost Detection When a START Condition is Generated (MALE = 1)

38.9.2 NACK Transmission Arbitration-Lost Detection Function (NALE Bit)

The RIICHS has a function to generate an arbitration-lost if the internal SDA output level and the level on the SDAHS0 line do not match (the internal SDA output is high (the SDAHS0 pin is high-impedance) but low is detected on the SDAHS0 line) during NACK transmission in receive mode. NACK transmission arbitration-lost is generated by 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 conflicts can occur when multiple master devices exchange common information through a single slave device.

Figure 38.35 shows an example of NACK transmission arbitration-lost detection when two master devices (master A and master B) and one slave device are connected to 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 at the same time, arbitration-lost does not occur on either master A or master B when accessing the slave device because the slave addresses are the same. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. Here, master A sends a NACK when it has received second byte, the last byte of data, from the slave device. On the other hand, master B sends an ACK because it has not yet received four bytes of data from the slave device. At this time, a conflict occurs between the NACK of master A and the ACK of master B.

If master A cannot detect the ACK sent by master B when such a conflict occurs, master A generates a STOP condition. This conflicts with the SCL output of master B and interferes with communication.

The RIICHS can generate an arbitration-lost if it receives an ACK while sending an NACK. This prevents the generation of a STOP condition and prevents bus communication interference. Also, in the ARP command processing of SMBus, it is possible to omit surplus processing (FFh transmission processing) after the NACK transmission when the UDID (unique device identification) of the “Assign Address” does not match and after the NACK transmission of the “Get UDID (General)” after the “Assign Address” is confirmed.

The RIICHS detects NACK transmission arbitration-lost when the following condition is met while the ICSER.ALE and ICFER.NALE bits are 1 (NACK transmission arbitration-lost detection is enabled).

If the NACK transmission arbitration-lost occurs, the RIICHS cancels the slave address matched state and enters slave receive mode.

Condition for detecting NACK transmission arbitration-lost

- When the internal SDA output level does not match the SDAHS0 line (ACK is received) during transmission of NACK (the ICACKR.ACKBT bit is 1)

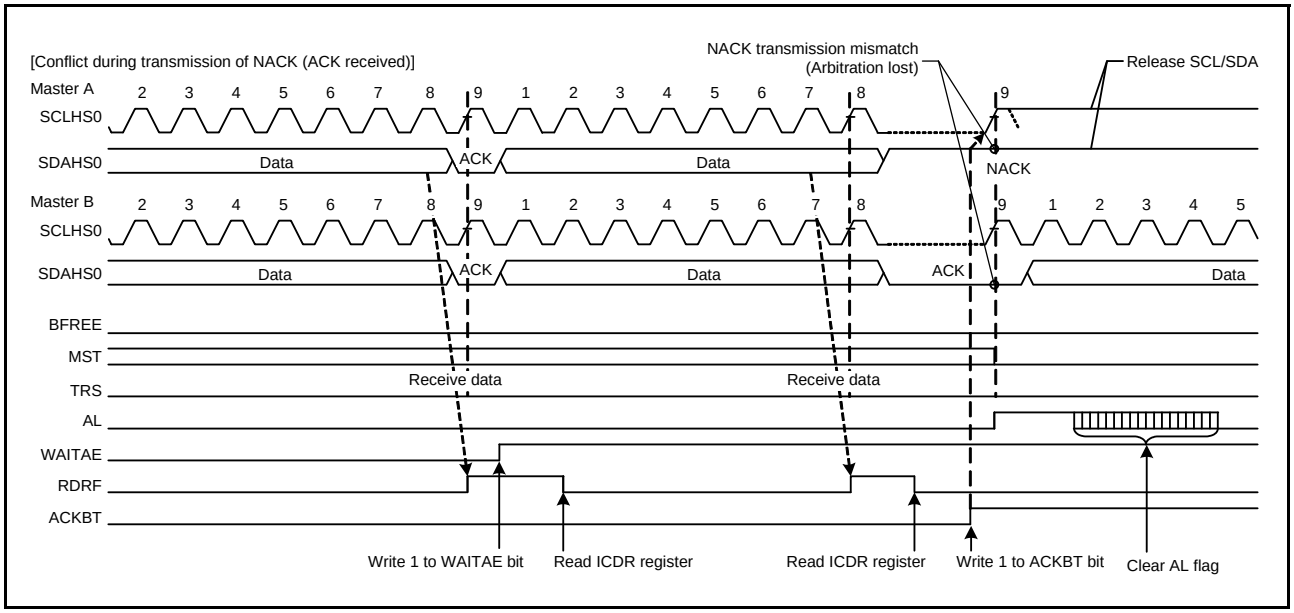


Figure 38.35 Example of NACK Transmission Arbitration-Lost Detection (NALE = 1)

38.9.3 Slave Arbitration-Lost Detection Function (SALE Bit)

The RIICHS has a function to generate an arbitration-lost if the transmit data (i.e. the internal SDA output level) and the level on the SDAHS0 line do not match (the internal SDA output is high (the SDAHS0 pin is high-impedance) but low is detected on the SDAHS0 line) in slave transmit mode.
The slave arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identification) of SMBus. This function can detect a data conflict during UDID transmission of SMBus and omit surplus processing (FFh transmission processing) after the data conflict.
The RIICHS detects slave arbitration-lost when the following condition is met while the ICSER.ALE and ICFER.SALE bits are 1 (slave arbitration-lost detection enabled).
If the slave arbitration-lost occurs, the RIICHS cancels the slave address matched state and enters slave receive mode.

Condition for detecting slave arbitration-lost

- When the transmit data excluding acknowledgment bit (internal SDA output level) does not match the level on the SDAHS0 line in slave transmit mode (the ICMMR.MST flag is 0 and the ICMMR.TRS flag is 1)

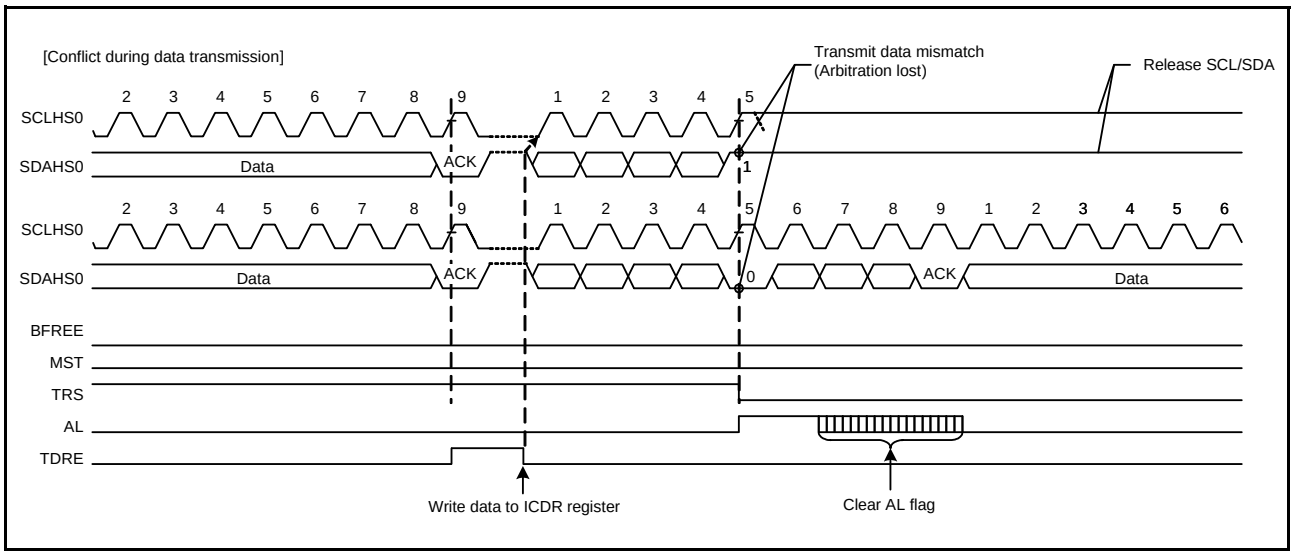


Figure 38.36 Example of Slave Arbitration-Lost Detection (SALE = 1)

38.10 START Condition/Repeated START Condition/STOP Condition Generating Function

38.10.1 Generating a START Condition

The RIICHS generates a START condition when the ICCGR.ST bit is set to 1.
Set the ST bit to 1 (requests to generate a START condition) when the ICBSR.BFREE flag is set to 1 (bus free state).
The RIICHS generates a START condition.
When a START condition is generated normally, the RIICHS automatically shifts to the master transmit mode.
A START condition is generated in the following sequence.

START condition generation

(1) Drive the SDAHS0 line low (high to low).
(2) Ensure the time set in ICFBR.HIGH[7:0] bits and the START condition hold time.
(3) Drive the SCLHS0 line low (high to low).
(4) Detect a low of the SCLHS0 line and ensure the low period of SCLHS0 line set in ICFBR.LOW[7:0] bits.

38.10.2 Generating a Repeated START Condition

The RIICHS generates a repeated START condition when the ICCGR.RS bit is set to 1.
Set the RS bit to 1 (requests to generate a repeated START condition) when the ICBSR.BFREE flag is 0 (bus busy state) and the ICMMR.MST flag is 1 (master mode).
The RIICHS generates a repeated START condition.
A repeated START condition is generated in the following sequence.

Repeated START condition generation

(1) Release the SDAHS0 line.
(2) Ensure the low period of SCLHS0 line set in ICFBR.LOW[7:0] bits.
(3) Release the SCLHS0 line (low to high).
(4) Detect a high of the SCLHS0 line and ensure the time set in ICFBR.LOW[7:0] bits and the repeated START condition setup time.
(5) Drive the SDAHS0 line low (high to low).
(6) Ensure the time set in ICFBR.HIGH[7:0] bits and the repeated START condition hold time.
(7) Drive the SCLHS0 line low (high to low).
(8) Detect a low of the SCLHS0 line and ensure the low period of SCLHS0 line set in ICFBR.LOW[7:0] bits.

Note: When generating repeated START conditions request, please write the slave address to ICDR register after confirming ICCGR.RS bit = 0. Data written in the period of ICCGR.RS bit = 1 is not transferred because it is judged to be the previous retransmission condition.

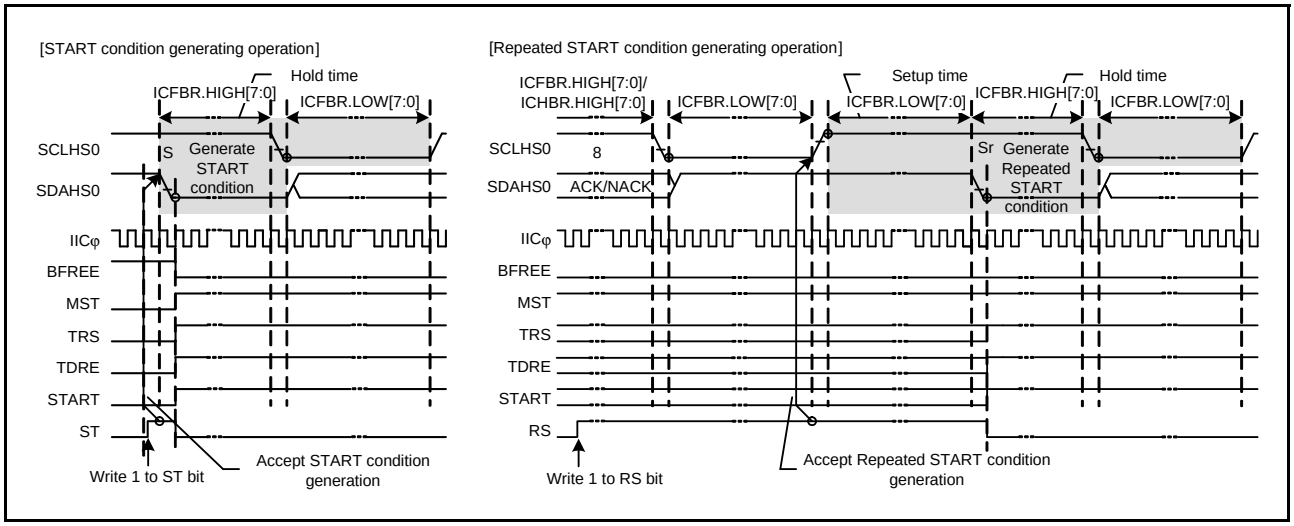


Figure 38.37 START Condition/Repeated START Condition Generating Timing (ST and RS Bits)

Repeated START condition generation after the master transmission

(1) Initial setting. For details, refer to section 38.3.2, Initial Settings.

(2) Read the ICBSR.BFREE flag to check that the bus is open, and then set the ICCGR.ST bit to 1 (requests to generate a START condition). Upon receiving the request, the RIICHS generates a START condition. At the same time, the BFREE flag is automatically set to 0 and the ICSR2.START flag is automatically set to 1 and the ST bit is automatically set to 0. At this time, if the START condition is detected and the internal levels for the SDA output state and the levels on the SDAHS0 line have matched while the ST bit is 1, the RIICHS recognizes that generating of the START condition as requested by the ST bit has been successfully completed, and ICMMR.MST and ICMMR.TRS flags is automatically set to 1, placing the RIICHS in master transmit mode. The ICCSR.TDRE flag is also automatically set to 1 in response to setting of the TRS flag to 1.

(3) Check that the ICCSR.TDRE flag is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDR register. Once the data for transmission are written to ICDR register, the TDRE flag is automatically set to 0, the data are transferred from ICDR register, and the TDRE flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRS flag is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the RIICHS 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 ICCGR.SP bit to generate a STOP condition. For data transmission with an address in the 10-bit format, start by writing 1111 0b, the 2 higher-order bits of the slave address, and W to ICDR register as the first address transmission. Then, as the second address transmission, write the 8 lower-order bits of the slave address to ICDR register.

(4) After confirming that the ICCSR.TDRE flag is 1, write the data for transmission to the ICDR register. The RIICHS automatically holds the SCLHS0 line low until the data for transmission are ready, a repeated START condition is generated, or a STOP condition is generated.

(5) After all bytes of data for transmission have been written to the ICDR register, wait until the value of the ICSR2.TEND flag returns to 1, and then, after check that the ICSR2.START flag is 1, set the ICSR2.START flag to 0.

(6) Set the ICCGR.RS bit to 1 (requests to generate a repeated START condition). Upon receiving the request, the RIICHS generates a repeated START condition.

(7) After check that the ICSR2.START flag is 1, write the value for transmission (the slave address and the R/W# bit) to ICDR register.

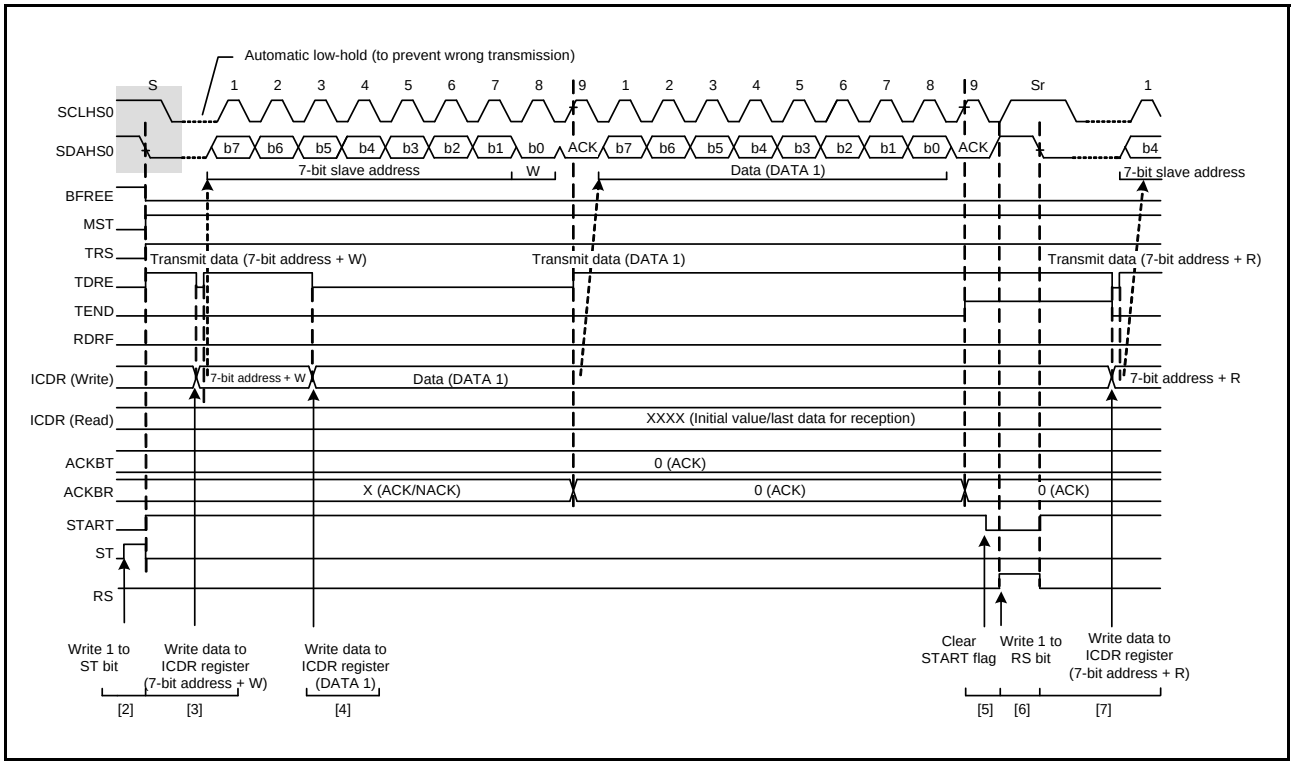


Figure 38.38 Repeated START Condition Generation after the Master Transmission Timing

38.10.3 Generating a STOP Condition

The RIICHS generates a STOP condition when the ICCGR.SP bit is set to 1.
Set the SP bit to 1 (requests to generate a STOP condition) when the ICBSR.BFREE flag is 0 (bus busy state) and the ICMMR.MST flag is 1 (master mode).
The RIICHS generates a STOP condition.
A STOP condition is generated in the following sequence.

STOP condition generation

(1) Drive the SDAHS0 line low (high to low).
(2) Ensure the low period of SCLHS0 line set in ICFBR.LOW[7:0] bits.
(3) Release the SCLHS0 line (low to high).
(4) Detect a high of the SCLHS0 line and ensure the time set in ICFBR.HIGH[7:0] bits and the STOP condition setup time.
(5) Release the SDAHS0 line (low to high).
(6) Ensure the time set in ICFBR.LOW[7:0] bits and the bus free time.
(7) Set the BFREE flag to 1 (to release the bus mastership).

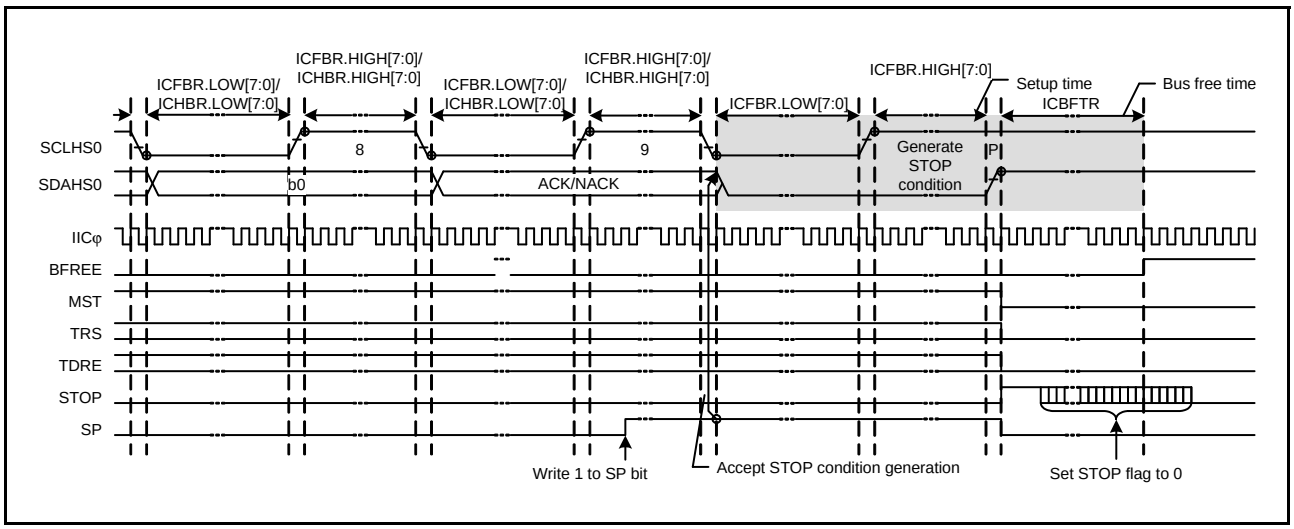


Figure 38.39 STOP Condition Generating Timing (SP Bit)

38.11 Bus Hanging

38.11.1 Timeout Function

The RIICHS includes a timeout function for detecting when the SCLHS0 line has been stuck longer than the predetermined time. The RIICHS can detect an abnormal bus state by monitoring that the SCLHS0 line is stuck low or high for a predetermined time.

The timeout function monitors the SCLHS0 line state and counts the low period or high period using the internal counter. The timeout function resets the internal counter each time the SCLHS0 line changes (rising or falling), but continues to count unless the SCLHS0 line changes. If the internal counter overflows due to no SCLHS0 line change, the RIICHS can detect the timeout and report the bus hung state.

This timeout function is enabled when the ICSER.TMOE bit is 1. It detects a hung state that the SCLHS0 line is stuck low or high during the following conditions: (when ICTOR.TMOM bit = 00b)

- The bus is busy (ICBSR.BFREE flag is 0) in master mode (ICMMR.MST flag is 1).
- The RIICHS's own slave address is detected (ICSSR register is not 0000 0000h) and the bus is busy (ICBSR.BFREE flag is 0) in slave mode (ICMMR.MST flag is 0).
- The bus is free (ICBSR.BFREE flag is 1) while generation of a START condition is requested (ICCGR.ST bit is 1).

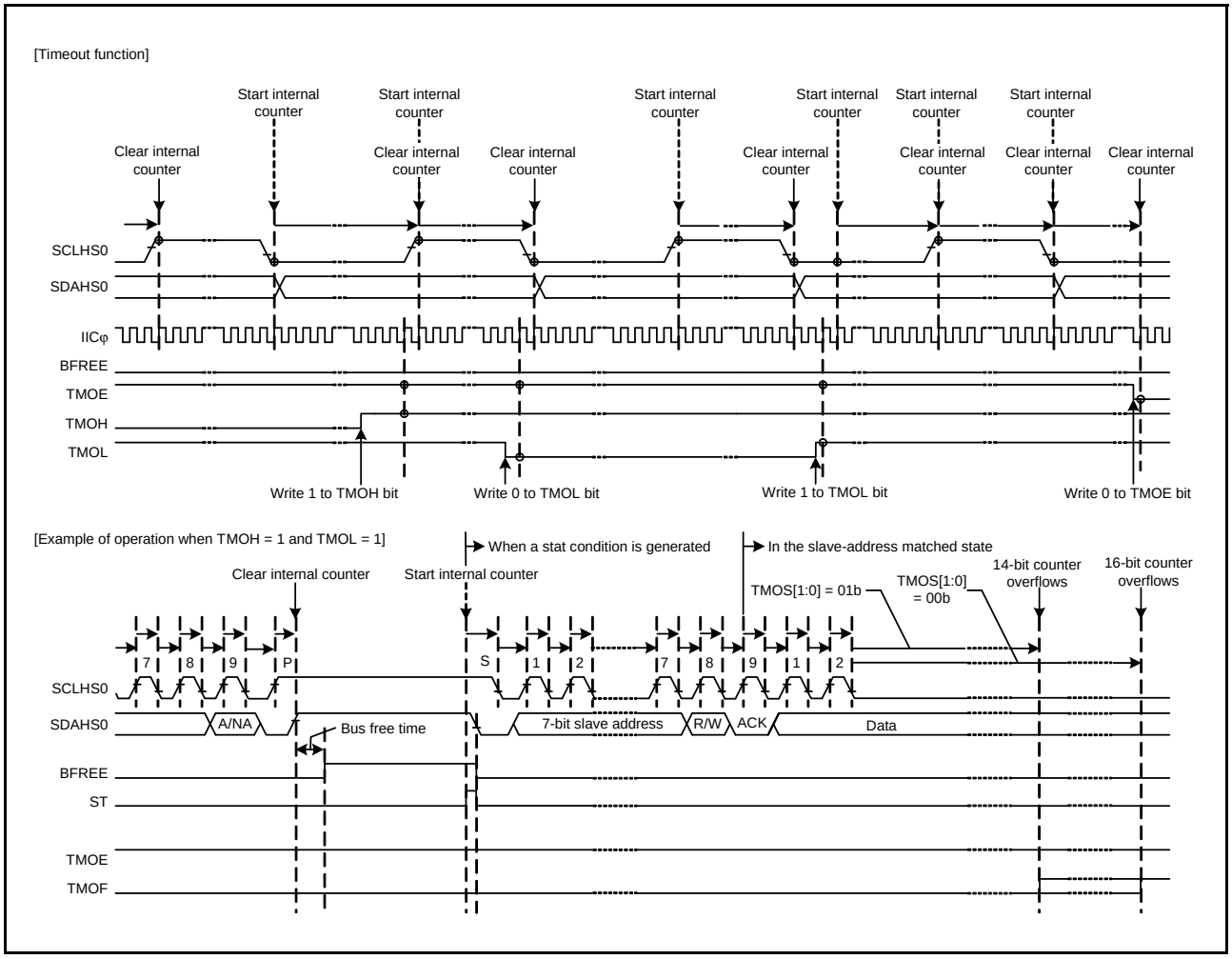


Figure 38.40 Timeout Function (TMOE, TMOS[1:0], TMOH, and TMOL Bits)

38.11.2 Additional SCL Output Function

In master mode, the RIICHS module has a function for the output of additional SCL to release the SDAHS0 line from being held low by the slave device due to the master being out of synchronization with the slave device.
This function is mainly used in master mode to release the SDAHS0 line from the state of being stuck low by including additional SCL output from the RIICHS with single cycles of the SCL as the unit in the case of a bus error where the RIICHS cannot generate a STOP condition because the slave device is holding the SDAHS0 line low. Do not use this function in normal situations. Using it when communications are proceeding correctly will lead to malfunctions.
When the ICOCR.CLO bit is set to 1, an additional clock pulse at the frequency set by the ICRCCR.CKS[2:0] bits and the HIGH[7:0] and LOW[7:0] bits in the ICFBR register (ICHBR register in Hs-mode) is output from the SCLHS0 pin. After output of this clock pulse, the CLO bit automatically becomes 0. The SCLHS0 pin is held low when the ICBSR.BFREE flag is 0 and held high when the BFREE flag is 1. Consecutive additional clock pulses can be output by the software program writing 1 to the CLO bit after having read CLO = 0.
When the RIICHS module is in master mode and the slave device is holding the SDAHS0 line low 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 additional SCL output function can be used to output additional clock pulses one by one to make the slave device release the SDAHS0 line from being held low, thus recovering the bus from an unusable state. Release of the SDAHS0 line by the slave device can be monitored by reading the ICIMR.SDAI flag After confirming release of the SDAHS0 line by the slave device, complete communications by regenerating the STOP condition.
Use this function with the ICFER.MALE bit set to 0 (master arbitration-lost detection disabled). If using this function with the ICFER.MALE bit set to 1, arbitration may be lost due to a mismatch between the value of the ICOCR.SDAO bit and the level of the SDAHS0 line.

Conditions for using the ICOCR.CLO bit

- When the bus is free (ICBSR.BFREE flag = 1) or in master mode (ICMMR.MST flag = 1 and ICBSR.BFREE flag = 0)
- When the communication device does not hold the SCLHS0 line low

Figure 38.41 shows the operation timing of the additional SCL output function (CLO bit).

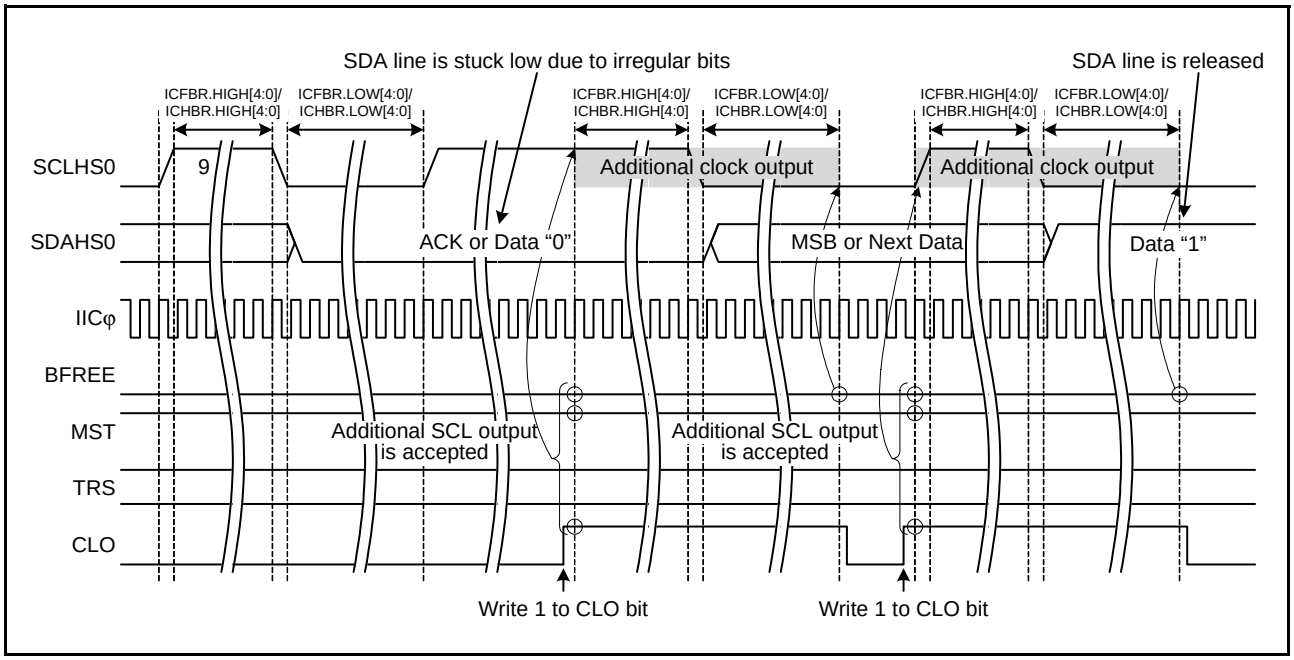


Figure 38.41 Additional SCL Output Function (CLO Bit)

38.12 SMBus Operation

The RIICHS is available for data communication conforming to the SMBus (Version 2.0). To perform SMBus communication, set the ICFER.SMBS bit to 1. To use the transfer rate within a range of 10 kbps to 100 kbps of the SMBus standard, set the ICRCCR.CKS[2:0] bits, the ICFBR.HIGH[7:0] bits, and the ICFBR.LOW[7:0] bits. In addition, determine the values of the ICOCR.DLCS bit and the ICOCR.SDDL[2:0] bits to meet the data hold time (300 ns (min.)). When the RIICHS is used only as a slave device, the transfer rate setting is not necessary, whereas the ICFBR.LOW[7:0] bits need 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 0 to 2 (SAR0, SAR1, and SAR2), and set the corresponding FS bit (7-bit/10-bit address format select) in SARy (y = 0 to 2) to 0 (7-bit address format).
When transmitting the UDID (Unique Device Identifier), set the ICFER.SALE bit to 1 to enable the slave arbitration lost detection function.

38.12.1 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 RIICHS. The measured timeout period must be within cumulative clock low extend time (slave device) ($T_{LOW:SEXT}$: 25 ms (max.)) of the SMBus standard.
If the time measured with the MTU or TMR exceeds the detect clock low timeout ($T_{TIMEOUT}$: 25 ms (min.)) of the SMBus standard, the slave device must release the bus by writing 1 to the ICRCR.ISRST bit to perform an internal reset of the RIICHS. When an internal reset is performed, the RIICHS stops driving the bus for the SCLHS0 pin and SDAHS0 pin and make the SCLHS0/SDAHS0 pin outputs high-impedance, which releases the bus.

(2) Measuring timeout of master device

The following periods (timeout interval: $T_{LOW:MEXT}$) must be measured for master devices in SMBus communication.

- From START condition to acknowledgment bit
- Between acknowledgment bits
- From acknowledgment 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 transmission end interrupt (TEI) or receive data full interrupt (RXI) of the RIICHS. The measured timeout period must be within the cumulative clock low extend time (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 SCL), monitor the ICSR2.TEND flag in master transmit mode (master transmitter) and the ICCSR.RDRF flag in master receive mode (master receiver). For this reason, perform bytewise transmit operation in master transmit mode, and hold the ICCSCR.WAITAE bit 0 until the byte just before reception of the final byte in master receive mode. While the WAITAE bit is 0, the RDRF flag is set to 1 at the rising edge of the ninth SCL.
If the period measured with the MTU or TMR exceeds the cumulative clock low extend time (master device) ($T_{LOW:MEXT}$: 10 ms (max.)) of the SMBus standard or the total of measured periods exceeds the detect clock low timeout ($T_{TIMEOUT}$: 25 ms (min.)) of the SMBus standard, the master device must stop the transaction by generating a STOP condition. In master transmit mode, immediately stop the transmit operation (writing data to ICDR register).

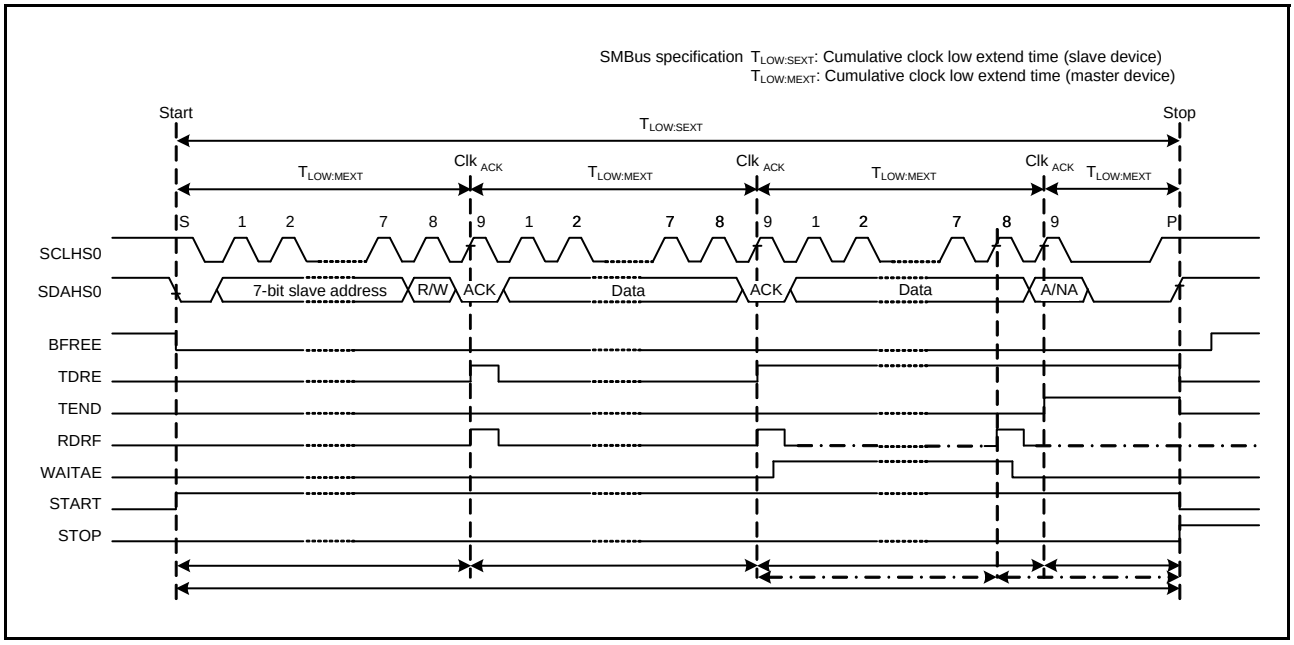


Figure 38.42 SMBus Timeout Measurement

38.12.2 Packet Error Code (PEC)

This MCU incorporates a CRC calculator. The CRC calculator enables transmission of a packet error code (PEC) or checking the received data of the SMBus in data communication of the RIICHS. For the CRC generating polynomials of the CRC calculator, refer to section 43, CRC Calculator (CRCA).

The PEC data in master transmit mode can be generated by writing all transmit data to the CRC data input register (CRCDIR) in the CRC calculator.

The PEC data in master receive mode can be checked by writing all receive data to CRCDIR in the CRC calculator and comparing the obtained value in the CRC data output register (CRCDOR) with the received PEC data.

To send ACK or NACK according to the match or mismatch result when the final byte is received as a result of the PEC code check, set the ICCSCR.WAITAE bit to 1 before the rising edge of the eighth SCL during reception of the final byte, and hold the SCLHS0 line low at the falling edge of the eighth clock pulse.

38.12.3 SMBus Host Notification Protocol (Notify ARP Master Command)

In communications over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of (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 this MCU to operate as an SMBus host (or ARP master), the host address (0001 000b) sent from the slave device must be detected as a slave address, so the RIICHS has a function for detecting the host address. To detect the host address as a slave address, set the ICFER.SMBS bit and the ICSCR.HOAE bit to 1. Operation after the host address has been detected is the same as normal slave operation.

38.13 Interrupt Sources

The RIICHS generates four types of interrupt requests: transmit data empty, receive data full, transmission end, and communication error or communication event (START condition detection, STOP condition detection, NACK detection, arbitration-lost detection, and timeout detection).

Table 38.5 lists details of the interrupt requests. The transmit data empty and receive data full interrupt requests allow the DTC or DMAC to start data transfer.

Table 38.5 Interrupt Sources

Symbol	Interrupt Source	Interrupt Flag	Interrupt Enable Bit	Start DTC/DMA Transfer	Interrupt Generation Condition
TXI*1	Transmit data empty	ICCSR.TDRE	ICCSIER.TIE	Possible	TDRE = 1 and TIE = 1
RXI*2	Receive data full	ICCSR.RDRF	ICCSIER.RIE	Possible	RDRF = 1 and RIE = 1
TEI*3	Transmission end	ICSR2.TEND	ICSIER.TEIE	Not possible	TEND = 1 and TEIE = 1
EEI	Communication error/ communication event	ICSR2.START	ICSIER.STIE	Not possible	START = 1 and STIE = 1
		ICSR2.STOP	ICSIER.SPIE		STOP = 1 and SPIE = 1
		ICSR2.NACKF	ICSIER.NAKIE		NACKF = 1 and NAKIE = 1
		ICSR2.AL	ICSIER.ALIE		AL = 1 and ALIE = 1
		ICSR2.TMOF	ICSIER.TMOIE		TMOF = 1 and TMOIE = 1

Note: There is a 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 an interrupt request has been masked, read the relevant flag again to check whether clearing or masking has been completed, and then return from interrupt handling. Returning from interrupt handling without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.

Note 1. Because TXI is an edge-detected interrupted, it does not require clearing. Furthermore, the ICSR2.TDRE flag (a condition for TXI) is automatically set to 0 when transmit data is written to the ICDRT register or when a STOP condition is detected (ICSR2.STOP flag is 1).

Note 2. Because RXI is an edge-detected interrupted, it does not require clearing. Furthermore, the ICSR2.RDRF flag (a condition for RXI) is automatically set to 0 when data is read from the ICDRR register.

Note 3. When using the TEI interrupt, clear the ICSR2.TEND flag in the TEI interrupt handling.
Note that the ICSR2.TEND flag is automatically set to 0 when transmit data is written to the ICDRT register or when a STOP condition is detected (ICSR2.STOP flag is 1).

Clear each flag or mask the interrupt request during interrupt handling.

38.13.1 Buffer Operation for TXI and RXI Interrupts

If the conditions for generating a TXI or RXI interrupt is satisfied while the corresponding ICU.IRn.IR flag is 1, the interrupt request is not output to the ICU but retained internally (up to one request can be retained internally for each source).

When the IR flag becomes 0, the retained interrupt request is output to the ICU. The retained interrupt request is automatically cleared under normal conditions of usage.

The retained interrupt request can also be cleared by writing 0 to the corresponding interrupt enable bit in the ICCSIER register.

38.14 Event Link Function

The RIICHS outputs the corresponding event signal to the event link controller (ELC) when any of the following events occur.

- Transmit data empty
- Receive data full
- Transmission end
- Communication error/communication event

38.14.1 Interrupt Handling and Event Linking

The RIICHS has four types of interrupts: transmit data empty, receive data full, transmission end, and communication error/communication event (START condition detection, STOP condition detection, NACK detection, arbitration-lost detection, and timeout detection). Each of these interrupt sources has an enable bit that controls whether the interrupt is enabled or disabled. When an interrupt request source occurs, an interrupt request signal is output to the ICU if the corresponding interrupt enable bit is 1.
On the other hand, the event signal is transmitted via the ELC to other modules when the corresponding interrupt request source occurs, regardless of the interrupt enable bit.
For details on interrupt sources, see Table 38.5.

38.15 Reset Description

Table 38.6 Reset control of Registers (1/3)

Register	Bit	System Reset	ICRCR Register	
			MRST Bit	ISRST Bit
ICCR	ICE	To be reset	To be reset	Retained
ICRCR	ISRST	To be reset	To be reset	Retained
	MRST	To be reset	Retained	Retained
ICMMR	MSWP	To be reset	To be reset	To be reset
	TRS	To be reset	To be reset	To be reset
	MST	To be reset	To be reset	To be reset
ICFER	HSME	To be reset	To be reset	Retained
	FMPE	To be reset	To be reset	Retained
	SMBS	To be reset	To be reset	Retained
	SCLE	To be reset	To be reset	Retained
	SALE	To be reset	To be reset	Retained
	NALE	To be reset	To be reset	Retained
	MALE	To be reset	To be reset	Retained
ICSCR	SAR2E	To be reset	To be reset	Retained
	SAR1E	To be reset	To be reset	Retained
	SAR0E	To be reset	To be reset	Retained
	HOAE	To be reset	To be reset	Retained
	DIDE	To be reset	To be reset	Retained
	HSMCE	To be reset	To be reset	Retained
	GCAE	To be reset	To be reset	Retained
ICRCCR	CKS[2:0]	To be reset	To be reset	Retained
ICFBR	HIGH[7:0]	To be reset	To be reset	Retained
	LOW[7:0]	To be reset	To be reset	Retained
ICHBR	HIGH[7:0]	To be reset	To be reset	Retained
	LOW[7:0]	To be reset	To be reset	Retained
ICBFTR		To be reset	To be reset	Retained
ICOCR	DLCS	To be reset	To be reset	Retained
	SDDL[2:0]	To be reset	To be reset	Retained
	CLO	To be reset	To be reset	Retained
	SOWP	To be reset	To be reset	To be reset
	SCLO	To be reset	To be reset	Retained
	SDAO	To be reset	To be reset	Retained
ICICR	SDIDL[1:0]	To be reset	To be reset	Retained
	NFE	To be reset	To be reset	Retained
	NF[1:0]	To be reset	To be reset	Retained
ICTOR	TMOM[1:0]	To be reset	To be reset	Retained
	TMOH	To be reset	To be reset	Retained
	TMOL	To be reset	To be reset	Retained
	TMOS[1:0]	To be reset	To be reset	Retained
ICACKR	ACKWP	To be reset	To be reset	To be reset
	ACKBT	To be reset	To be reset	To be reset
	ACKBR	To be reset	To be reset	To be reset

Table 38.6 Reset control of Registers (2/3)

Register	Bit	System Reset	ICRCR Register	
			MRST Bit	ISRST Bit
ICCSCR	WAITRE	To be reset	To be reset	Retained
	WAITAE	To be reset	To be reset	Retained
ICCGR	SP	To be reset	To be reset	To be reset
	RS	To be reset	To be reset	To be reset
	ST	To be reset	To be reset	To be reset
ICDR		To be reset	To be reset	To be reset
ICSR2	TMOF	To be reset	To be reset	To be reset
	AL	To be reset	To be reset	To be reset
	TEND	To be reset	To be reset	To be reset
	NACKF	To be reset	To be reset	To be reset
	STOP	To be reset	To be reset	To be reset
	START	To be reset	To be reset	To be reset
ICSER	TMOE	To be reset	To be reset	Retained
	ALE	To be reset	To be reset	Retained
	TEDE	To be reset	To be reset	Retained
	NAKDE	To be reset	To be reset	Retained
	SPDE	To be reset	To be reset	Retained
	STDE	To be reset	To be reset	Retained
ICSIER	TMOIE	To be reset	To be reset	Retained
	ALIE	To be reset	To be reset	Retained
	TEIE	To be reset	To be reset	Retained
	NAKIE	To be reset	To be reset	Retained
	SPIE	To be reset	To be reset	Retained
	STIE	To be reset	To be reset	Retained
ICCSR	RDRF	To be reset	To be reset	To be reset
	TDRE	To be reset	To be reset	To be reset
ICCSER	RDE	To be reset	To be reset	Retained
	TDE	To be reset	To be reset	Retained
ICCSIER	RIE	To be reset	To be reset	Retained
	TIE	To be reset	To be reset	Retained
ICBSR	BFREE	To be reset	To be reset	Retained
ICSSR	AAS2	To be reset	To be reset	To be reset
	AAS1	To be reset	To be reset	To be reset
	AAS0	To be reset	To be reset	To be reset
	HOA	To be reset	To be reset	To be reset
	DID	To be reset	To be reset	To be reset
	HSMC	To be reset	To be reset	To be reset
	GCA	To be reset	To be reset	To be reset
SAR0	FS	To be reset	To be reset	Retained
	SVA[9:0]	To be reset	To be reset	Retained
SAR1	FS	To be reset	To be reset	Retained
	SVA[9:0]	To be reset	To be reset	Retained
SAR2	FS	To be reset	To be reset	Retained
	SVA[9:0]	To be reset	To be reset	Retained

Table 38.6 Reset control of Registers (3/3)

Register	Bit	System Reset	ICRCR Register	
			MRST Bit	ISRST Bit
SAMR0	SAV	To be reset	To be reset	Retained
	FS	To be reset	To be reset	Retained
	SVA[9:0]	To be reset	To be reset	Retained
SAMR1	SAV	To be reset	To be reset	Retained
	FS	To be reset	To be reset	Retained
	SVA[9:0]	To be reset	To be reset	Retained
SAMR2	SAV	To be reset	To be reset	Retained
	FS	To be reset	To be reset	Retained
	SVA[9:0]	To be reset	To be reset	Retained
ICBCR	BC[4:0]	To be reset	To be reset	To be reset
ICIMR	SDAO	To be reset	To be reset	To be reset
	SCLO	To be reset	To be reset	To be reset
	SDAI	To be reset	To be reset	Retained
	SCLI	To be reset	To be reset	Retained

38.16 Usage Notes

38.16.1 Settings of the Module Stop Function

RIICHS operation can be disabled or enabled by module stop control register D (MSTPCRD).
After a reset, the RIICHS is stopped. The registers become accessible when it is released from the module stop state. For details, refer to section 11., Low Power Consumption.

39. CAN Module (CAN)

39.1 Overview

This MCU implements two channels of the CAN (Controller Area Network) module that complies with the ISO 11898-1 Standards. The CAN module transmits and receives both formats of messages, namely the standard identifier (11 bits) (identifier is hereafter referred to as ID) and extended ID (29 bits).

Table 39.1 lists the specifications of the CAN module, and Figure 39.1 shows a block diagram of the CAN module (i = 0, 1).

Connect the CAN bus transceiver externally.

Table 39.1 Specifications of CAN Module

Item	Description
Protocol	• ISO 11898-1 compliant (standard and extended frames)
Bit rate	• Programmable bit rate up to 1 Mbps (fCAN ≥ 8 MHz) fCAN: CAN clock source
Message box	• 32 mailboxes: Two selectable mailbox modes Normal mailbox mode: 32 mailboxes can be configured for either transmission or reception. FIFO mailbox mode: 24 mailboxes can be configured for either transmission or reception. Of the other mailboxes, four FIFO stages can be configured for transmission and four FIFO stages for reception.
Reception	• Data frame and remote frame can be received. • Selectable receiving ID format (only standard ID, only extended ID, or both IDs) • Programmable one-shot reception function • Selectable from overwrite mode (message overwritten) and overrun mode (message discarded) • The reception complete interrupt can be individually enabled or disabled for each mailbox.
Acceptance filter	• Eight acceptance masks (one mask for every four mailboxes) • The mask can be individually enabled or disabled for each mailbox.
Transmission	• Data frame and remote frame can be transmitted. • Selectable transmitting ID format (only standard ID, only extended ID, or both IDs) • Programmable one-shot transmission function • Selectable from ID priority mode and mailbox number priority mode • Transmission request can be aborted (the completion of abort can be confirmed with a flag) • The transmission complete interrupt can be individually enabled or disabled for each mailbox.
Mode transition for bus-off recovery	• Mode transition for the recovery from the bus-off state can be selected: ISO 11898-1 Standards compliant Automatic entry to CAN halt mode at bus-off entry Automatic entry to CAN halt mode at bus-off end Entry to CAN halt mode by a program Transition into error-active state by a program
Error status monitoring	• CAN bus errors (stuff error, form error, ACK error, CRC error, bit error, and ACK delimiter error) can be monitored. • Transition to error states can be detected (error-warning, error-passive, bus-off entry, and bus-off recovery). • The error counters can be read.
Time stamp function	• Time stamp function using a 16-bit counter • The reference clock can be selected from 1-, 2-, 4- and 8-bit time periods.
Interrupt function	• Five types of interrupt sources (reception complete, transmission complete, receive FIFO, transmit FIFO, and error interrupts)
CAN sleep mode	• Current consumption can be reduced by stopping the CAN clock.
Software support unit	• Three software support units: Acceptance filter support Mailbox search support (receive mailbox search, transmit mailbox search, and message lost search) Channel search support
CAN clock source	Peripheral module clock (PCLKB) or CANMCLK
Test mode	• Three test modes available for user evaluation Listen-only mode Self-test mode 0 (external loopback) Self-test mode 1 (internal loopback)
Power consumption reducing function	Module-stop state can be set.

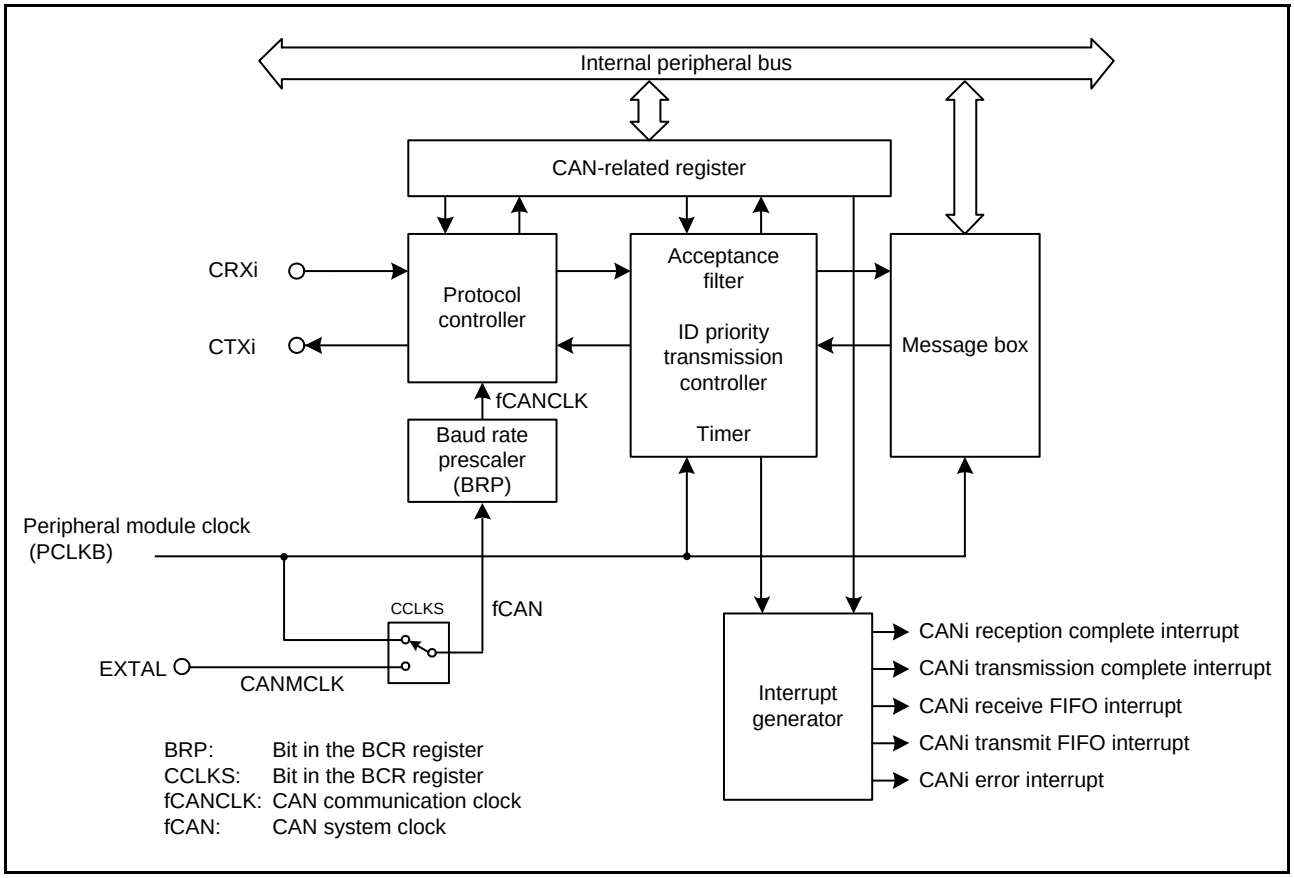


Figure 39.1 Block Diagram of CAN Module (i = 0, 1)

- CRXi and CTXi (i = 0, 1)
  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 39.2 lists the CAN module pins.
The CAN functions should be selected for the pins multiplexed with other signals. For details, see section 22, I/O Ports.

Table 39.2 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

39.2 Register Descriptions

39.2.1 Control Register (CTLR)

Address(es): CAN0.CTLR 0009 0840h, CAN1.CTLR 0009 1840h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RBOC	BOM[1:0]		SLPM	CANM[1:0]		TSPS[1:0]		TSRC	TPM	MLM	IDFM[1:0]		MBM
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MBM	CAN Mailbox Mode Select *1	0: Normal mailbox mode 1: FIFO mailbox mode	R/W
b2, b1	IDFM[1:0]	ID Format Mode Select*1	b2 b1 0 0: Standard ID mode All mailboxes (including FIFO mailboxes) handle only standard Ids. 0 1: Extended ID mode All mailboxes (including FIFO mailboxes) handle only extended IDs. 1 0: Mixed ID mode All mailboxes (including FIFO mailboxes) handle both standard IDs and extended IDs. Standard IDs or extended IDs are specified by using the IDE bit in the corresponding mailbox in normal mailbox mode. In FIFO mailbox mode, the IDE bit in the corresponding mailbox is used for mailboxes [0] to [23], the IDE bits in FIDCR0 and FIDCR1 are used for the receive FIFO, and the IDE bit in mailbox [24] is used for the transmit FIFO. 1 1: Do not use this combination	R/W
b3	MLM	Message Lost Mode Select*2	0: Overwrite mode 1: Overrun mode	R/W
b4	TPM	Transmission Priority Mode Select*2	0: ID priority transmit mode 1: Mailbox number priority transmit mode	R/W
b5	TSRC	Time Stamp Counter Reset Command*4	0: Nothing occurred 1: Reset*3	R/W
b7, b6	TSPS[1:0]	Time Stamp Prescaler Select*1	b7 b6 0 0: Every bit time 0 1: Every 2-bit time 1 0: Every 4-bit time 1 1: Every 8-bit time	R/W
b9, b8	CANM[1:0]	CAN Operating Mode Select*5	b9 b8 0 0: CAN operation mode 0 1: CAN reset mode 1 0: CAN halt mode 1 1: CAN reset mode (forcible transition)	R/W
b10	SLPM	CAN Sleep Mode*5, *6	0: Other than CAN sleep mode 1: CAN sleep mode	R/W
b12, b11	BOM[1:0]	Bus-Off Recovery Mode *1	b12 b11 0 0: Normal mode (ISO 11898-1 compliant) 0 1: Entry to CAN halt mode automatically at bus-off entry 1 0: Entry to CAN halt mode automatically at bus-off end 1 1: Entry to CAN halt mode (during bus-off recovery period) by a program request	R/W
b13	RBOC	Forcible Return From Bus-Off*2	0: Nothing occurred 1: Forcible return from bus-off*3	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Write to the BOM[1:0], TSPS[1:0], TPM, MLM, IDFM[1:0], and MBM bits in CAN reset mode.
Note 2. Set the RBOC bit to 1 in the bus-off state.
Note 3. This bit is automatically set back to 0 after being set to 1. It should be read as 0.
Note 4. Set the TSRC bit to 1 in CAN operation mode.
Note 5. When the CANM[1:0] and SLPM bits are changed, check STR to ensure that the mode has been switched. Do not change the CANM[1:0] bits or SLPM bit until the mode has been switched.

Note 6. Write to the SLPM bit in CAN reset mode or CAN halt mode. When changing the SLPM bit, write 0 or 1 to only the SLPM bit.

MBM Bit (CAN Mailbox Mode Select)

When the MBM bit is 0 (normal mailbox mode), mailboxes [0] to [31] are configured as transmit or receive mailboxes.
When the MBM bit is 1 (FIFO mailbox mode), mailboxes [0] to [23] are configured as transmit or receive mailboxes. Mailboxes [24] to [27] are configured as a transmit FIFO and mailboxes [28] to [31] as a receive FIFO.
Transmit data is written into mailbox [24] (mailbox [24] is a window mailbox for the transmit FIFO). Receive data is read from mailbox [28] (mailbox [28] is a window mailbox for the receive FIFO).
Table 39.3 lists the mailbox configuration.

IDFM[1:0] Bits (ID Format Mode Select)

The IDFM[1:0] bits specify the ID format.

MLM Bit (Message Lost Mode Select)

The MLM bit specifies the operation when a new message is captured in the unread mailbox. Overwrite mode or overrun mode can be selected. All mailboxes (including the receive FIFO) are set to either overwrite mode or overrun mode.
When the MLM bit is 0, all mailboxes are set to overwrite mode and the new message is overwriting the old message.
When the MLM bit is 1, all mailboxes are set to overrun mode and the new message is discarded.

TPM Bit (Transmission Priority Mode Select)

The TPM bit specifies the priority of modes when transmitting messages.
ID priority transmit mode or mailbox number transmit mode can be selected. All mailboxes are set for either ID priority transmission or mailbox number priority transmission.
When the TPM bit is 0, ID priority transmit mode is selected and transmission priority complies with the CAN bus arbitration rule, as defined in the ISO 11898-1 Standards. In ID priority transmit mode, mailboxes [0] to [31] (in normal mailbox mode), and mailboxes [0] to [23] (in FIFO mailbox mode), and the transmit FIFO are compared for the IDs of mailboxes configured for transmission. If two or more mailbox IDs are the same, the mailbox with the smaller number has higher priority.
Only the next message to be transmitted from the transmit FIFO is included in the transmission arbitration. If a transmit FIFO message is being transmitted, the next pending message within the transmit FIFO is included in the transmission arbitration.
When the TPM bit is 1, mailbox number transmit mode is selected and the transmit mailbox with the smallest mailbox number has the highest priority. In FIFO mailbox mode, the transmit FIFO has lower priority than normal mailboxes (mailboxes [0] to [23]).

TSRC Bit (Time Stamp Counter Reset Command)

The TSRC bit is used to reset the time stamp counter. When the TSRC bit is set to 1, TSR is set to 0000h. This bit is automatically set to 0.

TSPS[1:0] Bits (Time Stamp Prescaler Select)

The TSPS[1:0] bits select the prescaler for the time stamp. The reference clock for the time stamp can be selected to be either 1-, 2-, 4- or 8-bit time periods.

CANM[1:0] Bits (CAN Operating Mode Select)

The CANM[1:0] bits select one of the following modes for the CAN module: CAN operation mode, CAN reset mode, or CAN halt mode. CAN sleep mode is set by the SLPM bit. For details, refer to section 39.3, 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 39.3, Operating Mode.

BOM[1:0] Bits (Bus-Off Recovery Mode)

The BOM[1:0] bits are used to select bus-off recovery mode for the CAN module.

When the BOM[1:0] bits are 00b, the recovery from bus-off is compliant with the ISO 11898-1 Standards, i.e. the CAN module reenters CAN communication (error-active state) after detecting 11 consecutive recessive bits 128 times. A bus-off recovery interrupt request is generated when recovering from bus-off.

When the BOM[1:0] bits are 01b and the CAN module reaches the bus-off state, the CANM[1:0] bits in CTLR are set to 10b (CAN halt mode) to enter CAN halt mode. No bus-off recovery interrupt request is generated when recovering from bus-off, and TECR and RECR are set to 00h.

When the BOM[1:0] bits are 10b, the CANM[1:0] bits are set to 10b as soon as the CAN module reaches the bus-off state. The CAN module enters CAN halt mode after the recovery from the bus-off state, i.e. after detecting 11 consecutive recessive bits 128 times. A bus-off recovery interrupt request is generated when recovering from bus-off, and TECR and RECR are set to 00h.

When the BOM[1:0] bits are 11b, the CAN module enters CAN halt mode by setting the CANM[1:0] bits to 10b while the CAN module is still in the bus-off state. No bus-off recovery interrupt request is generated when recovering from bus-off and TECR and RECR are set to 00h. However, if the CAN module recovers from bus-off after detecting 11 consecutive recessive bits 128 times before the CANM[1:0] bits are set to 10b, a bus-off recovery interrupt request is generated.

If the CPU requests an entry to CAN reset mode at the same time as the CAN module attempts to enter CAN halt mode (at bus-off entry when the BOM[1:0] bits are 01b, or at bus-off end when the BOM[1:0] bits are 10b), then the CPU request to enter CAN reset mode has higher priority.

RBOC Bit (Forcible Return From Bus-Off)

When the RBOC bit is set to 1 (force return from bus-off) in the bus-off state, the CAN module forcibly returns from the bus-off state. This bit is automatically set to 0. The error state changes from bus-off to error-active. When the RBOC bit is set to 1, RECR and TECR are set to 00h and the BOST bit in STR is set to 0 (the CAN module is not in bus-off state). The other registers remain unchanged even when the RBOC bit is set to 1. No bus-off recovery interrupt request is generated by this recovery from the bus-off state. Use the RBOC bit only when the BOM[1:0] bits are 00b (normal mode).

Table 39.3 Mailbox Configuration

Mailbox	MBM Bit = 0 (Normal Mailbox Mode)	MBM Bit = 1 (FIFO Mailbox Mode)
Mailboxes [0] to [23]	Normal mailbox	Normal mailbox
Mailboxes [24] to [27]		Transmit FIFO
Mailboxes [28] to [31]		Receive FIFO

Points 1 to 5 below should be considered when the CTLR.MBM bit is set to 1.

Note 1. Transmit FIFO is controlled by TFCR. MCTLj of mailboxes [24] to [27] is disabled. MCTL24 to MCTL27 cannot be used by the transmit FIFO.

Note 2. Receive FIFO is controlled by RFCR. MCTLj of mailboxes [28] to [31] is disabled. MCTL28 to MCTL31 cannot be used by the receive FIFO.

Note 3. Refer to MIER about the FIFO interrupts.

Note 4. The corresponding bits in MKIVLR for mailboxes [24] to [31] are disabled. Set 0 to these bits.

Note 5. Transmit/receive FIFOs can be used for both data frames and remote frames.

39.2.2 Bit Configuration Register (BCR)

Address(es): CAN0.BCR 0009 0844h, CAN1.BCR 0009 1844h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TSEG1[3:0]				—	—	BRP[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	SJW[1:0]		—	TSEG2[2:0]			—	—	—	—	—	—	—	CCLKS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CCLKS	CAN Clock Source Selection	0: PCLKB (generated by the PLL clock) 1: CANMCLK (generated by the main clock)	R/W
b7 to b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b8	TSEG2[2:0]	Time Segment 2 Control	b10 b8 0 0 0: Setting prohibited 0 0 1: 2 Tq 0 1 0: 3 Tq 0 1 1: 4 Tq 1 0 0: 5 Tq 1 0 1: 6 Tq 1 1 0: 7 Tq 1 1 1: 8 Tq	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b13, b12	SJW[1:0]	Resynchronization Jump Width Control	b13 b12 0 0: 1 Tq 0 1: 2 Tq 1 0: 3 Tq 1 1: 4 Tq	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25 to b16	BRP[9:0]	Prescaler Division Ratio Select*1	These bits set the frequency of the CAN communication clock (fCANCLK).	R/W
b26	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b31 to b28	TSEG1[3:0]	Time Segment 1 Control	b31 b28 0 0 0 0: Setting prohibited 0 0 0 1: Setting prohibited 0 0 1 0: Setting prohibited 0 0 1 1: 4 Tq 0 1 0 0: 5 Tq 0 1 0 1: 6 Tq 0 1 1 0: 7 Tq 0 1 1 1: 8 Tq 1 0 0 0: 9 Tq 1 0 0 1: 10 Tq 1 0 1 0: 11 Tq 1 0 1 1: 12 Tq 1 1 0 0: 13 Tq 1 1 0 1: 14 Tq 1 1 1 0: 15 Tq 1 1 1 1: 16 Tq	R/W

Tq: Time Quantum

Note 1. Do not select the value less than 1 while the SCKCR3.CKSEL[2:0] bits are 010b (selecting the main clock oscillator).

For bit timing setting, refer to section 39.4, CAN Communication Speed Setting.
Set BCR before entering CAN halt mode or CAN operation mode from CAN reset mode. After the setting is made once, this register can be written to in CAN reset mode or CAN halt mode.
BCR consists of 24 bits. A 32-bit read/write access should be performed carefully not to rewrite bits b0 to b7.

CCLKS Bit (CAN Clock Source Selection)

When the CCLKS bit is 0, the peripheral module clock (PCLKB) produced by the PLL frequency synthesizer is used as the CAN clock source (fCAN).
When the CCLKS bit is 1, CANMCLK produced externally by the EXTAL pins is used as the CAN clock source (fCAN).

TSEG2[2:0] Bits (Time Segment 2 Control)

The TSEG2[2:0] bits are used to specify the length of the phase buffer segment 2 (PHASE_SEG2) with a Tq value. A value from 2 to 8 Tq can be set. Set a value smaller than that of the TSEG1[3:0] bits.

SJW[1:0] Bits (Resynchronization Jump Width Control)

The SJW[1:0] bits are used to specify the resynchronization jump width with a Tq value. A value from 1 to 4 Tq can be set. Set a value smaller than or equal to that of the TSEG2[2:0] bits.

BRP[9:0] Bits (Prescaler Division Ratio Select)

The BRP[9:0] bits are used to set the frequency of the CAN communication clock (fCANCLK). The fCANCLK cycle is 1 Tq. If the setting is P (0 to 1023), the baud rate prescaler divides fCAN by P + 1.

TSEG1[3:0] Bits (Time Segment 1 Control)

The TSEG1[3:0] bits are used to specify the total length of the propagation time segment (PROP_SEG) and phase buffer segment 1 (PHASE_SEG1) with a time quantum (Tq) value.
A value from 4 to 16 Tq can be set.

39.2.3 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

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	SID[10:0]											EID[17:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EID[17:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b17 to b0	EID[17:0]	Extended ID	0: Corresponding EID[17:0] bit is not compared 1: Corresponding EID[17:0] bit is compared	R/W
b28 to b18	SID[10:0]	Standard ID	0: Corresponding SID[10:0] bit is not compared 1: Corresponding SID[10:0] bit is compared	R/W
b31 to b29	—	Reserved	The read value is undefined. The write value should be 0.	R/W

For the mask function in FIFO mailbox mode, refer to section 39.6, 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.

39.2.4 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

	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 39.6, Acceptance Filtering and Masking Functions.
Write to FIDCR0 and FIDCR1 in CAN reset mode or CAN halt mode.

EID[17:0] Bits (Extended ID)

The EID[17:0] bits set the extended ID of data frames and remote frames.
These bits are used to receive extended ID messages.

SID[10:0] Bits (Standard ID)

The SID[10:0] bits set the standard ID of data frames and remote frames.
These bits are used to receive both standard ID and extended ID messages.

RTR Bit (Remote Transmission Request)

The RTR bit sets the specified frame format of data frames or remote frames.

- When both RTR bits in FIDCR0 and FIDCR1 are set to 0, only data frames can be received.
- When both RTR bits in FIDCR0 and FIDCR1 are set to 1, only remote frames can be received.
- When the RTR bits in FIDCR0 and FIDCR1 are set to 0 or 1 individually, both data frames and remote frames can be received.

IDE Bit (ID Extension)

The IDE bit sets the ID format to standard ID or extended ID. The IDE bit is enabled when the IDFM[1:0] bits in CTLR are 10b (mixed ID mode).

- When both IDE bits in FIDCR0 and FIDCR1 are set to 0, only standard ID frames can be received.
- When both IDE bits in FIDCR0 and FIDCR1 are set to 1, only extended ID frames can be received.
- When the IDE bits in FIDCR0 and FIDCR1 are set to 0 or 1 individually, both standard ID and extended ID frames can be received.

39.2.5 Mask Invalid Register (MKIVLR)

Address(es): CAN0.MKIVLR 0009 0428h, CAN1.MKIVLR 0009 1428h

	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.

39.2.6 Mailbox Register j (MBj) (j = 0 to 31)

Table 39.4 lists the CANi mailbox memory mapping, and Table 39.5 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 39.4 for detailed register addresses.

Table 39.4 CANi Mailbox Memory Mapping

Address		Message Content
CAN0	**CAN1**	**Memory Mapping**
0009 0200h + 16 × j + 0	0009 1200h + 16 × j + 0	IDE, RTR, SID10 to SID6
0009 0200h + 16 × j + 1	0009 1200h + 16 × j + 1	SID5 to SID0, EID17, EID16
0009 0200h + 16 × j + 2	0009 1200h + 16 × j + 2	EID15 to EID8
0009 0200h + 16 × j + 3	0009 1200h + 16 × j + 3	EID7 to EID0
0009 0200h + 16 × j + 4	0009 1200h + 16 × j + 4	—
0009 0200h + 16 × j + 5	0009 1200h + 16 × j + 5	Data length code (DLC[3:0])
0009 0200h + 16 × j + 6	0009 1200h + 16 × j + 6	Data byte 0
0009 0200h + 16 × j + 7	0009 1200h + 16 × j + 7	Data byte 1
0009 0200h + 16 × j + 8	0009 1200h + 16 × j + 8	Data byte 2
0009 0200h + 16 × j + 9	0009 1200h + 16 × j + 9	Data byte 3
0009 0200h + 16 × j + 10	0009 1200h + 16 × j + 10	Data byte 4
0009 0200h + 16 × j + 11	0009 1200h + 16 × j + 11	Data byte 5
0009 0200h + 16 × j + 12	0009 1200h + 16 × j + 12	Data byte 6
0009 0200h + 16 × j + 13	0009 1200h + 16 × j + 13	Data byte 7
0009 0200h + 16 × j + 14	0009 1200h + 16 × j + 14	Time stamp upper byte
0009 0200h + 16 × j + 15	0009 1200h + 16 × j + 15	Time stamp lower byte

Note: When accessing to the content of the MBj register, access to an address of a multiple of 4 (the suffix of the address is 0h, 4h, 8h, or Ch) in 32-bit units and to an even address in 16-bit units.

Table 39.5 CAN Data Frame Configuration

SID10 to SID6	SID5 to SID0	EID17 to EID16	EID15 to EID8	EID7 to EID0	DLC3 to DLC1	DATA0	DATA1	...	DATA7

The previous value of each mailbox is retained unless a new message is received.

Address(es): CAN0.MB0 to CAN0.MB31 0009 0200h to 0009 03FFh, CAN1.MB0 to CAN1.MB31 0009 1200h to 0009 13FFh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	IDE	RTR	—	SID[10:0]											EID[17:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EID[17:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b17 to b0	EID[17:0]	Extended ID*1	0: Corresponding EID[17:0] bits are 0 1: Corresponding EID[17:0] bits are 1	R/W
b28 to b18	SID[10:0]	Standard ID	0: Corresponding SID[10:0] bits are 0 1: Corresponding SID[10:0] bits are 1	R/W
b29	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b30	RTR	Remote Frame Request	0: Data frame 1: Remote frame	R/W
b31	IDE	ID Extension*2	0: Standard ID 1: Extended ID	R/W

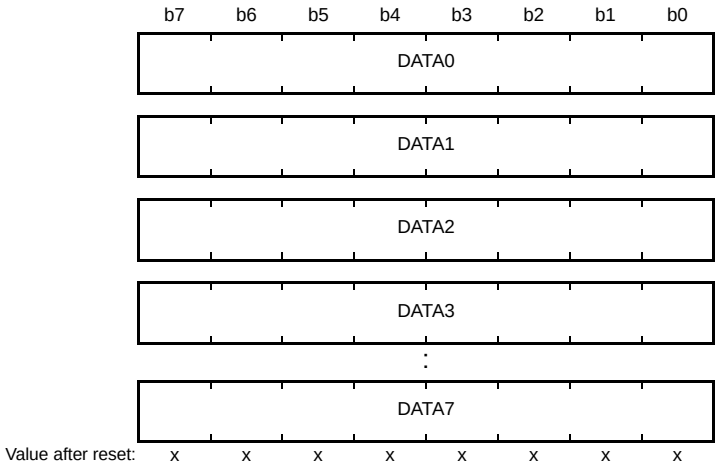
Note 1. If the mailbox has received a standard ID message, the EID bits in the mailbox are undefined.
Note 2. The IDE bit is enabled when the IDFM[1:0] bits in CTLR are 10b (mixed ID mode). When the IDFM[1:0] bits are not 10b, it should be written with 0 and read as 0.

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

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DLC[3:0]	Data Length Code*1	b3 b0 0 0 0 0: Data length = 0 byte 0 0 0 1: Data length = 1 byte 0 0 1 0: Data length = 2 bytes 0 0 1 1: Data length = 3 bytes 0 1 0 0: Data length = 4 bytes 0 1 0 1: Data length = 5 bytes 0 1 1 0: Data length = 6 bytes 0 1 1 1: Data length = 7 bytes 1 x x x: Data length = 8 bytes	R/W
b15 to b4	—	Reserved	The read value is undefined. The write value should be 0.	R/W

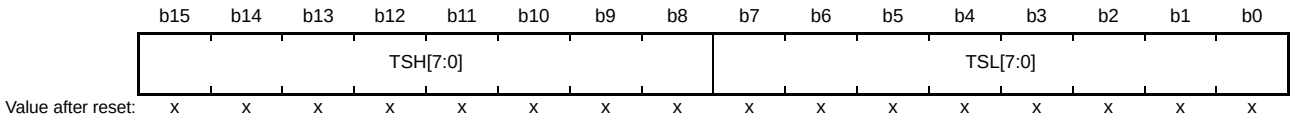
x: Don't care
Note 1. If the mailbox has received a message whose data length set by the DLC[3:0] bits is less than 8 bytes, the values of DATA larger than the data length set by the DLC[3:0] bits in the mailbox are undefined.



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DATA0 to DATA7	Data Bytes 0 to 7*1, *2, *3	DATA0 to DATA7 store the transmitted or received CAN message data. Transmission or reception starts from DATA0. The bit order on the CAN bus is MSB first, and transmission or reception starts from bit 7.	R/W

Note 1. If the mailbox has received a message with n bytes less than 8 bytes, the values of DATAn to DATA7 in the mailbox are undefined.
Note 2. If the mailbox has received a remote frame, the previous values of DATA0 to DATA7 in the mailbox are retained.
Note 3. DATA0 to DATA3 or DATA4 to DATA7 cannot be accessed in 32-bit units at one time. The access must be divided into 3 times: DATA0 to DATA1, DATA2 to DATA5, and DATA6 to DATA7, or 4 times: DATA0 to DATA1, DATA2 to DATA3, DATA4 to DATA5, and DATA6 to DATA7.



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TSL[7:0]	Time Stamp Lower Byte	Bits TSH[7:0] and TSL[7:0] store the counter value of the time stamp when received messages are stored in the mailbox.	R/W
b15 to b8	TSH[7:0]	Time Stamp Higher Byte		R/W

EID[17:0] Bits (Extended ID)

The EID[17:0] bits set the extended ID of data frames and remote frames.
These bits are used to transmit or receive extended ID messages.

SID[10:0] Bits (Standard ID)

The SID[10:0] bits set the standard ID of data frames and remote frames.
These bits are used to transmit or receive both standard ID and extended ID messages.

RTR Bit (Remote Frame Request)

The RTR bit sets the frame format of data frames or remote frames.

- Receive mailbox receives only frames with the format specified by the RTR bit.
- Transmit mailbox transmits according to the frame format specified by the RTR bit.
- Receive FIFO mailbox receives the data frame, remote frame, or both frames specified by the RTR bit in FIDCR0 and FIDCR1.
- Transmit FIFO mailbox transmits the data frame or remote frame specified by the RTR bit in the relevant transmit message.

IDE Bit (ID Extension)

The IDE bit sets the ID format of standard IDs or extended IDs. The IDE bit is enabled when the IDFM[1:0] bits in CTLR is 10b (mixed ID mode).

- Receive mailbox receives only the ID format specified by the IDE bit.
- Transmit mailbox transmits with the ID format specified by the IDE bit.
- Receive FIFO mailbox receives messages with the standard ID and extended ID specified by the IDE bit in FIDCR0 and FIDCR1.
- Transmit FIFO mailbox transmits messages with the standard ID or extended ID specified by the IDE bit in the relevant transmit message.

DLC[3:0] Bits (Data Length Code)

The DLC[3:0] bits specify the number of bytes of data to be transmitted in data frames. When a remote frame is used to request data, this field specifies the requested number of bytes of data.
When a data frame is received, the number of bytes received is stored in this field. When a remote frame is received, this field is used to store the number requested by the frame.

39.2.7 Mailbox Interrupt Enable Register (MIER)

Address(es): CAN0.MIER 0009 042Ch, CAN1.MIER 0009 142Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MB31	MB30	MB29	MB28	MB27	MB26	MB25	MB24	MB23	MB22	MB21	MB20	MB19	MB18	MB17	MB16
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MB15	MB14	MB13	MB12	MB11	MB10	MB9	MB8	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

- Normal mailbox mode

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	MB31 to MB0	Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled Bit 31 corresponds to mailbox 31 (MB31), and bit 0 corresponds to mailbox 0 (MB0).	R/W

- FIFO mailbox mode

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	MB23 to MB0	Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled Bit 23 corresponds to mailbox 23 (MB23), and bit 0 corresponds to mailbox 0 (MB0).	R/W
b24	MB24	Transmit FIFO Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/W
b25	MB25	Transmit FIFO Interrupt Generation Timing Control	0: Every time transmission is completed 1: When the transmit FIFO becomes empty due to completion of transmission	R/W
b27, b26	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b28	MB28	Receive FIFO Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/W
b29	MB29	Receive FIFO Interrupt Generation Timing Control*1	0: Every time reception is completed 1: When the receive FIFO becomes buffer warning by completion of reception	R/W
b31, b30	—	Reserved	The read value is undefined. The write value should be 0.	R/W

Note 1. No interrupt request is generated when the receive FIFO becomes buffer warning from full. “Buffer warning” indicates a state in which the third message is stored in the receive FIFO.

MIER can individually enable interrupts for each mailbox.
In normal mailbox mode (all bits) and in FIFO mailbox mode (bits 24 to 0 in MIER), each bit corresponds to the mailbox with the related number. These bits enable or disable transmission/reception complete interrupts for the corresponding mailboxes.

- Bit 0 in MIER corresponds to mailbox 0 (MB0).
- Bit 31 in MIER corresponds to mailbox 31 (MB31).

In FIFO mailbox mode, bits 29, 28, 25, and 24 of MIER specify whether transmit/receive FIFO interrupts are enabled/disabled and the timing when interrupt requests are generated.
Write to MIER only when the related MCTLj (j = 0 to 31) is 00h and the corresponding mailbox is not processing a transmission or reception abort request. In FIFO mailbox mode, change the bits in MIER for the related FIFO only when the TFE bit in TFCR is 0 and the TFEST flag is 1, and the RFE bit in RFCR is 0 and the RFEST flag in RFCR is 1.

39.2.8 Message Control Register j (MCTLj) (j = 0 to 31)

Address(es): CAN0.MCTL0 to CAN0.MCTL31 0009 0820h to 0009 083Fh, CAN1.MCTL0 to CAN1.MCTL31 0009 1820h to 0009 183Fh

- Transmit mode (when the TRMREQ bit is 1 and the RECREQ bit is 0)

	b7	b6	b5	b4	b3	b2	b1	b0
	TRMREQ	RECREQ	—	ONESHOT	—	TRMABT	TRMACTIVE	SENTDATA
Value after reset:	0	0	0	0	0	0	0	0

- Receive mode (when the TRMREQ bit is 0 and the RECREQ bit is 1)

	b7	b6	b5	b4	b3	b2	b1	b0
	TRMREQ	RECREQ	—	ONESHOT	—	MSGLOST	INVALDATA	NEWDATA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SENTDATA	Transmission Complete Flag *1, *2	0: Transmission is not completed 1: Transmission is completed	R/W
	NEWDATA	Reception Complete Flag *1, *2	0: No data has been received or 0 is written to the NEWDATA flag 1: A new message is being stored or has been stored to the mailbox	R/W
b1	TRMACTIVE	Transmission-in-Progress Status Flag	(Transmit mailbox setting enabled) 0: Transmission is pending or transmission is not requested 1: From acceptance of transmission request to completion of transmission, or error/arbitration-lost	R
	INVALDATA	Reception-in-Progress Status Flag	(Receive mailbox setting enabled) 0: Message valid 1: Message being updated	R
b2	TRMABT	Transmission Abort Complete Flag*1, *2	(Transmit mailbox setting enabled) 0: Transmission has started, transmission abort failed because transmission is completed, or transmission abort is not requested 1: Transmission abort is completed	R/W
	MSGLOST	Message Lost Flag*1, *2	(Receive mailbox setting enabled) 0: Message is not overwritten or overrun 1: Message is overwritten or overrun	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	ONESHOT	One-Shot Enable*3	0: One-shot reception or one-shot transmission disabled 1: One-shot reception or one-shot transmission enabled	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	RECREQ	Receive Mailbox Request *2, *3, *4, *5	0: Not configured for reception 1: Configured for reception	R/W
b7	TRMREQ	Transmit Mailbox Request *2, *4	0: Not configured for transmission 1: Configured for transmission	R/W

Note 1. Write 0 only. Writing 1 has no effect.
Note 2. When writing 0 to bits NEWDATA, SENTDATA, MSGLOST, TRMABT, RECREQ, and TRMREQ by a program, do not use the logic operation instruction (AND). Write 0 to only the specified bit and write 1 to the other bits before using the transfer (MOV) instruction.
Note 3. To enter one-shot receive mode, write 1 to the ONESHOT bit at the same time as setting the RECREQ bit to 1.
To exit one-shot receive mode, write 0 to the ONESHOT bit after writing 0 to the RECREQ bit and confirming that it has been set to 0.
To enter one-shot transmit mode, write 1 to the ONESHOT bit at the same time as setting the TRMREQ bit to 1.
To exit one-shot transmit mode, write 0 to the ONESHOT bit after the message has been transmitted or aborted.
Note 4. Do not set both the RECREQ and TRMREQ bits to 1.
Note 5. When setting the RECREQ bit to 0, set bits MSGLOST, NEWDATA, and RECREQ to 0 simultaneously.

Write to the MCTLj in CAN operation mode or CAN halt mode.
Do not use MCTL24 to MCTL31 in FIFO mailbox mode.

SENTDATA Flag (Transmission Complete Flag)

The SENTDATA flag is set to 1 when data transmission from the corresponding mailbox is completed. The SENTDATA flag is set to 0 by writing 0 by a program.
To set the SENTDATA flag to 0, first set the TRMREQ bit to 0. Bits SENTDATA and TRMREQ cannot be set to 0 simultaneously. To transmit a new message from the corresponding mailbox, set the SENTDATA flag to 0.

NEWDATA Flag (Reception Complete Flag)

The NEWDATA flag is set to 1 when a new message is being stored or has been stored to the mailbox. The timing for setting this bit to 1 is simultaneous with the INVALDATA flag. The NEWDATA flag is set to 0 by writing 0 by a program. The NEWDATA flag cannot be set to 0 by writing 0 by a program while the related INVALDATA flag is 1.

TRMACTIVE Flag (Transmission-in-Progress Status Flag)

The TRMACTIVE flag is set to 1 when the corresponding mailbox of the CAN module begins transmitting a message. The TRMACTIVE flag is set to 0 when the CAN module has lost CAN bus arbitration, a CAN bus error occurs, or data transmission is completed.

INVALDATA Flag (Reception-in-Progress Status Flag)

After the completion of a message reception, the INVALDATA flag is set to 1 while the received message is being updated into the corresponding mailbox. The INVALDATA flag is set to 0 immediately after the message has been stored. If the mailbox is read while the INVALDATA flag is 1, the data is undefined.

TRMABT Flag (Transmission Abort Complete Flag)

The TRMABT flag is set to 1 in the following cases:

- Following a transmission abort request, when the transmission abort is completed before starting transmission.
- Following a transmission abort request, when the CAN module detects CAN bus arbitration-lost or a CAN bus error.
- In one-shot transmission mode (RECREQ bit = 0, TRMREQ bit = 1, and ONESHOT bit = 1), when the CAN module detects CAN bus arbitration-lost or a CAN bus error.

The TRMABT flag is not set to 1 when data transmission is completed. In this case, the SENTDATA flag is set to 1. The TRMABT flag is set to 0 by writing 0 by a program.

MSGLOST Flag (Message Lost Flag)

The MSGLOST flag is set to 1 when the mailbox is overwritten or overrun by a new received message while the NEWDATA flag is 1. The MSGLOST flag is set to 1 at the end of the 6th bit of EOF. The MSGLOST flag is set to 0 by writing 0 by a program.
In both overwrite and overrun modes, the MSGLOST flag cannot be set to 0 by writing 0 by a program during five peripheral module clock (PCLKB) cycles following the sixth bit of EOF.

ONESHOT Bit (One-Shot Enable)

The ONESHOT bit can be used in the following two ways, receive mode and transmit mode.

- One-shot receive mode
  When the ONESHOT bit is set to 1 in receive mode (RECREQ bit = 1 and TRMREQ bit = 0), the mailbox receives a message only one time. (The mailbox does not behave as a receive mailbox after having received a message one time.) The behavior of flags NEWDATA and INVALDATA is the same as in normal receive mode. In one-shot receive mode, the MSGLOST flag is not set to 1. To set the ONESHOT bit to 0, first write 0 to the RECREQ bit and ensure that it has been set to 0.
- One-shot transmit mode
  When the ONESHOT bit is set to 1 in transmit mode (RECREQ bit = 0 and TRMREQ bit = 1), the CAN module transmits a message only one time. (The CAN module does not transmit the message again if a CAN bus error or CAN bus arbitration-lost occurs.) When transmission is completed, the SENTDATA flag is set to 1. If transmission is not completed due to a CAN bus error or CAN bus arbitration-lost, the TRMABT flag is set to 1. Set the ONESHOT bit to 0 after the SENTDATA or TRMABT flag is set to 1.

RECREQ Bit (Receive Mailbox Request)

The RECREQ bit selects receive modes listed in Table 39.10.
When the RECREQ bit is set to 1, the corresponding mailbox is configured for reception of a data frame or a remote frame.
When the RECREQ bit is set to 0, the corresponding mailbox is not configured for reception of a data frame or a remote frame.
Due to hardware protection, the RECREQ bit cannot be set to 0 by writing 0 by a program during the following period.

- Hardware protection is started
  From the acceptance filter processing (the beginning of CRC field)

- Hardware protection is released
  – For the mailbox that is specified to receive the incoming message, after the received data is stored into the mailbox or a CAN bus error occurs (i.e. maximum period of hardware protection is from the beginning of CRC field to the end of the 7th bit of EOF)
  – For the other mailboxes, after the acceptance filter processing
  – If no mailbox is specified to receive the message, after the acceptance filter processing
  When setting the RECREQ bit to 1, do not set the TRMREQ bit to 1. To change the configuration of a mailbox from transmission to reception, first abort the transmission and then set flags SENTDATA and TRMABT to 0 before changing to reception.

TRMREQ Bit (Transmit Mailbox Request)

The TRMREQ bit selects transmit modes listed in Table 39.10.
When the TRMREQ bit is set to 1, the corresponding mailbox is configured for transmission of a data frame or a remote frame.
When the TRMREQ bit is set to 0, the corresponding mailbox is not configured for transmission of a data frame or a remote frame.
If the TRMREQ bit is changed from 1 to 0 to cancel the corresponding transmission request, either the TRMABT or SENTDATA flag is set to 1. When setting the TRMREQ bit to 1, do not set the RECREQ bit to 1. To change the configuration of a mailbox from reception to transmission, first abort the reception and then set flags NEWDATA and MSGLOST to 0 before changing to transmission.

39.2.9 Receive FIFO Control Register (RFCR)

Address(es): CAN0.RFCR 0009 0848h, CAN1.RFCR 0009 1848h

	b7	b6	b5	b4	b3	b2	b1	b0
	RFEST	RFWST	RFFST	RFMLF	RFUST[2:0]			RFE
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFE	Receive FIFO Enable	0: Receive FIFO disabled 1: Receive FIFO enabled	R/W
b3 to b1	RFUST[2:0]	Receive FIFO Unread Message Number Status Flag	b3 b1 0 0 0: No unread message 0 0 1: 1 unread message 0 1 0: 2 unread messages 0 1 1: 3 unread messages 1 0 0: 4 unread messages 1 0 1: Reserved 1 1 0: Reserved 1 1 1: Reserved	R
b4	RFMLF	Receive FIFO Message Lost Flag	0: No receive FIFO message lost has occurred 1: Receive FIFO message lost has occurred	R/W
b5	RFFST	Receive FIFO Full Status Flag	0: Receive FIFO is not full 1: Receive FIFO is full (4 unread messages)	R
b6	RFWST	Receive FIFO Buffer Warning Status Flag	0: Receive FIFO is not buffer warning 1: Receive FIFO is buffer warning (3 unread messages)	R
b7	RFEST	Receive FIFO Empty Status Flag	0: Unread message in receive FIFO 1: No unread message in receive FIFO	R

Write to RFCR in CAN operation mode or CAN halt mode.

RFE Bit (Receive FIFO Enable)

When the RFE bit is set to 1, the receive FIFO is enabled.

When the RFE bit is set to 0, the receive FIFO is disabled for reception and becomes empty (RFEST flag = 1). Write 0 to the RFE bit simultaneously with setting the RFMLF flag.

Do not set this bit to 1 in normal mailbox mode (MBM bit in CTLR = 0). Due to hardware protection, the RFE bit is not set to 0 by writing 0 by a program during the following period.

- Hardware protection is started
  - From the acceptance filter processing (the beginning of CRC field)
- Hardware protection is released
  - If the receive FIFO is specified to receive the incoming message, after the received data is stored into the receive FIFO or a CAN bus error occurs (i.e. maximum period of hardware protection is from the beginning of CRC field to the end of 7th bit of EOF)
  - If the receive FIFO is not specified to receive the message, after the acceptance filter processing

RFUST[2:0] Flags (Receive FIFO Unread Message Number Status Flag)

The RFUST[2:0] flags indicate the number of unread messages in the receive FIFO.

The value of the RFUST[2:0] flags is initialized to 000b when the RFE bit is set to 0.

RFMLF Flag (Receive FIFO Message Lost Flag)

The RFMLF flag is set to 1 (receive FIFO message lost has occurred) when the receive FIFO receives a new message and the receive FIFO is full. The timing for setting this bit to 1 is at the end of the 6th bit of EOF.
The RFMLF flag is set to 0 by writing 0 by a program (writing 1 has no effect). In both overwrite and overrun modes, the RFMLF flag cannot be set to 0 (receive FIFO message lost has not occurred) by writing 0 by a program due to hardware protection during five peripheral module clock (PCLKB) cycles following the sixth bit of EOF, if the receive FIFO is full and determined to receive a message.

RFFST Flag (Receive FIFO Full Status Flag)

The RFFST flag is set to 1 (receive FIFO is full) when the number of unread messages in the receive FIFO is 4. The RFFST flag is 0 (receive FIFO is not full) when the number of unread messages in the receive FIFO is less than 4. The RFFST flag is set to 0 when the RFE bit is 0.

RFWST Flag (Receive FIFO Buffer Warning Status Flag)

The RFWST flag is set to 1 (receive FIFO is buffer warning) when the number of unread messages in the receive FIFO is 3. The RFWST flag is 0 (receive FIFO is not buffer warning) when the number of unread messages in the receive FIFO is less than 3 or equal to 4. The RFWST flag is set to 0 when the RFE bit is 0.

RFEST Flag (Receive FIFO Empty Status Flag)

The RFEST flag is set to 1 (no unread message in receive FIFO) when the number of unread messages in the receive FIFO is 0. The RFEST flag is set to 1 when the RFE bit is set to 0. The RFEST flag is set to 0 (unread message in receive FIFO) when the number of unread messages in the receive FIFO is one or more.

Figure 39.2 shows the receive FIFO mailbox operation.

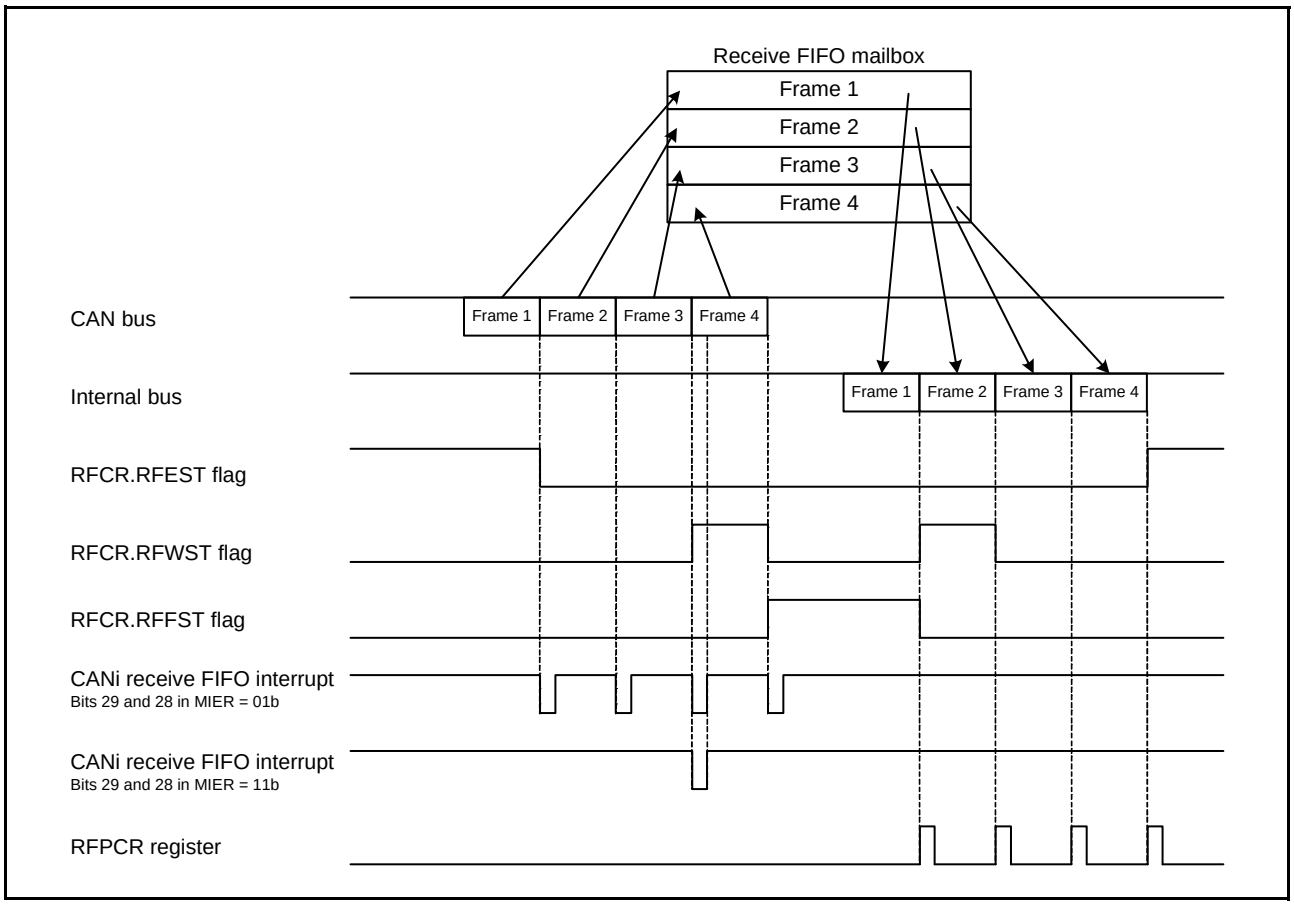


Figure 39.2 Receive FIFO Mailbox Operation (Bits 29 and 28 in MIER = 01b or 11b)

39.2.10 Receive FIFO Pointer Control Register (RFPCR)

Address(es): CAN0.RFPCR 0009 0849h, CAN1.RFPCR 0009 1849h

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

x: Undefined

Bit	Description	R/W
b7 to b0	The CPU-side pointer for the receive FIFO is incremented by writing FFh to RFPCR.	W

When the receive FIFO is not empty, write FFh to RFPCR by a program to increment the CPU-side pointer for the receive FIFO to the next mailbox location.
Do not write to RFPCR when the RFE bit in RFCR is 0 (receive FIFO disabled).
Both the CAN-side pointer and the CPU-side pointer are incremented when a new message is received and the RFFST flag is 1 (receive FIFO is full) in overwrite mode. When the RFMLF flag is 1 in this condition, the CPU-side pointer cannot be incremented by writing to RFPCR by a program.

39.2.11 Transmit FIFO Control Register (TFCR)

Address(es): CAN0.TFCR 0009 084Ah, CAN1.TFCR 0009 184Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	TFEST	TFFST	—	—	TFUST[2:0]			TFE
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TFE	Transmit FIFO Enable	0: Transmit FIFO disabled 1: Transmit FIFO enabled	R/W
b3 to b1	TFUST[2:0]	Transmit FIFO Unsent Message Number Status Flag	b3 b2 b1 0 0 0: No unsent message 0 0 1: 1 unsent message 0 1 0: 2 unsent messages 0 1 1: 3 unsent messages 1 0 0: 4 unsent messages 1 0 1: Reserved 1 1 0: Reserved 1 1 1: Reserved	R
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	TFFST	Transmit FIFO Full Status Flag	0: Transmit FIFO is not full 1: Transmit FIFO is full (4 unsent messages)	R
b7	TFEST	Transmit FIFO Empty Status Flag	0: Unsent message in transmit FIFO 1: No unsent message in transmit FIFO	R

Write to TFCR in CAN operation mode or CAN halt mode.

TFE Bit (Transmit FIFO Enable)

When the TFE bit is set to 1, the transmit FIFO is enabled.
When the TFE bit is set to 0, the transmit FIFO becomes empty (TFEST flag = 1) and then unsent messages from the transmit FIFO are lost as described below:

- Immediately if a message from the transmit FIFO is not scheduled for the next transmission or during transmission
- Following the completion of transmission, a CAN bus error, CAN bus arbitration-lost, or entry to CAN halt mode if a message from the transmit FIFO is scheduled for the next transmission or already during transmission

Before setting the TFE bit to 1 again, ensure that the TFEST flag has been set to 1. After setting the TFE bit to 1, write transmit data into MB24.
Do not set the TFE bit to 1 in normal mailbox mode (MBM bit in CTLR = 0).

TFUST[2:0] Flags (Transmit FIFO Unsent Message Number Status Flag)

The TFUST[2:0] flags indicate the number of unsent messages in the transmit FIFO.
The TFUST[2:0] flags are set to 000b after TFE bit is cleared to 0 and transmission is aborted or completed.

TFFST Flag (Transmit FIFO Full Status Flag)

The TFFST flag is set to 1 (transmit FIFO is full) when the number of unsent messages in the transmit FIFO is 4. The TFFST flag is set to 0 (transmit FIFO is not full) when the number of unsent messages in the transmit FIFO is less than 4. The TFFST flag is set to 0 when transmission from the transmit FIFO has been aborted.

TFEST Flag (Transmit FIFO Empty Status Flag)

The TFEST flag is set to 1 (no message in transmit FIFO) when the number of unsent messages in the transmit FIFO is 0. The TFEST flag is set to 1 when transmission from the transmit FIFO has been aborted. The TFEST flag is set to 0 (message in transmit FIFO) when the number of unsent messages in the transmit FIFO is not 0.

Figure 39.3 shows the transmit FIFO mailbox operation.

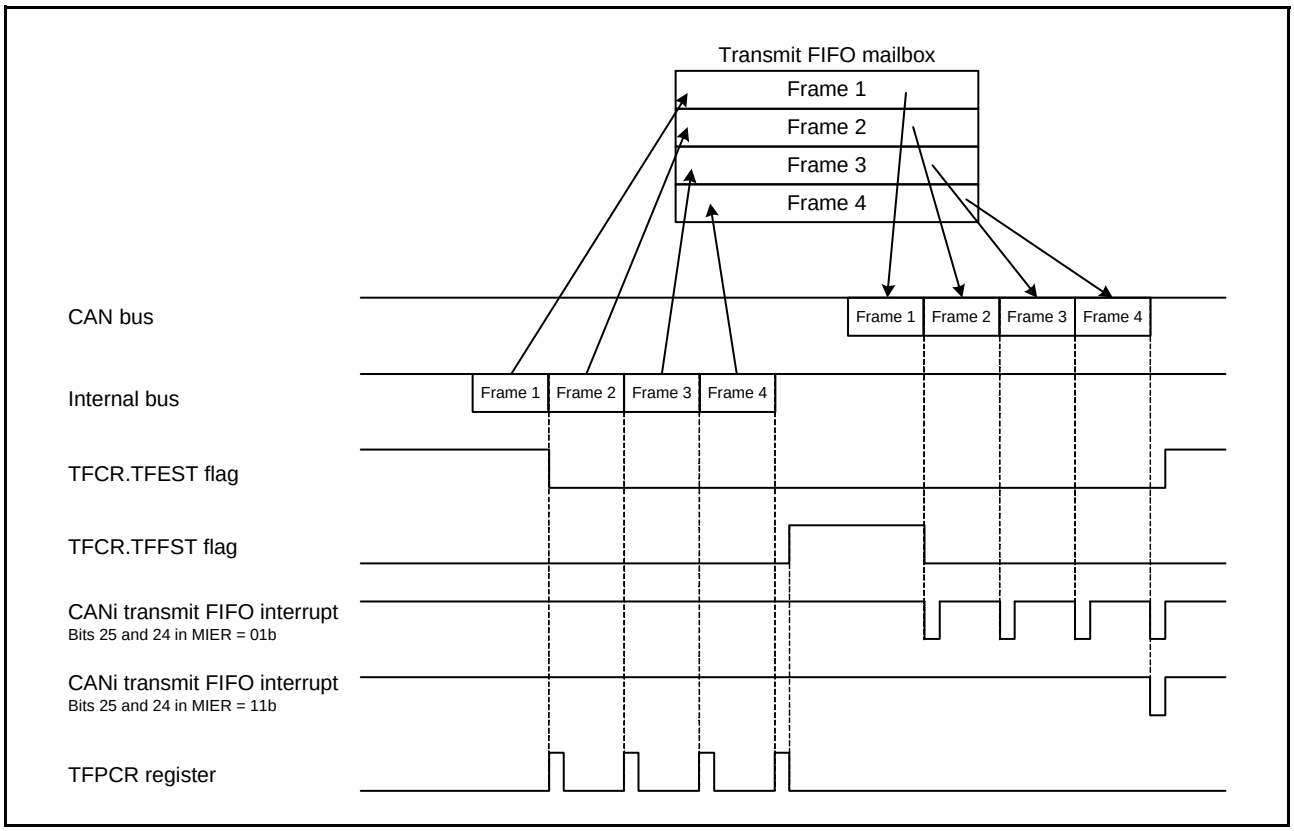


Figure 39.3 Transmit FIFO Mailbox Operation (Bits 25 and 24 in MIER = 01b or 11b)

39.2.12 Transmit FIFO Pointer Control Register (TFPCR)

Address(es): CAN0.TFPCR 0009 084Bh, CAN1.TFPCR 0009 184Bh

	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).

39.2.13 Status Register (STR)

Address(es): CAN0.STR 0009 0842h, CAN1.STR 0009 1842h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	RECST	TRMST	BOST	EPST	SLPST	HLTST	RSTST	EST	TABST	FMLST	NMLST	TFST	RFST	SDST	NDST
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	NDST	NEWDATA Status Flag	0: No mailbox with NEWDATA flag = 1 1: Mailbox(es) with NEWDATA flag = 1	R
b1	SDST	SENTDATA Status Flag	0: No mailbox with SENTDATA flag = 1 1: Mailbox(es) with SENTDATA flag = 1	R
b2	RFST	Receive FIFO Status Flag	0: No message in receive FIFO (empty) 1: Message in receive FIFO	R
b3	TFST	Transmit FIFO Status Flag	0: Transmit FIFO is full 1: Transmit FIFO is not full	R
b4	NMLST	Normal Mailbox Message Lost Status Flag	0: No mailbox with MSGLOST flag = 1 1: Mailbox(es) with MSGLOST flag = 1	R
b5	FMLST	FIFO Mailbox Message Lost Status Flag	0: RFMLF flag = 0 1: RFMLF flag = 1	R
b6	TABST	Transmission Abort Status Flag	0: No mailbox with TRMABT flag = 1 1: Mailbox(es) with TRMABT flag = 1	R
b7	EST	Error Status Flag	0: No error occurred 1: Error occurred	R
b8	RSTST	CAN Reset Status Flag	0: Not in CAN reset mode 1: In CAN reset mode	R
b9	HLTST	CAN Halt Status Flag	0: Not in CAN halt mode 1: In CAN halt mode	R
b10	SLPST	CAN Sleep Status Flag	0: Not in CAN sleep mode 1: In CAN sleep mode	R
b11	EPST	Error-Passive Status Flag	0: Not in error-passive state 1: In error-passive state	R
b12	BOST	Bus-Off Status Flag	0: Not in bus-off state 1: In bus-off state	R
b13	TRMST	Transmit Status Flag (transmitter)	0: Bus idle or reception in progress 1: Transmission in progress or in bus-off state	R
b14	RECST	Receive Status Flag (receiver)	0: Bus idle or transmission in progress 1: Reception in progress	R
b15	—	Reserved	The read value is 0.	R

NDST Flag (NEWDATA Status Flag)

The NDST flag is set to 1 when at least one NEWDATA flag in MCTLj (j = 0 to 31) is 1 regardless of the value of MIER. The NDST flag is set to 0 when all NEWDATA flags are 0.

SDST Flag (SENTDATA Status Flag)

The SDST flag is set to 1 when at least one SENTDATA flag in MCTLj (j = 0 to 31) is 1 regardless of the value of MIER. The SDST flag is set to 0 when all SENTDATA flags are 0.

RFST Flag (Receive FIFO Status Flag)

The RFST flag is set to 1 when the receive FIFO is not empty. The RFST flag is set to 0 when the receive FIFO is empty or normal mailbox mode is selected.

TFST Flag (Transmit FIFO Status Flag)

The TFST flag is set to 1 when the transmit FIFO is not full. The TFST flag is set to 0 when the transmit FIFO is full or normal mailbox mode is selected.

NMLST Flag (Normal Mailbox Message Lost Status Flag)

The NMLST flag is set to 1 when at least one MSGLOST flag in MCTLj (j = 0 to 31) is 1 regardless of the value of MIER. The NMLST flag is set to 0 when all MSGLOST flags are 0.

FMLST Flag (FIFO Mailbox Message Lost Status Flag)

The FMLST flag is set to 1 when the RFMLF flag in RFCR is 1 regardless of the value of MIER. The FMLST flag is set to 0 when the RFMLF flag is 0.

TABST Flag (Transmission Abort Status Flag)

The TABST flag is set to 1 when at least one TRMABT flag in MCTLj (j = 0 to 31) is 1 regardless of the value of MIER. The TABST flag is set to 0 when all TRMABT flags are 0.

EST Flag (Error Status Flag)

The EST flag is set to 1 when at least one error is detected by EIFR regardless of the value of EIER. The EST flag is set to 0 when no error is detected by EIFR.

RSTST Flag (CAN Reset Status Flag)

The RSTST flag is set to 1 when the CAN module is in CAN reset mode. The RSTST flag is 0 when the CAN module is not in CAN reset mode. Even when the state is changed from CAN reset mode to CAN sleep mode, the RSTST flag remains 1.

HLTST Flag (CAN Halt Status Flag)

The HLTST flag is set to 1 when the CAN module is in CAN halt mode. The HLTST flag is set to 0 when the CAN module is not in CAN halt mode. Even when the state is changed from CAN halt mode to CAN sleep mode, the HLTST flag remains 1.

SLPST Flag (CAN Sleep Status Flag)

The SLPST flag is set to 1 when the CAN module is in CAN sleep mode. The SLPST flag is set to 0 when the CAN module is not in CAN sleep mode.

EPST Flag (Error-Passive Status Flag)

The EPST flag is set to 1 when the value of TECR or RECR exceeds 127 and the CAN module is in the error-passive state ($128 \leq TEC < 256$ or $128 \leq REC < 256$). The EPST flag is set to 0 when the CAN module is not in the error-passive state.

BOST Flag (Bus-Off Status Flag)

The BOST flag is set to 1 when the value of TECR exceeds 255 and the CAN module is in the bus-off state ($TEC \geq 256$). The BOST flag is set to 0 when the CAN module is not in the bus-off state.

TRMST Flag (Transmit Status Flag (transmitter))

The TRMST flag is set to 1 when the CAN module performs as a transmitter node or is in the bus-off state. The TRMST flag is set to 0 when the CAN module performs as a receiver node or is in the bus-idle state.

RECST Flag (Receive Status Flag (receiver))

The RECST flag is set to 1 when the CAN module performs as a receiver node. The RECST flag is set to 0 when the CAN module performs as a transmitter node or is in the bus-idle state.

39.2.14 Mailbox Search Mode Register (MSMR)

Address(es): CAN0.MSMR 0009 0853h, CAN1.MSMR 0009 1853h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	MBSM[1:0]	Mailbox Search Mode Select	b1 b0 0 0: Receive mailbox search mode 0 1: Transmit mailbox search mode 1 0: Message lost search mode 1 1: Channel search mode	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Write to MSMR in CAN operation mode or CAN halt mode.

MBSM[1:0] Bits (Mailbox Search Mode Select)

The MBSM[1:0] bits select the search mode for the mailbox search function.

When the MBSM[1:0] bits are 00b, receive mailbox search mode is selected. In this mode, the search targets are the NEWDATA flag in MCTLj (j = 0 to 31) for the normal mailbox and the RFEST flag in RFCR.

When the MBSM[1:0] bits are 01b, transmit mailbox search mode is selected. In this mode, the search target is the SENTDATA flag in MCTLj.

When the MBSM[1:0] bits are 10b, message lost search mode is selected. In this mode, the search targets are the MSGLOST flag in MCTLj for the normal mailbox and the RFMLF flag in RFCR.

When the MBSM[1:0] bits are 11b, channel search mode is selected. In this mode, the search target is CSSR. Refer to section 39.2.16, Channel Search Support Register (CSSR).

39.2.15 Mailbox Search Status Register (MSSR)

Address(es): CAN0.MSSR 0009 0852h, CAN1.MSSR 0009 1852h

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

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	MBNST[4:0]	Search Result Mailbox Number Status Flag	These bits output the smallest mailbox number that is searched in each mode of MSMR.	R
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SEST	Search Result Status Flag	0: Search result found 1: No search result	R

MBNST[4:0] Flags (Search Result Mailbox Number Status Flag)

The MBNST[4:0] flags output the smallest mailbox number that is searched in each mode of MSMR. In receive mailbox search mode, transmit mailbox search mode, and message lost search mode, the value of the mailbox i.e., the search result to be output, is updated as described below:

- When the NEWDATA, SENTDATA or MSGLOST flag for the output mailbox is set to 0
- When the NEWDATA, SENTDATA or MSGLOST flag for a higher-priority mailbox is set to 1

If the MBSM[1:0] bits are set to 00b (receive mailbox search mode) or 10b (message lost search mode), the receive FIFO (mailbox [28]) is output when the receive FIFO is not empty and there are no unread received messages or no lost messages in any of the normal mailboxes (mailboxes [0] to [23]). If the MBSM[1:0] bits are set to 01b (transmit mailbox search mode), the transmit FIFO (mailbox [24]) is not output. Table 39.6 lists the behavior of the MBNST[4:0] flags in FIFO mailbox mode.

In channel search mode, the MBNST[4:0] flags output the corresponding channel number. After MSSR is read by a program, the next target channel number is output.

SEST Flag (Search Result Status Flag)

The SEST flag is set to 1 (no search result) when no corresponding mailbox is found after searching all mailboxes. For example, in transmit mailbox search mode, the SEST flag is set to 1 when no SENTDATA flag for mailboxes is 1. The SEST flag is set to 0 when at least one SENTDATA flag is 1. When the SEST flag is 1, the value of the MBNST[4:0] flags is undefined.

Table 39.6 Behavior of MBNST[4:0] Flags in FIFO Mailbox Mode

MBSM[1:0] Bits	Mailbox [24] (Transmit FIFO)	Mailbox [28] (Receive FIFO)
00b	Mailbox [24] is not output.	Mailbox [28] is output when no MCTLj.NEWDATA flag for the normal mailboxes is set to 1 (new message is being stored or has been stored to the mailbox) and the receive FIFO is not empty.
01b		Mailbox [28] is not output.
10b		Mailbox [28] is output when no MCTLj.MSGLOST flag for the normal mailboxes is set to 1 (message is overwritten or overrun) and the RFCR.RFMLF bit is set to 1 (receive FIFO message lost has occurred) in the receive FIFO.
11b		Mailbox [28] is not output.

39.2.16 Channel Search Support Register (CSSR)

Address(es): CAN0.CSSR 0009 0851h, CAN1.CSSR 0009 1851h

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

x: Undefined

Bit	Description	R/W
b7 to b0	When the value for the channel search is input, the channel number is output to MSSR.	R/W

The bits in CSSR, which are set to 1, are encoded by an 8/3 encoder (the LSB position has the higher priority) and output to the MBNST[4:0] flags in MSSR.
MSSR outputs the updated value whenever MSSR is read by a program.
Write to CSSR only when the MSMR.MBSM[1:0] bits are 11b (channel search mode). Write to CSSR in CAN operation mode or CAN halt mode.

Figure 39.4 shows the write and read of CSSR and MSSR.

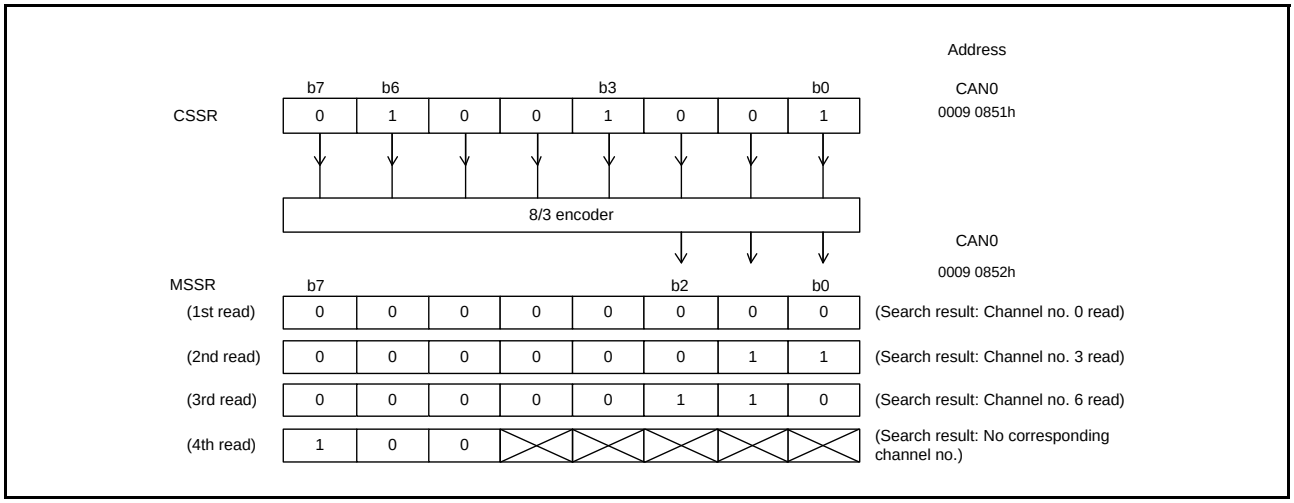


Figure 39.4 Write and Read of CSSR and MSSR

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.

39.2.17 Acceptance Filter Support Register (AFSR)

Address(es): CAN0.AFSR 0009 0856h, CAN1.AFSR 0009 1856h

	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 39.5 shows the write and read of AFSR.

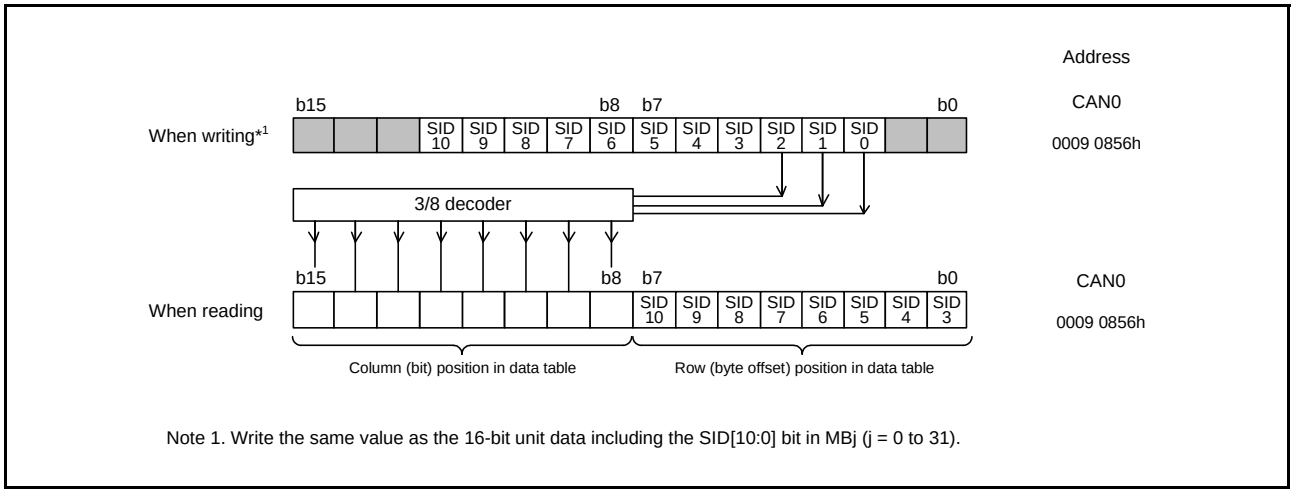


Figure 39.5 Write and Read of AFSR

39.2.18 Error Interrupt Enable Register (EIER)

Address(es): CAN0.EIER 0009 084Ch, CAN1.EIER 0009 184Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	BLIE	OLIE	ORIE	BORIE	BOEIE	EPIE	EWIE	BEIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BEIE	Bus Error Interrupt Enable	0: Bus error interrupt disabled 1: Bus error interrupt enabled	R/W
b1	EWIE	Error-Warning Interrupt Enable	0: Error-warning interrupt disabled 1: Error-warning interrupt enabled	R/W
b2	EPIE	Error-Passive Interrupt Enable	0: Error-passive interrupt disabled 1: Error-passive interrupt enabled	R/W
b3	BOEIE	Bus-Off Entry Interrupt Enable	0: Bus-off entry interrupt disabled 1: Bus-off entry interrupt enabled	R/W
b4	BORIE	Bus-Off Recovery Interrupt Enable	0: Bus-off recovery interrupt disabled 1: Bus-off recovery interrupt enabled	R/W
b5	ORIE	Overrun Interrupt Enable	0: Receive overrun interrupt disabled 1: Receive overrun interrupt enabled	R/W
b6	OLIE	Overload Frame Transmit Interrupt Enable	0: Overload frame transmit interrupt disabled 1: Overload frame transmit interrupt enabled	R/W
b7	BLIE	Bus Lock Interrupt Enable	0: Bus lock interrupt disabled 1: Bus lock interrupt enabled	R/W

EIER is used to enable or disable the error interrupt individually for each error interrupt source in EIFR.
Write to EIER in CAN reset mode.

BEIE Bit (Bus Error Interrupt Enable)

When the BEIE bit is 0, no error interrupt request is generated even if the BEIF flag in EIFR is set to 1. When the BEIE bit is 1, an error interrupt request is generated if the BEIF flag is set to 1.

EWIE Bit (Error-Warning Interrupt Enable)

When the EWIE bit is 0, no error interrupt request is generated even if the EWIF flag in EIFR is set to 1. When the EWIE bit is 1, an error interrupt request is generated if the EWIF flag is set to 1.

EPIE Bit (Error-Passive Interrupt Enable)

When the EPIE bit is 0, no error interrupt request is generated even if the EPIF flag in EIFR is set to 1. When the EPIE bit is 1, an error interrupt request is generated if the EPIF flag is set to 1.

BOEIE Bit (Bus-Off Entry Interrupt Enable)

When the BOEIE bit is 0, no error interrupt request is generated even if the BOEIF flag in EIFR is set to 1. When the BOEIE bit is 1, an error interrupt request is generated if the BOEIF flag is set to 1.

BORIE Bit (Bus-Off Recovery Interrupt Enable)

When the BORIE bit is 0, an error interrupt request is not generated even if the BORIF flag in EIFR is set to 1. When the BORIE bit is set to 1, an error interrupt request is generated if the BORIF flag is set to 1.

ORIE Bit (Overrun Interrupt Enable)

When the ORIE bit is 0, an error interrupt request is not generated even if the ORIF flag in EIFR is set to 1. When the ORIE bit is 1, an error interrupt request is generated if the ORIF flag is set to 1.

OLIE Bit (Overload Frame Transmit Interrupt Enable)

When the OLIE bit is 0, no error interrupt request is generated even if the OLIF flag in EIFR is set to 1. When the OLIE bit is 1, an error interrupt request is generated if the OLIF flag is set to 1.

BLIE Bit (Bus Lock Interrupt Enable)

When the BLIE bit is 0, no error interrupt request is generated even if the BLIF flag in EIFR is set to 1. When the BLIE bit is 1, an error interrupt request is generated if the BLIF flag is set to 1.

39.2.19 Error Interrupt Factor Judge Register (EIFR)

Address(es): CAN0.EIFR 0009 084Dh, CAN1.EIFR 0009 184Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	BLIF	OLIF	ORIF	BORIF	BOEIF	EPIF	EWIF	BEIF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BEIF	Bus Error Detect Flag	0: No bus error detected 1: Bus error detected	R/W
b1	EWIF	Error-Warning Detect Flag	0: No error-warning detected 1: Error-warning detected	R/W
b2	EPIF	Error-Passive Detect Flag	0: No error-passive detected 1: Error-passive detected	R/W
b3	BOEIF	Bus-Off Entry Detect Flag	0: No bus-off entry detected 1: Bus-off entry detected	R/W
b4	BORIF	Bus-Off Recovery Detect Flag	0: No bus-off recovery detected 1: Bus-off recovery detected	R/W
b5	ORIF	Receive Overrun Detect Flag	0: No receive overrun detected 1: Receive overrun detected	R/W
b6	OLIF	Overload Frame Transmission Detect Flag	0: No overload frame transmission detected 1: Overload frame transmission detected	R/W
b7	BLIF	Bus Lock Detect Flag	0: No bus lock detected 1: Bus lock detected	R/W

If an event corresponding to each bit occurs, the corresponding bit in EIFR is set to 1 regardless of the setting of EIER. To set each bit to 0, write 0 by a program. If the set timing occurs simultaneously with the clear timing by the program, the bit becomes 1.

When a single bit is set to 0 by a program, do not use the logic operation instruction (AND) – use the transfer instruction (MOV) to ensure that only the specified bit is set to 0 and the other bits are set to 1. Writing 1 has no effect to these bit values.

BEIF Flag (Bus Error Detect Flag)

The BEIF flag is set to 1 when a bus error is detected.

EWIF Flag (Error-Warning Detect Flag)

The EWIF flag is set to 1 when the value of the receive error counter (REC) or transmit error counter (TEC) exceeds 95. The EWIF flag is set to 1 only when the REC or TEC initially exceeds 95. Thus, if 0 is written to the EWIF flag by a program while the REC or TEC remains greater than 95, the EWIF flag is not set to 1 until the REC and TEC go below 95 and then REC or TEC exceeds 95 again.

EPIF Flag (Error-Passive Detect Flag)

The EPIF flag is set to 1 when the CAN error state becomes error-passive (the REC (receive error counter) or TEC (transmit error counter) value exceeds 127).
The EPIF flag is set to 1 only when the REC or TEC initially exceeds 127. Thus, if 0 is written by a program while the REC or TEC remains greater than 127, the EPIF flag is not set to 1 until the REC and TEC go below 127 and then REC or TEC exceeds 127 again.

BOEIF Flag (Bus-Off Entry Detect Flag)

The BOEIF flag is set to 1 when the CAN error state becomes bus-off (the TEC (transmit error counter) value exceeds 255). The BOEIF flag is also set to 1 when the BOM[1:0] bits in CTLR are 01b (entry to CAN halt mode automatically at bus-off entry) and the CAN module becomes the bus-off state.

BORIF Flag (Bus-Off Recovery Detect Flag)

The BORIF flag is set to 1 when the CAN module recovers from the bus-off state normally by detecting 11 consecutive bits 128 times in the following conditions:

- When the BOM[1:0] bits in CTLR are 00b
- When the BOM[1:0] bits in CTLR are 10b
- When the BOM[1:0] bits in CTLR are 11b

However, the BORIF flag is not set to 1 if the CAN module recovers from the bus-off state in the following conditions:

- When the CANM[1:0] bits in CTLR are set to 01b or 11b (CAN reset mode)
- When the RBOC bit in CTLR is set to 1 (forcible return from bus-off)
- When the BOM[1:0] bits in CTLR are set to 01b
- When the BOM[1:0] bits in CTLR are set to 11b and the CANM[1:0] bits in CTLR are set to 10b (CAN halt mode) before normal recovery occurs

Table 39.7 lists the behavior of BOEIF and BORIF flags according to the CTLR.BOM[1:0] bit setting.

Table 39.7 Behavior of BOEIF and BORIF Flags according to CTLR.BOM[1:0] Bit Setting

BOM[1:0] Bits	BOEIF Flag	BORIF Flag
00b	Set to 1 on entry to the bus-off state.	Set to 1 on exit from the bus-off state.
01b		Do not set to 1.
10b		Set to 1 on exit from the bus-off state.
11b		Set to 1 if normal bus-off recovery occurs before the CANM[1:0] bits are set to 10b (CAN halt mode).

ORIF Flag (Receive Overrun Detect Flag)

The ORIF flag is set to 1 when a receive overrun occurs. This flag is not to set to 1 in overwrite mode.
In overwrite mode, a reception complete interrupt request is generated if an overwrite condition occurs and the ORIF flag is not set to 1.
In normal mailbox mode, if an overrun occurs in any of mailboxes [0] to [31] in overrun mode, this flag is set to 1. In

FIFO mailbox mode, if an overrun occurs in any of mailboxes [0] to [23] or the receive FIFO in overrun mode, this flag is set to 1.

OLIF Flag (Overload Frame Transmission Detect Flag)

The OLIF flag is set to 1 if the transmitting condition of an overload frame is detected when the CAN module performs transmission or reception.

BLIF Flag (Bus Lock Detect Flag)

The BLIF flag is set to 1 if 32 consecutive dominant bits are detected on the CAN bus while the CAN module is in CAN operation mode.

After the BLIF flag is set to 1, a bus lock is detected again under either of the following conditions:

- After this flag is set to 0 from 1, recessive bits are detected
- After this flag is set to 0 from 1, the CAN module enters CAN reset mode or CAN halt mode and then enters CAN operation mode again (internal reset).

39.2.20 Receive Error Count Register (RECR)

Address(es): CAN0.RECR 0009 084Eh, CAN1.RECR 0009 184Eh

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

Bit	Description	R/W
b7 to b0	Receive error count function RECR increments or decrements the counter value according to the error status of the CAN module during reception.	R

RECR indicates the value of the receive error counter.

Refer to the ISO 11898-1 Standards about the increment/decrement conditions of the receive error counter.

The value of RECR in the bus-off state is undefined.

39.2.21 Transmit Error Count Register (TECR)

Address(es): CAN0.TECR 0009 084Fh, CAN1.TECR 0009 184Fh

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

Bit	Description	R/W
b7 to b0	Transmit error count function TECR increments or decrements the counter value according to the error status of the CAN module during transmission.	R

TECR indicates the value of the transmit error counter.
Refer to the ISO 11898-1 Standards about the increment/decrement conditions of the transmit error counter.
The value of TECR in the bus-off state is undefined.

39.2.22 Error Code Store Register (ECSR)

Address(es): CAN0.ECSR 0009 0850h, CAN1.ECSR 0009 1850h

	b7	b6	b5	b4	b3	b2	b1	b0
	EDPM	ADEF	BE0F	BE1F	CEF	AEF	FEF	SEF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SEF	Stuff Error Flag*1, *2	0: No stuff error detected 1: Stuff error detected	R/W
b1	FEF	Form Error Flag*1, *2	0: No form error detected 1: Form error detected	R/W
b2	AEF	ACK Error Flag*1, *2	0: No ACK error detected 1: ACK error detected	R/W
b3	CEF	CRC Error Flag*1, *2	0: No CRC error detected 1: CRC error detected	R/W
b4	BE1F	Bit Error (recessive) Flag*1, *2	0: No bit error (recessive) detected 1: Bit error (recessive) detected	R/W
b5	BE0F	Bit Error (dominant) Flag*1, *2	0: No bit error (dominant) detected 1: Bit error (dominant) detected	R/W
b6	ADEF	ACK Delimiter Error Flag*1, *2	0: No ACK delimiter error detected 1: ACK delimiter error detected	R/W
b7	EDPM	Error Display Mode Select*3, *4	0: Output of first detected error code 1: Output of accumulated error code	R/W

Note 1. Writing 1 has no effect to these flag values.
Note 2. To write 0 to SEF, FEF, AEF, CEF, BE1F, BE0F, and ADEF flags, do not use the logic operation instruction (AND). Use the transfer (MOV) instruction to ensure that only the specified flag is set to 0 and the other flags are set to 1.
Note 3. Write to the EDPM bit in CAN reset mode or CAN halt mode.
Note 4. If more than one error condition is detected simultaneously, all related flags are set to 1.

ECSR can be used to monitor whether an error has occurred on the CAN bus.
Refer to the ISO 11898-1 Standards to check the generation conditions of each error.

To set each flag except for the EDPM bit to 0, write 0 by a program. If the timing at which each flag is set to 1 and the timing at which 0 is written by a program are the same, the relevant flag is set to 1.

SEF Flag (Stuff Error Flag)

The SEF flag is set to 1 when a stuff error is detected.

FEF Flag (Form Error Flag)

The FEF flag is set to 1 when a form error is detected.

AEF Flag (ACK Error Flag)

The AEF flag is set to 1 when an ACK error is detected.

CEF Flag (CRC Error Flag)

The CEF flag is set to 1 when a CRC error is detected.

BE1F Flag (Bit Error (recessive) Flag)

The BE1F flag is set to 1 when a recessive bit error is detected.

BE0F Flag (Bit Error (dominant) Flag)

The BE0F flag is set to 1 when a dominant bit error is detected.

ADEF Flag (ACK Delimiter Error Flag)

The ADEF flag is set to 1 when a form error is detected with the ACK delimiter during transmission.

EDPM Bit (Error Display Mode Select)

The EDPM bit selects the output mode of ECSR. When the EDPM bit is set to 0, ECSR outputs the first error code. When the EDPM bit is set to 1, ECSR outputs the accumulated error code.

39.2.23 Time Stamp Register (TSR)

Address(es): CAN0.TSR 0009 0854h, CAN1.TSR 0009 1854h

	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.

39.2.24 Test Control Register (TCR)

Address(es): CAN0.TCR 0009 0858h, CAN1.TCR 0009 1858h

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

Bit	Symbol	Bit Name	Description	R/W
b0	TSTE	CAN Test Mode Enable	0: CAN test mode disabled 1: CAN test mode enabled	R/W
b2, b1	TSTM[1:0]	CAN Test Mode Select	b2 b1 0 0: Other than CAN test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (external loopback) 1 1: Self-test mode 1 (internal loopback)	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TCR controls the CAN test mode. Write to TCR in CAN halt mode only.

(1) Listen-Only Mode

The ISO 11898-1 Standards recommend an optional bus monitoring mode. In listen-only mode, valid data frames and valid remote frames can be received. However, only recessive bits can be sent on the CAN bus, and the ACK bit, overload flag, and active error flag cannot be sent.

Listen-only mode can be used for baud rate detection.

Do not request transmission from any mailboxes in listen-only mode.

Figure 39.6 shows the connection when listen-only mode is selected (i = 0, 1).

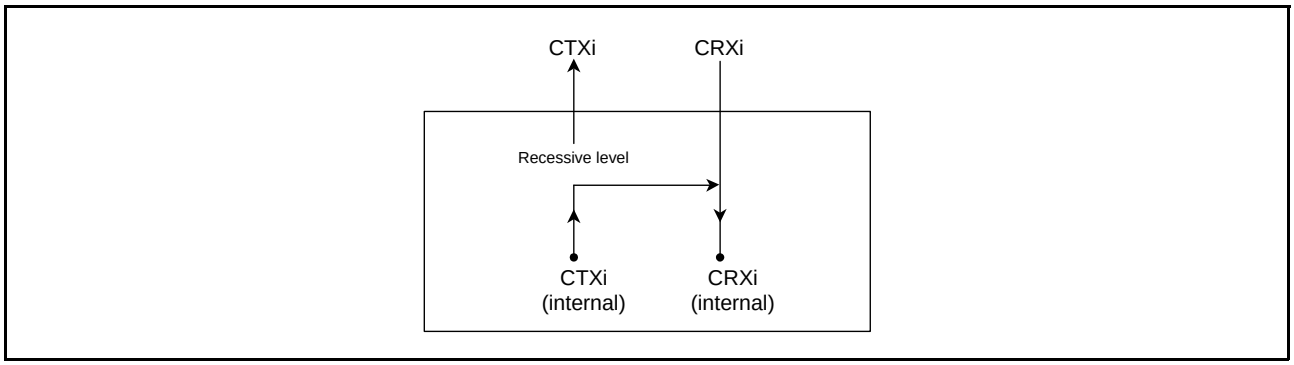


Figure 39.6 Connection when Listen-Only Mode is Selected (i = 0, 1)

(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 39.7 shows the connection when self-test mode 0 is selected (i = 0, 1).

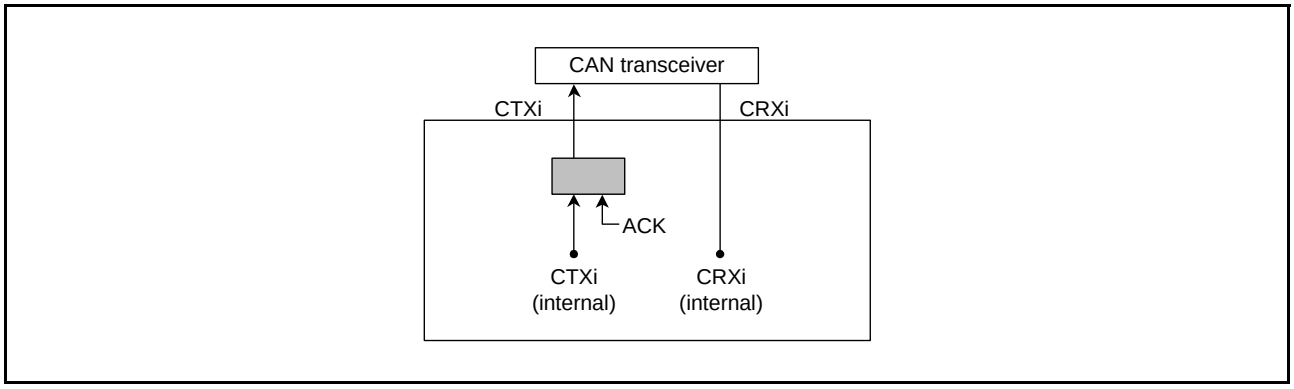


Figure 39.7 Connection when Self-Test Mode 0 is Selected (i = 0, 1)

(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 39.8 shows the connection when self-test mode 1 is selected (i = 0, 1).

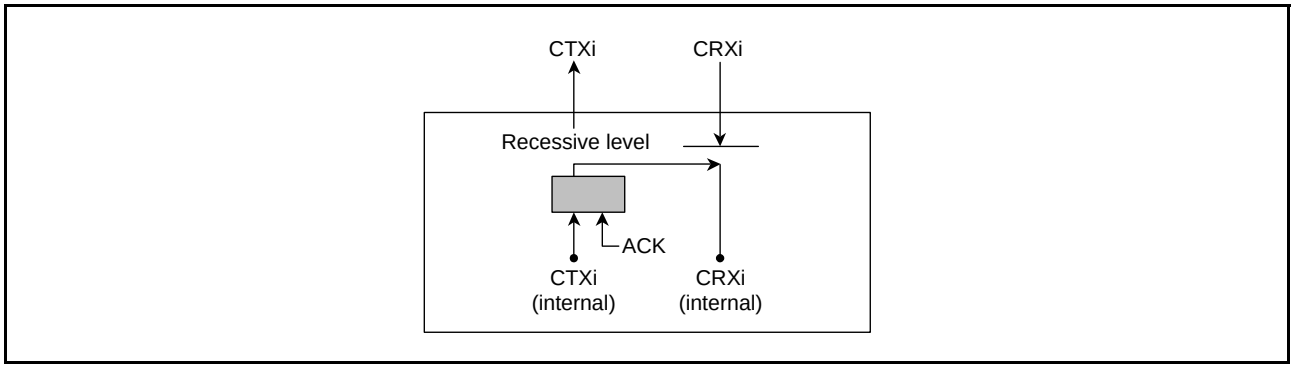


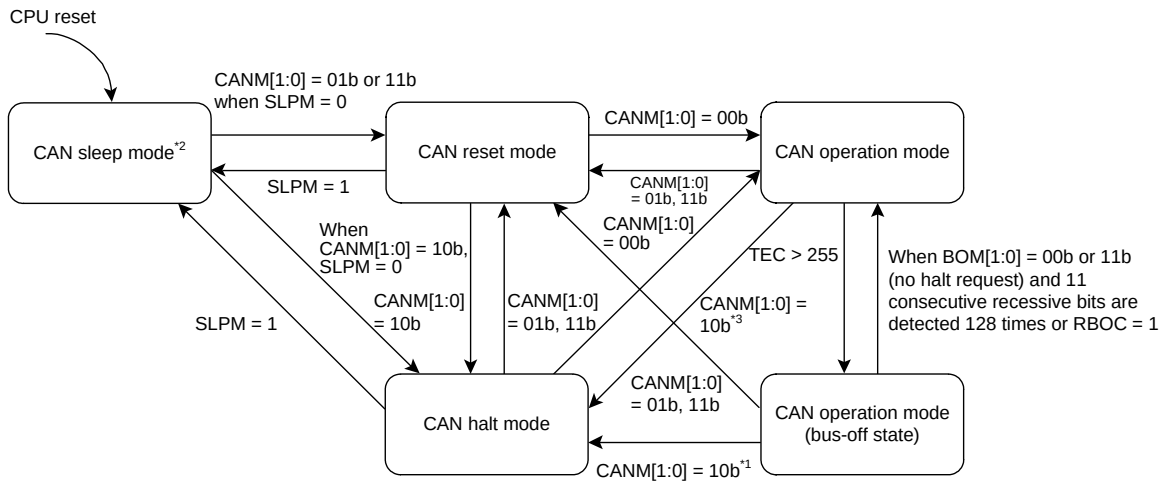
Figure 39.8 Connection when Self-Test Mode 1 is Selected (i = 0, 1)

39.3 Operating Mode

The CAN module has the following four operating modes.

- CAN reset mode
- CAN halt mode
- CAN operation mode
- CAN sleep mode

Figure 39.9 shows the transition between CAN operating modes.



CANM, SLPM, BOM, RBOC: Bits in CTLR

Note 1. The transition timing from the bus-off state to CAN halt mode depends on the setting of the CTLR.BOM[1:0] bits.
When the CTLR.BOM[1:0] bits are 01b, the state transition timing is immediately after entering the bus-off state.
When the CTLR.BOM[1:0] bits are 10b, the state transition timing is at the end of the bus-off state.
When the CTLR.BOM[1:0] bits are 11b, the state transition timing is at the setting of the CTLR.CANM[1:0] bits to 10b (CAN halt mode).
Note 2. Change the CTLR.SLPM bit to set or cancel CAN sleep mode.
Note 3. While the CAN bus is locked at the dominant level, the CAN module does not enter CAN halt mode. In that case, enter CAN reset mode.

Figure 39.9 Transition between CAN Operating Modes

39.3.1 CAN Reset Mode

CAN reset mode is provided for CAN communication configuration.
When the CTLR.CANM[1:0] bits are set to 01b or 11b, the CAN module enters CAN reset mode. Then, the STR.RSTST flag is set to 1. Do not change the CTLR.CANM[1:0] bits until the RSTST flag is set to 1. Set BCR before exiting CAN reset mode to any other modes.
The following registers are initialized to their reset values after entering CAN reset mode, and their initial values are retained during CAN reset mode:

- MCTLj
- STR (except for the SLPST and TFST flags)
- EIFR
- RECR
- TECR
- TSR
- MSSR
- MSMR
- RFCR
- TFCR
- TCR
- ECSR (except for the EDPM bit)

The following registers retain their previous values even after entering CAN reset mode.

- CTLR
- STR (only the SLPST and TFST flags)
- MIER
- EIER
- BCR
- CSSR
- ECSR (only the EDPM bit)
- MBj
- MKR0 to MKR7
- FIDCR0 and FIDCR1
- MKIVLR
- AFSR
- RFPCR
- TFPCR

39.3.2 CAN Halt Mode

CAN halt mode is used for mailbox configuration and test mode setting.

When the CTLR.CANM[1:0] bits are set to 10b, CAN halt mode is selected. Then the STR.HLTST flag is set to 1. Do not change the CTLR.CANM[1:0] bits until the HLTST flag is set to 1.

See Table 39.8 for the state transition conditions when transmitting or receiving.

All registers except for RSTST, HLTST, and SLPST flags in STR remain unchanged when the CAN enters CAN halt mode.

Do not change CTLR (except for bits CANM[1:0] and SLPM) and EIER in CAN halt mode. BCR can be changed in CAN halt mode only when listen-only mode is selected for automatic baud rate detection.

Table 39.8 Operation in CAN Reset Mode and CAN Halt Mode

Mode	Receiver	Transmitter	Bus-Off
CAN reset mode (forcible transition) CANM[1:0] = 11b	CAN module enters CAN reset mode without waiting for the end of message reception.	CAN module enters CAN reset mode without waiting for the end of message transmission.	CAN module enters CAN reset mode without waiting for the end of bus-off recovery.
CAN reset mode CANM[1:0] = 01b	CAN module enters CAN reset mode without waiting for the end of message reception.	CAN module enters CAN reset mode after waiting for the end of message transmission.*1, *4	CAN module enters CAN reset mode without waiting for the end of bus-off recovery.
CAN halt mode	CAN module enters CAN halt mode after waiting for the end of message reception.*2, *3	CAN module enters CAN halt mode after waiting for the end of message transmission.*1, *2, *4	[When the BOM[1:0] bits are 00b] A halt request from a program will be accepted only after bus-off recovery. [When the BOM[1:0] bits are 01b] CAN module automatically enters CAN halt mode without waiting for the end of bus-off recovery (regardless of a halt request from a program). [When the BOM[1:0] bits are 10b] CAN module automatically enters CAN halt mode after waiting for the end of bus-off recovery (regardless of a halt request from a program). [When the BOM[1:0] bits are 11b] CAN module enters CAN halt mode (without waiting for the end of bus-off recovery) if a halt is requested by a program during bus-off.

CANM[1:0], BOM[1:0]: Bits in CTLR

Note 1. If several messages are requested to be transmitted, mode transition occurs after the completion of the first message transmission. In a case that the CAN reset mode is being requested during suspend transmission, mode transition occurs when the bus is idle, the next transmission ends, or the CAN module becomes a receiver.

Note 2. If the CAN bus is locked at the dominant level, the program can detect this state by monitoring the EIFR.BLIF flag. While the CAN bus is locked at the dominant level, the CAN module does not enter CAN halt mode. In that case, enter CAN reset mode.

Note 3. If a CAN bus error occurs during reception after CAN halt mode is requested, the CAN module transits to CAN halt mode. However, while the CAN bus is locked at the dominant level, the CAN module does not enter CAN halt mode.

Note 4. If a CAN bus error or arbitration-lost occurs during transmission after CAN reset mode or CAN halt mode is requested, the CAN module transits to the requested operating mode. However, while the CAN bus is locked at the dominant level, the CAN module does not enter CAN halt mode.

39.3.3 CAN Sleep Mode

CAN sleep mode is used for reducing current consumption by stopping the clock supply to the CAN module. After a reset from an MCU pin or software reset, the CAN module starts from CAN sleep mode.
When the SLPM bit in CTLR is set to 1, the CAN module enters CAN sleep mode. Then, the SLPST flag in STR is set to 1. Do not change the value of the SLPM bit until the SLPST flag is set to 1. The other registers remain unchanged when the CAN module enters CAN sleep mode.
Write to the SLPM bit in CAN reset mode and CAN halt mode. Do not change any registers (except for the SLPM bit) during CAN sleep mode. Read operation is still allowed.
When the SLPM bit is set to 0, the CAN module is released from CAN sleep mode. When the CAN module exits CAN sleep mode, the other registers remain unchanged.

39.3.4 CAN Operation Mode (Excluding Bus-Off State)

CAN operation mode is used for CAN communication.
When the CANM[1:0] bits in CTLR are set to 00b, the CAN module enters CAN operation mode.
Then RSTST and HLTST flags in STR are set to 0. Do not change the value of the CANM[1:0] until bits RSTST and HLTST flags are set to 0.
If 11 consecutive recessive bits are detected after entering CAN operation mode, the CAN module is in the following states:

- The CAN module becomes an active node on the network, thus enabling transmission and reception of CAN messages.
- Error monitoring of the CAN bus, such as receive and transmit error counters, is performed.

During CAN operation mode, the CAN module may be in one of the following three sub-modes, depending on the status of the CAN bus.

- Idle mode: Transmission or reception is not being performed.
- Receive mode: A CAN message sent by another node is being received.
- Transmit mode:A CAN message is being transmitted. The CAN module receives a message transmitted by the local node simultaneously when self-test mode 0 (TSTM[1:0] bits in TCR = 10b) or self-test mode 1 (TSTM[1:0] bits = 11b) is selected.

Figure 39.10 shows the sub-modes of CAN operation mode.

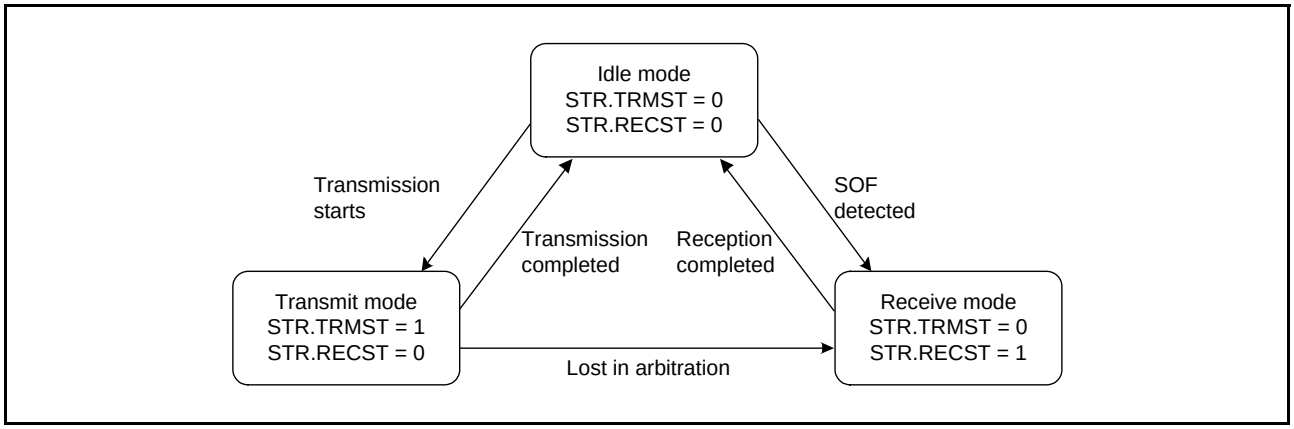


Figure 39.10 Sub-Modes of CAN Operation Mode

39.3.5 CAN Operation Mode (Bus-Off State)

The CAN module enters the bus-off state according to the increment/decrement rules for the transmit/error counters in the CAN Specifications.
The following cases apply when the CAN module is recovering from the bus-off state. When the CAN module is in the bus-off state, the values of the CAN-related registers, except for STR, EIFR, RECR, TECR and TSR, remain unchanged.

(1) When bits BOM[1:0] in CTLR are 00b (normal mode)

The CAN module enters the error-active state after it has completed the recovery from the bus-off state and CAN communication is enabled. The BORIF flag in EIFR is set to 1 (bus-off recovery detected) at this time.

(2) When bit RBOC in CTLR is set to 1 (forcible return from bus-off)

The CAN module enters the error-active state when it is in the bus-off state and the RBOC bit is set to 1. CAN communication is enabled again after 11b consecutive recessive bits are detected. The BORIF flag is not set to 1 at this time.

(3) When bits BOM[1:0] are 01b (automatic transition to CAN halt mode at bus-off entry)

The CAN module enters CAN halt mode when it reaches the bus-off state. The BORIF flag is not set to 1 at this time.

(4) When bits BOM[1:0] are 10b (automatic transition to CAN halt mode at bus-off end)

The CAN module enters CAN halt mode when it has completed the recovery from bus-off. The BORIF flag is set to 1 at this time.

(5) When bits BOM[1:0] are 11b (automatic transition to CAN halt mode by a program) and bits CANM[1:0] in CTLR are set to 10b (CAN halt mode) during bus-off state

The CAN module enters CAN halt mode when it is in the bus-off state and the CANM[1:0] bits are set to 10b (CAN halt mode). The BORIF flag is not set to 1 at this time.
If the CANM[1:0] bits are not set to 10b during bus-off, the same behavior as (1) applies.

39.4 CAN Communication Speed Setting

The following description explains about CAN communication speed setting.

39.4.1 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 39.11 shows a block diagram of the CAN clock generator.

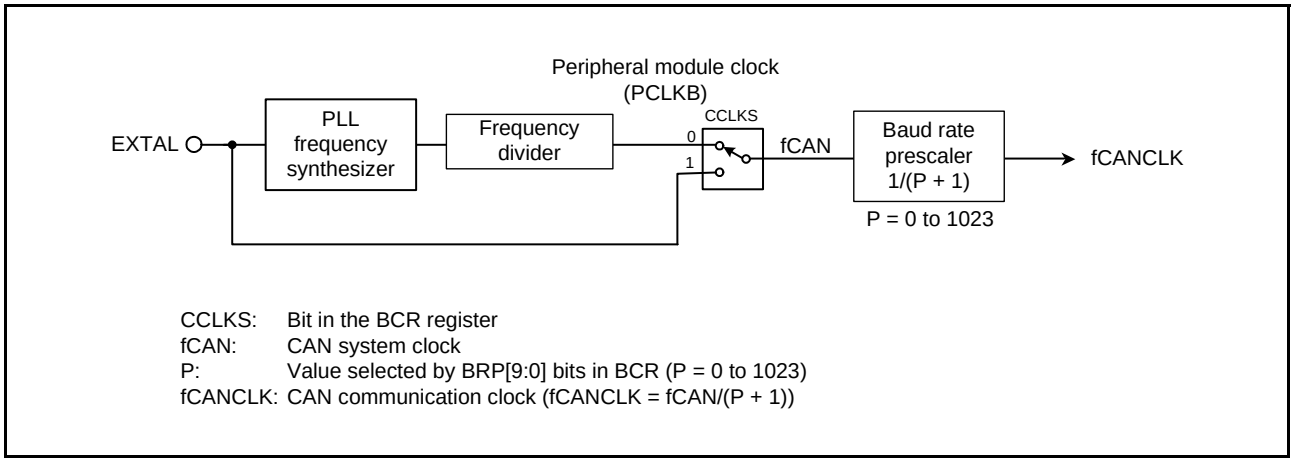


Figure 39.11 Block Diagram of CAN Clock Generator

39.4.2 Bit Timing Setting

The bit time consists of the following three segments.
Figure 39.12 shows the bit timing.

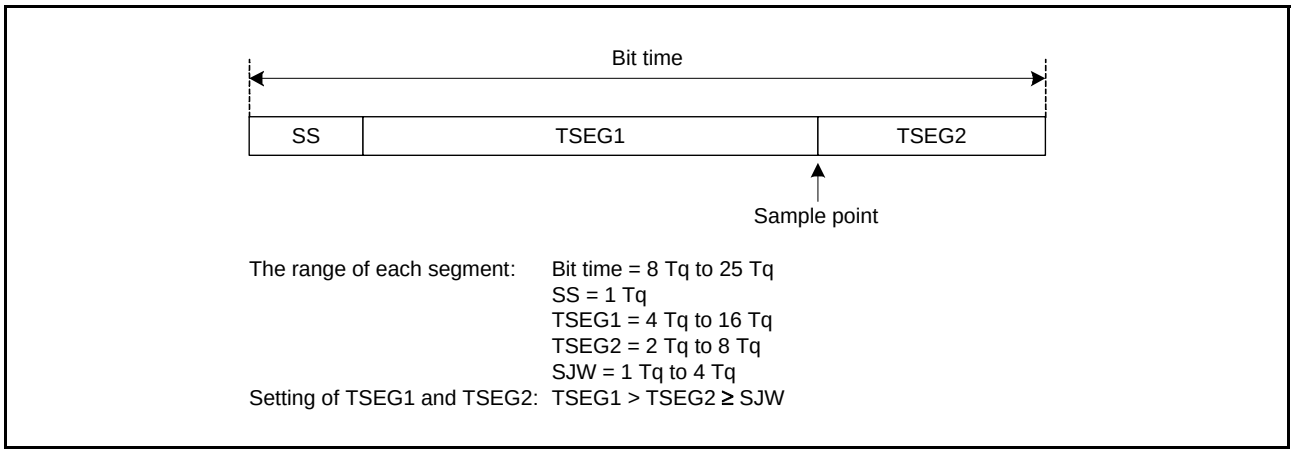


Figure 39.12 Bit Timing

39.4.3 Bit Rate

The bit rate depends on the division value of fCAN (CAN clock), the division value of the baud rate prescaler, and the number of Tq of 1 bit time.

$$\text{Bit rate [bps]} = \frac{\text{fCAN}}{\text{Baud rate prescaler division value}^{*1} \times \text{number of Tq of 1 bit time}} = \frac{\text{fCANCLK}}{\text{Number of Tq of 1 bit time}}$$

Note 1. Division value of baud rate prescaler = P + 1 (P: 0 to 1023)
P: Setting of the BRP[9:0] bits in BCR

Table 39.9 lists bit rate examples.

Table 39.9 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

39.5 Mailbox and Mask Register Structure

Figure 39.13 shows the structure of MBj.
There are 32 mailboxes with the same structure.

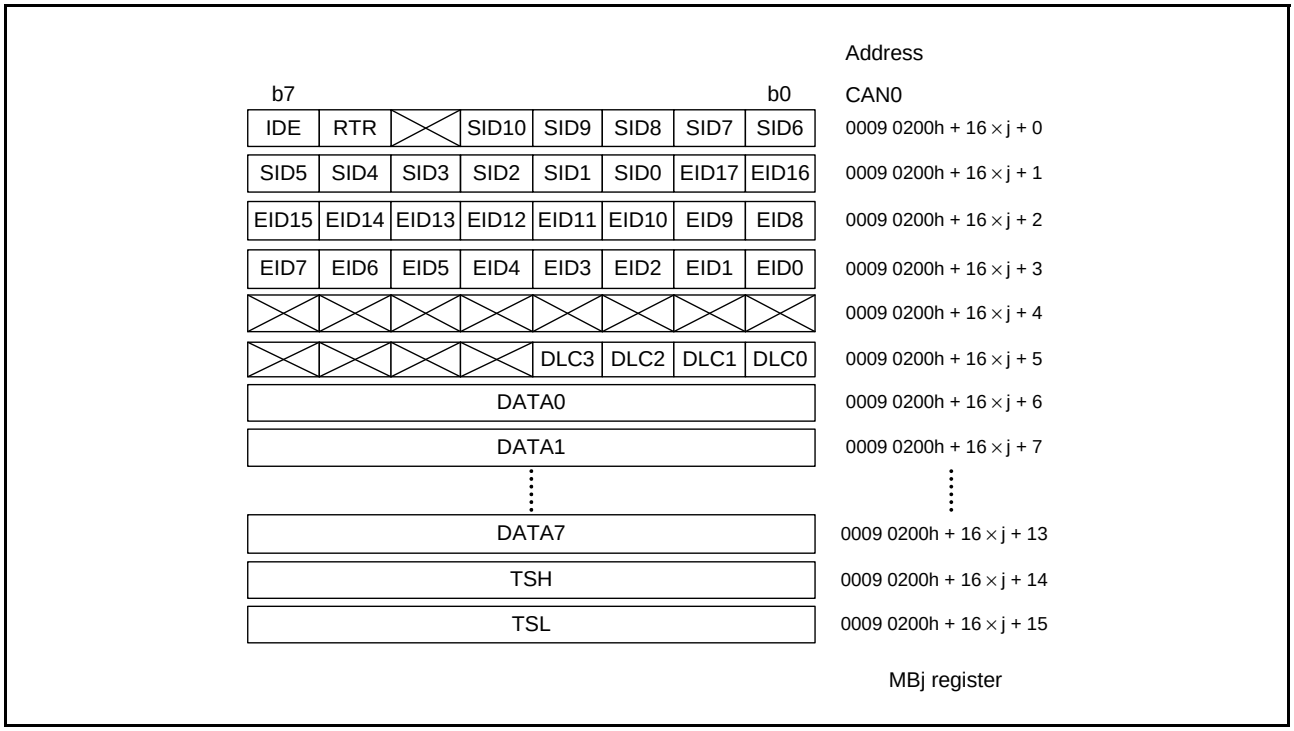


Figure 39.13 Structure of MBj (j = 0 to 31)

Figure 39.14 shows the structure of MKRk.
There are eight mask registers with the same structure.

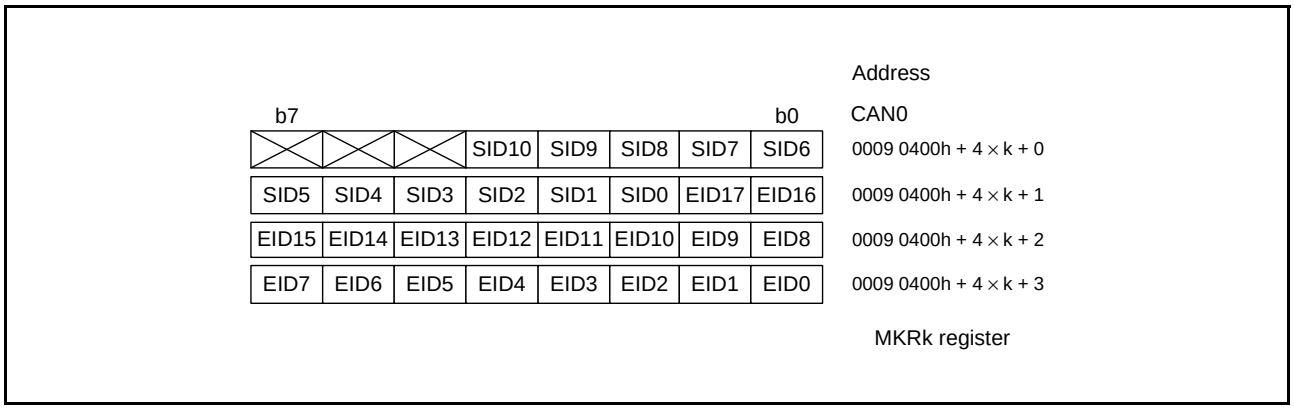


Figure 39.14 Structure of MKRk (k = 0 to 7)

Figure 39.15 shows the structure of FIDCR0 and FIDCR1.
There are two FIFO received ID compare registers with the same structure.

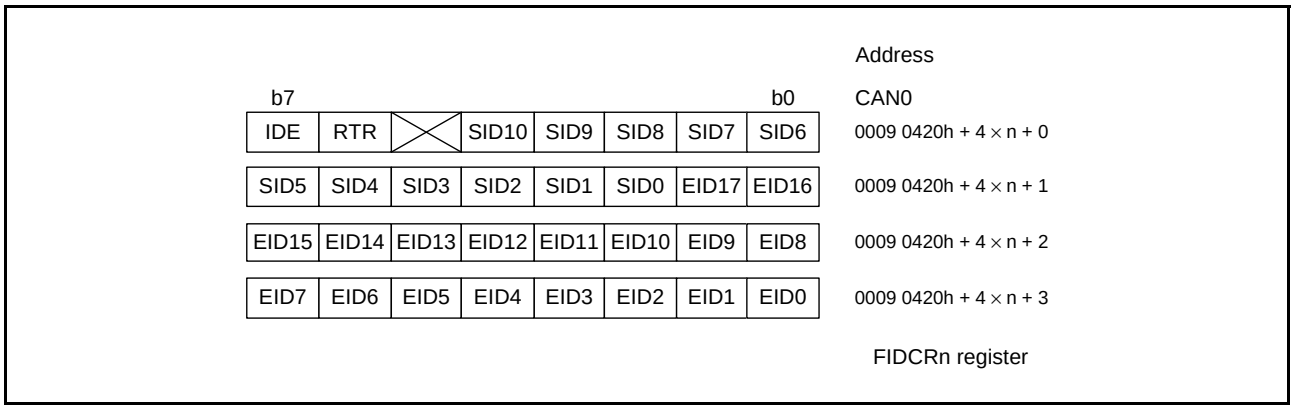


Figure 39.15 Structure of FIDCRn (n = 0, 1)

39.6 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 39.16 shows the correspondence between mask registers and mailboxes. Figure 39.17 shows acceptance filtering.

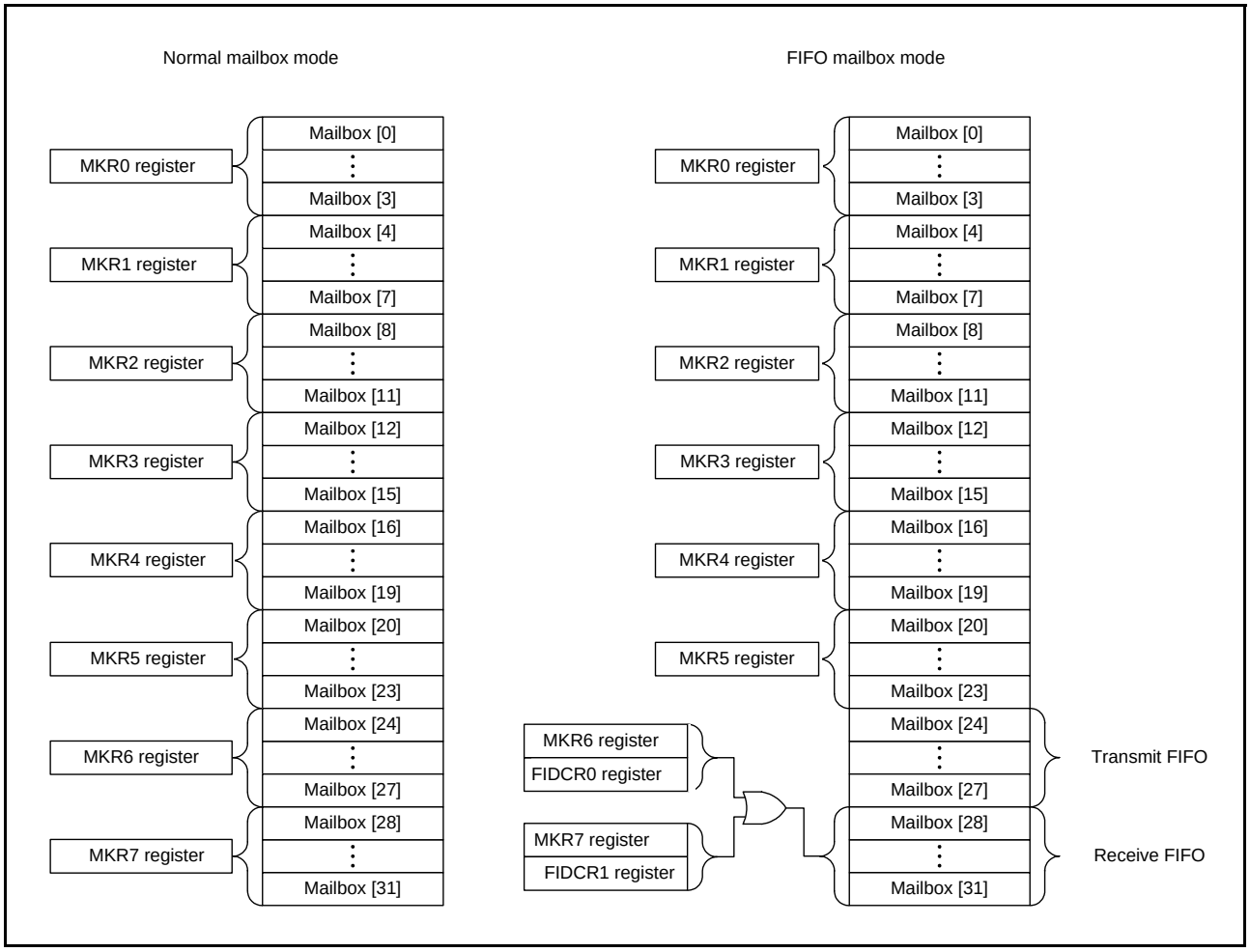
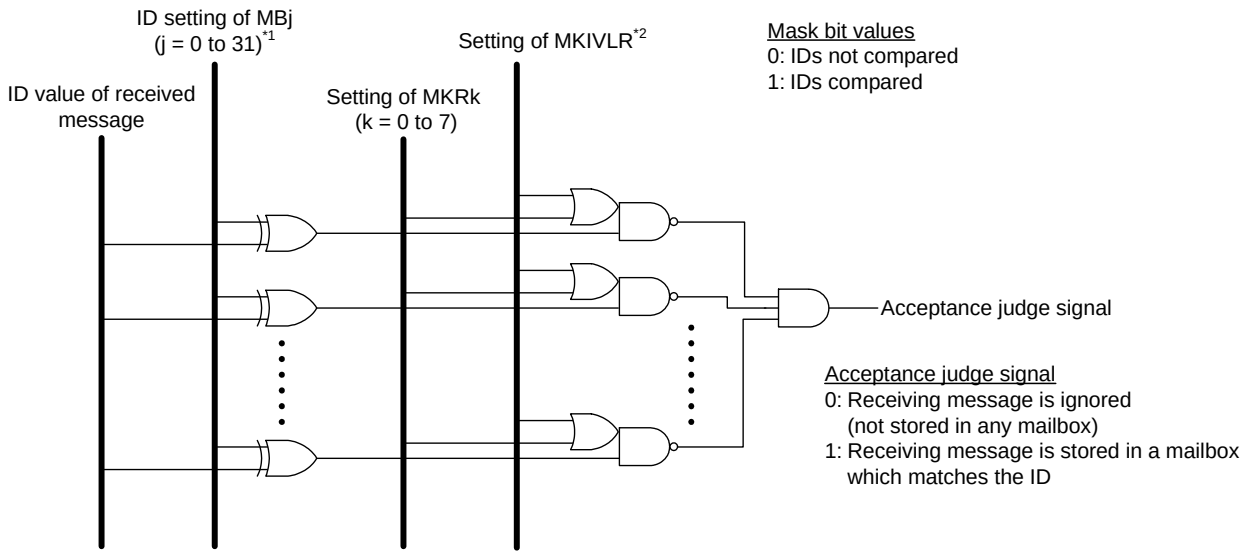


Figure 39.16 Correspondence between Mask Registers and Mailboxes



Note 1. The settings of FIDCR0 and FIDCR1 are used in FIFO mailbox mode.
Note 2. Invalid in FIFO mailboxes.

Figure 39.17 Acceptance Filtering

39.7 Reception and Transmission

Table 39.10 lists how to make the CAN communication mode settings.

Table 39.10 Setting of CAN Receive Mode and CAN Transmit Mode

MCTLj. TRMREQ	MCTLj. RECREQ	MCTLj. ONESHOT	Communication Mode of Mailbox
0	0	0	Mailbox disabled or transmission being aborted.
0	0	1	Can be configured only when transmission or reception from a mailbox programmed in one-shot mode is aborted.
0	1	0	Configured as a receive mailbox for a data frame or a remote frame.
0	1	1	Configured as a one-shot receive mailbox for a data frame or a remote frame.
1	0	0	Configured as a transmit mailbox for a data frame or a remote frame.
1	0	1	Configured as a one-shot transmit mailbox for a data frame or a remote frame.
1	1	0	Do not set.
1	1	1	Do not set.

j = 0 to 31

When a mailbox is configured as a receive mailbox or a one-shot receive mailbox, note the following:

1. Before a mailbox is configured as a receive mailbox or a one-shot receive mailbox, set MCTLj to 00h.
2. A received message is stored into the first mailbox that matches the condition according to the result of receive mode setting and acceptance filtering. Upon deciding the mailbox to store the received message, the mailbox with the smaller number has higher priority.
3. In CAN operation mode, when the CAN module transmits a message whose ID matches with the ID/mask set of a mailbox configured to receive messages, the CAN module never receives the transmitted data. In self-test mode, however, the CAN module will receive its transmitted data. In this case, the CAN module returns ACK.

When configuring a mailbox as a transmit mailbox or a one-shot transmit mailbox, note the following:

4. Before a mailbox is configured as a transmit mailbox or a one-shot transmit mailbox, ensure that MCTLj is 00h and that there is no pending abort process.

39.7.1 Reception

Figure 39.18 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).

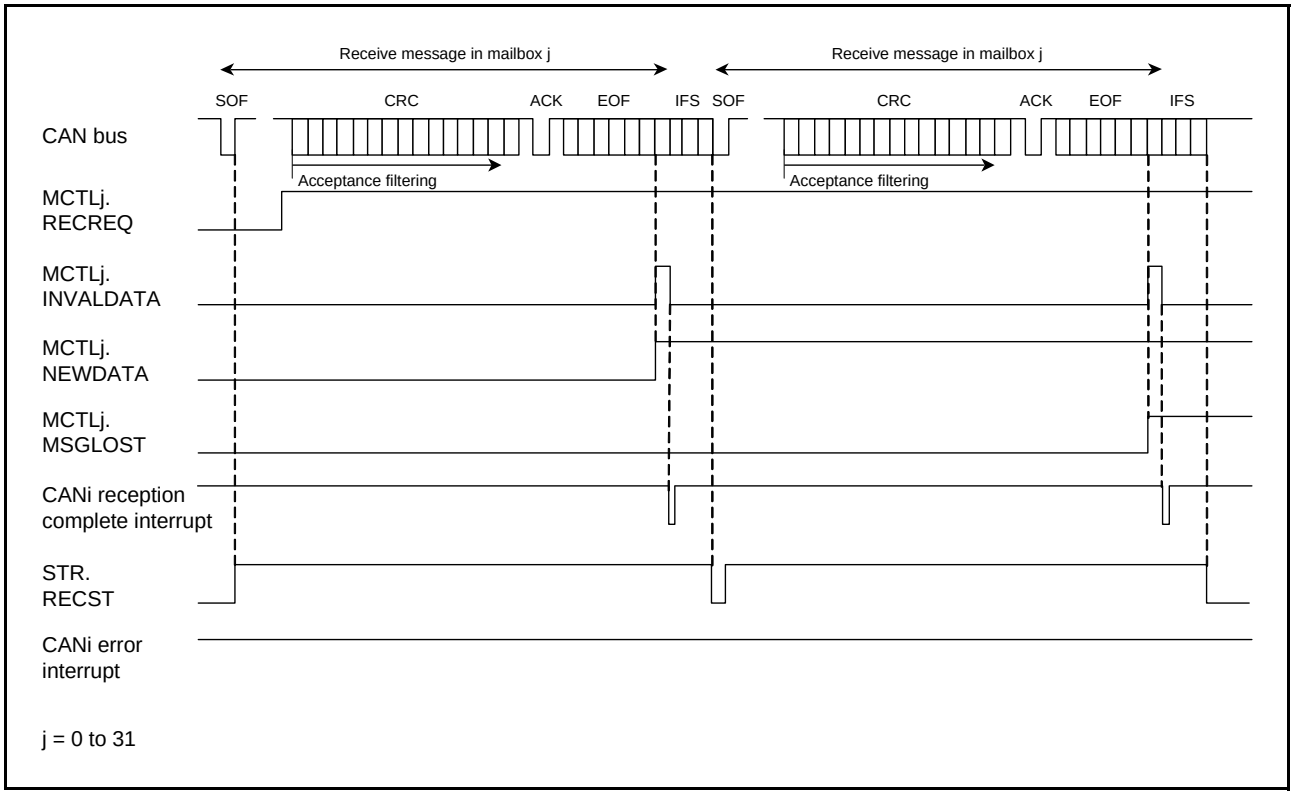


Figure 39.18 Operation Example of Data Frame Reception in Overwrite Mode

1. When an SOF is detected on the CAN bus, the RECST flag in STR is set to 1 (reception in progress) if the CAN module has no message ready to start transmission.
2. The acceptance filter processing starts at the beginning of the CRC field to select the receive mailbox.
3. After a message has been received, the NEWDATA flag in MCTLj for the receive mailbox is set to 1 (new message is being stored or has been stored to the mailbox). The INVALDATA flag in MCTLj is set to 1 (message is being updated) at the same time, and then the INVALDATA flag is set to 0 (message valid) again after the complete message is transferred to the mailbox.
4. When the interrupt enable bit in MIER for the receive mailbox is 1 (interrupt enabled), the CANi reception complete interrupt request is generated. This interrupt (CANi reception complete interrupt) is generated when the INVALDATA flag is set to 0.
5. After reading the message from the mailbox, the NEWDATA flag needs to be set to 0 by a program.
6. In overwrite mode, if the next CAN message has been received into a mailbox whose NEWDATA flag is still set to 1, the MSGLOST flag in MCTLj is set to 1 (message has been overwritten). The new received message is transferred to the mailbox. The CANi reception complete interrupt request is generated the same as in 4.

Figure 39.19 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).

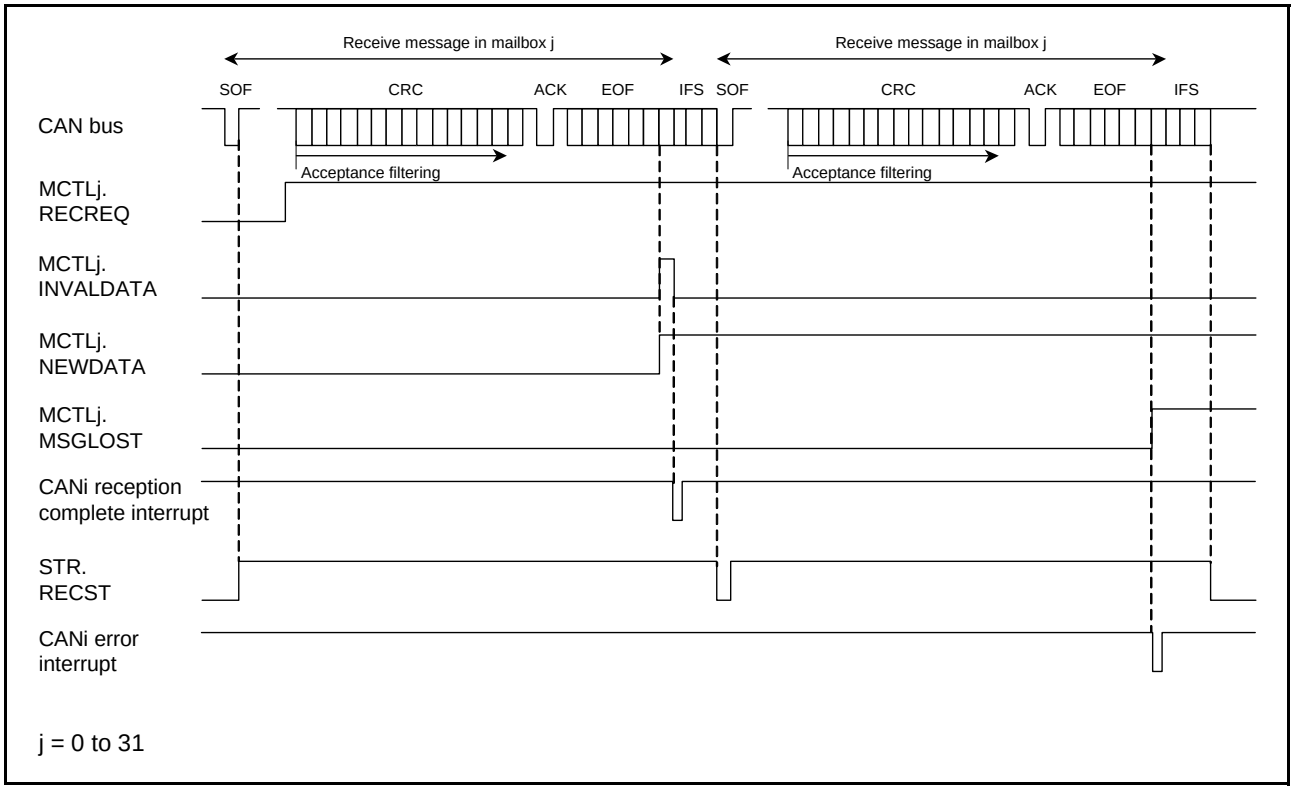


Figure 39.19 Operation Example of Data Frame Reception in Overrun Mode

1. to 5. are the same as in overwrite mode.

6. In overrun mode, if the next CAN message has been received before the NEWDATA flag in MCTLj is set to 0, the MSGLOST flag in MCTLj is set to 1 (message has been overrun). The new received message is discarded and a CANi error interrupt request is generated if the corresponding interrupt enable bit in EIER is set to 1 (interrupt enabled).

39.7.2 Transmission

Figure 39.20 shows an operation example of data frame transmission.

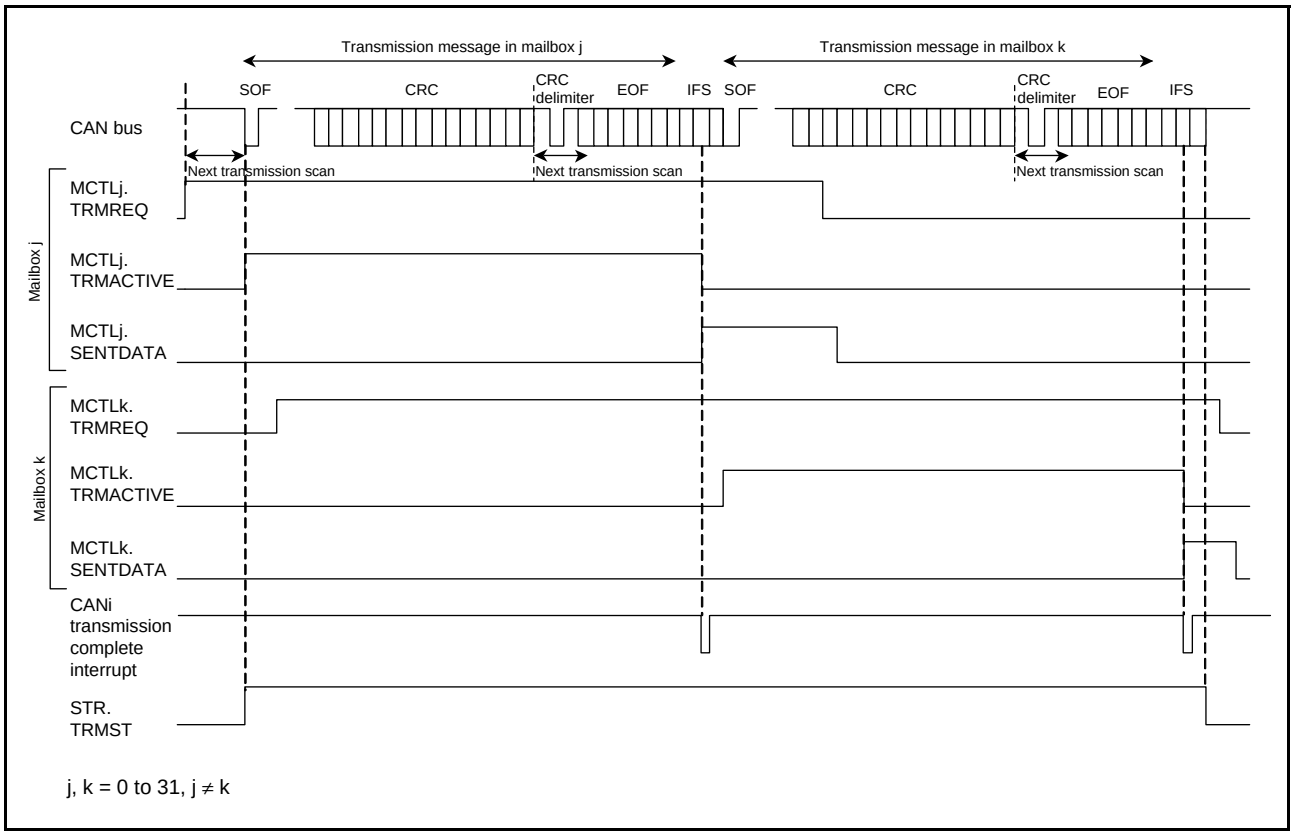


Figure 39.20 Operation Example of Data Frame Transmission

1. When a TRMREQ bit in MCTLj (j = 0 to 31) is set to 1 (transmit mailbox) in the bus-idle state, the mailbox scan processing starts to decide the highest-priority mailbox for transmission. Once the transmit mailbox is decided, the TRMACTIVE flag in MCTLj is set to 1 (from acceptance of transmission request to completion of transmission, or error/arbitration-lost), the TRMST flag in STR is set to 1 (transmission in progress), and the CAN module starts transmission.*1
2. If other TRMREQ bits are set, the transmission scan processing starts with the CRC delimiter for the next transmission.
3. If transmission is completed without losing arbitration, the SENTDATA flag in MCTLj is set to 1 (transmission completed) and the TRMACTIVE flag is set to 0 (transmission is pending or transmission is not requested). If the interrupt enable bit in MIER is 1 (interrupt enabled), the CANi transmission complete interrupt request is generated.
4. When requesting the next transmission from the same mailbox, set SENTDATA flag and TRMREQ bit to 0, then set the TRMREQ bit to 1 after checking that SENTDATA flag and TRMREQ bit have been set to 0.

Note 1. If arbitration is lost after the CAN module starts transmission, the TRMACTIVE flag is set to 0. The transmission scan processing is performed again to search for the highest-priority transmit mailbox from the beginning of the CRC delimiter. If an error occurs either during transmission or following the arbitration-lost, the transmission scan processing is performed again to search for the highest-priority transmit mailbox from the start of the error delimiter.

39.8 CAN Interrupt

The CAN module provides the following CAN interrupts for each channel. Table 39.11 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 39.11 CAN Interrupts

Module	Interrupt Symbol	Interrupt Source	Source Flag
CANi	ERSi	Bus lock detected	EIFR.BLIF
		Overload frame transmission detected	EIFR.OLIF
		Overrun detected	EIFR.ORIF
		Bus-off recovery detected	EIFR.BORIF
		Bus-off entry detected	EIFR.BOEIF
		Error-passive detected	EIFR.EPIF
		Error-warning detected	EIFR.EWIF
		Bus error detected	EIFR.BEIF
	RXFi	Receive FIFO message received (MIER[29] = 0)	RFCR.RFUST[2:0]
		Receive FIFO warning (MIER[29] = 1)	
	TXFi	Transmit FIFO message transmission completed (MIER[25] = 0)	TFCR.TFUST[2:0]
		FIFO last message transmission completed (MIER[25] = 1)	
	RXMi	Mailbox 0 to 31 message received	MCTL0.NEWDATA to MCTL31.NEWDATA
	TXMi	Mailbox 0 to 31 message transmission completed	MCTL0.SENTDATA to MCTL31.SENTDATA

i = 0, 1

39.9 Usage Notes

39.9.1 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.

40. Serial Peripheral Interface (RSPId)

In this section, “PCLK” is used to refer to PCLKA.

40.1 Overview

This MCU includes three channels of Serial Peripheral Interface (RSPI).
The RSPI channels are capable of high-speed, full-duplex or simplex synchronous serial communications with multiple processors and peripheral devices.
Table 40.1 lists the specifications of the RSPI, and Figure 40.1 shows a block diagram of the RSPI.
In this section, x 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.

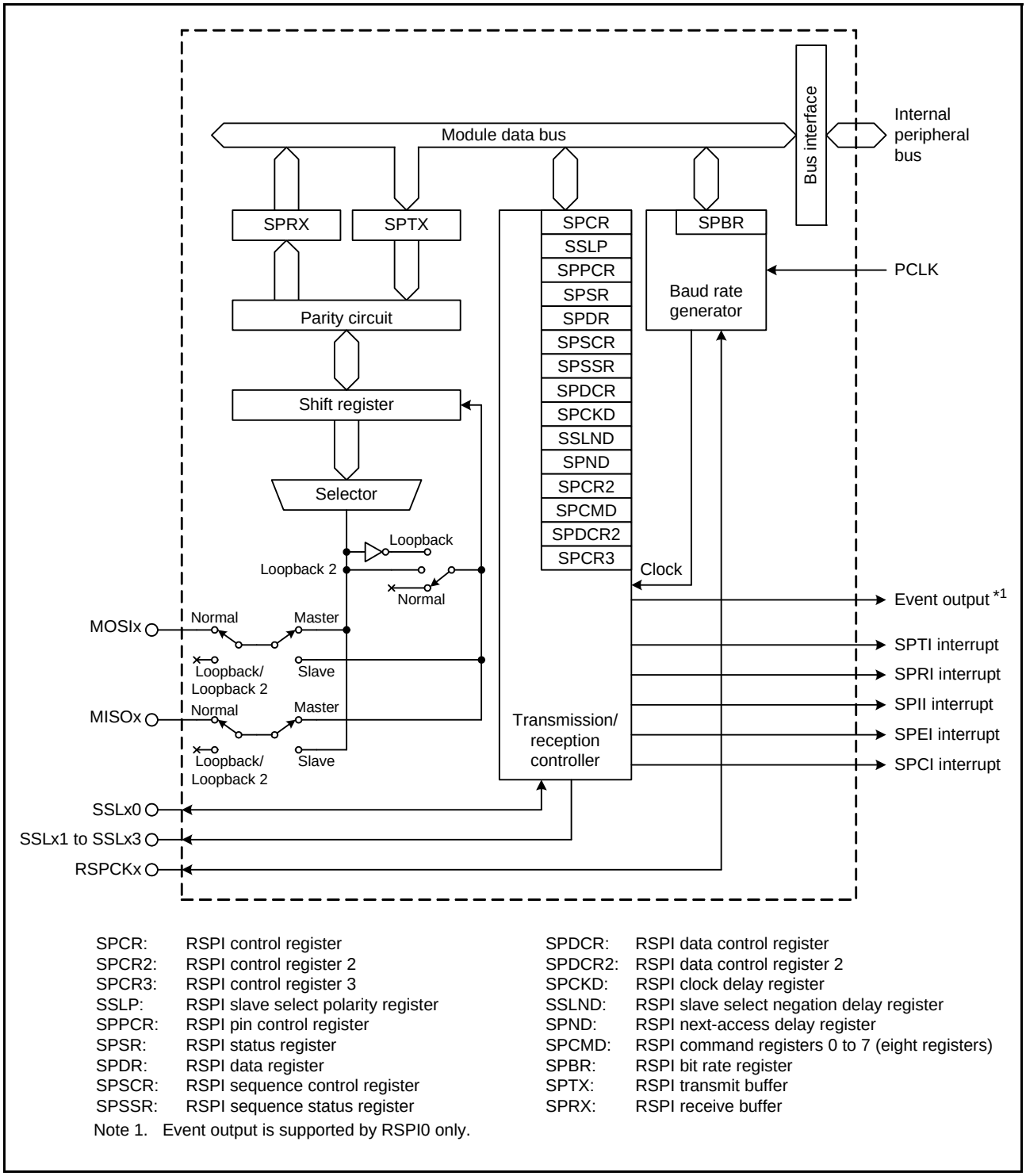
Table 40.1 RSPI Specifications (1/2)

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 (4-wire method) or clock synchronous operation (3-wire method). • Communication mode: Full-duplex or simplex (transmit-only or receive-only (in slave mode)) can be selected. • Switching of the polarity of RSPCK • Switching of the phase of RSPCK
Data format	• MSB first/LSB first selectable • Transfer bit length is selectable as 8, 9, 10, 11, 12, 13, 14, 15, 16, 20, 24, or 32 bits. • 128-bit transmit/receive buffers • Up to four frames can be transferred in one round of transmission/reception (each frame consisting of up to 32 bits). • Byte swapping of transmit and receive data is selectable. • Logic level of transmit and receive data can be inverted.
Bit rate	• In master mode, the on-chip baud rate generator generates RSPCK by frequency-dividing PCLK (the division ratio ranges from divided by 2 to divided by 4096). • In slave mode, the minimum PCLK clock divided by 4 can be input as RSPCK (the maximum frequency of RSPCK is that of PCLK divided by 4). Width at high level: 2 cycles of PCLK; width at low level: 2 cycles of PCLK
Buffer configuration	• Double buffer configuration for the transmit/receive buffers • 128 bits for the transmit/receive buffers
Error detection	• Mode fault error detection • Overrun error detection*1 • Parity error detection • Underrun error detection
SSL control function	• Four SSL pins (SSLx0 to SSLx3) for each channel • In single-master mode, SSLx0 to SSLx3 pins are output. • In multi-master mode: SSLx0 pin for input, and SSLx1 to SSLx3 pins for either output or unused. • In slave mode: SSLx0 pin for input, and SSLx1 to SSLx3 pins for unused. • Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable delay from RSPCK stop to SSL output negation (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) • Function for changing SSL polarity

Table 40.1 RSPI Specifications (2/2)

Item	Description
Control in master transfer	• A transfer of up to eight commands can be executed sequentially in looped execution. • For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity/phase, transfer data length, MSB/LSB first, burst, RSPCK delay, SSL negation delay, and next-access delay • A transfer can be initiated by writing to the transmit buffer. • MOSI signal value specifiable in SSL negation • RSPCK auto-stop function • Delay between data bytes under burst transfer can be reduced.
Interrupt sources	• Interrupt sources Receive buffer full interrupt Transmit buffer empty interrupt Error interrupt (mode fault, overrun, underrun, or parity error) Idle interrupt Communication end interrupt
Event link function (output)	• The following events can be output to the event link controller. (RSPI0) Receive buffer full event Transmit buffer empty event Error event (mode fault, overrun, underrun, or parity error) Idle event Communication end event
Others	• Function for initializing the RSPI • Loopback mode
Low power consumption function	Module stop state can be set.

Note 1. In master reception and when the RSPCK auto-stop function is enabled, an overrun error does not occur because the transfer clock is stopped at the timing of overrun error detection.



SPCR: RSPI control register
SPCR2: RSPI control register 2
SPCR3: RSPI control register 3
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
SPDCR2: RSPI data control register 2
SPCKD: RSPI clock delay register
SSLND: RSPI slave select negation 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

Note 1. Event output is supported by RSPI0 only.

Figure 40.1 RSPI Block Diagram

Table 40.2 lists the I/O pins used in the RSPI.
The RSPI automatically switches the I/O direction of the SSLx0 pin. SSLx0 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 RSPCKx, MOSIx, and MISOx are automatically set as inputs or outputs according to the setting of master or slave and the level input on the SSLx0 pin.
Refer to section 40.3.2, Controlling RSPI Pins for details.

Table 40.2 RSPI Pin Configuration

Channel	Pin Name	I/O	Function
RSPI0	RSPCKA	I/O	Clock I/O
	MOSIA	I/O	Master transmit data I/O
	MISOA	I/O	Slave transmit data I/O
	SSLA0	I/O	Slave selection I/O
	SSLA1	Output	Slave selection output
	SSLA2	Output	Slave selection output
	SSLA3	Output	Slave selection output
RSPI1	RSPCKB	I/O	Clock I/O
	MOSIB	I/O	Master transmit data I/O
	MISOB	I/O	Slave transmit data I/O
	SSLB0	I/O	Slave selection I/O
	SSLB1	Output	Slave selection output
	SSLB2	Output	Slave selection output
	SSLB3	Output	Slave selection output
RSPI2	RSPCKC	I/O	Clock I/O
	MOSIC	I/O	Master transmit data I/O
	MISOC	I/O	Slave transmit data I/O
	SSLC0	I/O	Slave selection I/O
	SSLC1	Output	Slave selection output
	SSLC2	Output	Slave selection output
	SSLC3	Output	Slave selection output

40.2 Register Descriptions

40.2.1 RSPI Control Register (SPCR)

Address(es): RSPI0.SPCR 000D 0100h, RSPI1.SPCR 000D 0140h, RSPI2.SPCR 000D 0300h

	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*1	0: SPI operation (4-wire method) 1: Clock synchronous operation (3-wire method)	R/W
b1	TXMD	Communications Operating Mode Select*1	0: Full-duplex communications (enables the receiver) 1: Transmit-only simplex communications (disables the receiver)	R/W
b2	MODFEN	Mode Fault Error Detection Enable *1	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*1	0: Slave mode 1: Master mode	R/W
b4	SPEIE	Error Interrupt Enable	0: Disables the generation of error interrupt requests 1: Enables the generation of error interrupt requests	R/W
b5	SPTIE	Transmit Buffer Empty Interrupt Enable	0: Disables the generation of transmit buffer empty interrupt requests 1: Enables the generation of transmit buffer empty interrupt requests	R/W
b6	SPE	RSPI Function Enable	0: Disables the RSPI function 1: Enables the RSPI function	R/W
b7	SPRIE	Receive Buffer Full Interrupt Enable	0: Disables the generation of receive buffer full interrupt requests 1: Enables the generation of receive buffer full interrupt requests	R/W

Note 1. Do not change the values of the MSTR, MODFEN, TXMD, and SPMS bits while the SPE bit is 1.

SPMS Bit (RSPI Mode Select)

The SPMS bit selects SPI operation (4-wire method) or clock synchronous operation (3-wire method).
The SSLx0 to SSLx3 pins are not used in clock synchronous operation. The RSPCKx, MOSIx, and MISOx pins handle communications. If clock synchronous operation is to proceed in master mode (SPCR.MSTR = 1), the SPCMDm.CPHA bit can be set to either 0 or 1. Set the CPHA bit to 1 if clock synchronous operation is to proceed in slave mode (SPCR.MSTR = 0). Do not set the CPHA bit 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 communications or transmit-only simplex communications.
When performing communications with the TXMD bit set to 1, the RSPI performs only transmit operations and not receive operations (refer to section 40.3.6, Communications Operating Mode).
When the TXMD bit is set to 1, receive buffer full interrupt requests cannot be used.
When the SPCR3.RXMD bit is set to 1 (receive-only simplex communications), the setting of this bit is ignored.

MODFEN Bit (Mode Fault Error Detection Enable)

The MODFEN bit enables or disables the detection of mode fault error (refer to section 40.3.10, Error Detection). In addition, the RSPI determines the I/O direction of the SSLx0 to SSLx3 pins based on combinations of the MODFEN and MSTR bits (refer to section 40.3.2, 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 RSPCKx, MOSIx, MISOx, and SSLx0 to SSLx3.

SPEIE Bit (Error Interrupt Enable)

The SPEIE bit enables or disables the generation of error interrupt requests when the RSPI detects a mode fault error or underrun error and sets the SPSR.MODF flag to 1, when the RSPI detects an overrun error and sets the SPSR.OVRF flag to 1, or when the RSPI detects a parity error and sets the SPSR.PERF flag to 1 (refer to section 40.3.10, Error Detection).

SPTIE Bit (Transmit Buffer Empty Interrupt Enable)

The SPTIE bit enables or disables the generation of transmit buffer empty interrupt requests when the RSPI detects when the transmit buffer is empty.

A transmit buffer empty interrupt request when transmission starts is generated by setting the SPE and SPTIE bits to 1 at the same time or by setting the SPE bit to 1 after setting the SPTIE bit to 1.

Note that a transmit buffer interrupt is generated when the SPTIE bit is 1 even if the RSPI function is disabled (the SPTIE bit is changed to 0).

SPE Bit (RSPI Function Enable)

The SPE bit enables or disables the RSPI function.

When the SPSR.MODF flag is 1, the SPE bit cannot be set to 1. For details, refer to section 40.3.10, 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 40.3.11, Initializing RSPI. Furthermore, a transmit buffer empty interrupt request is generated by the state of the SPE bit changing from 0 to 1 or from 1 to 0.

SPRIE Bit (Receive Buffer Full Interrupt Enable)

If the RSPI has detected a receive buffer full write after completion of a serial transfer, the SPRIE bit enables or disables the generation of a receive buffer full interrupt request.

40.2.2 RSPI Slave Select Polarity Register (SSLP)

Address(es): RSPI0.SSLP 000D 0101h, RSPI1.SSLP 000D 0141h, RSPI2.SSLP 000D 0301h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SSL0P	SSL0 Signal Polarity Setting	0: SSL0 signal is active low 1: SSL0 signal is active high	R/W
b1	SSL1P	SSL1 Signal Polarity Setting	0: SSL1 signal is active low 1: SSL1 signal is active high	R/W
b2	SSL2P	SSL2 Signal Polarity Setting	0: SSL2 signal is active low 1: SSL2 signal is active high	R/W
b3	SSL3P	SSL3 Signal Polarity Setting	0: SSL3 signal is active low 1: SSL3 signal is active high	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Do not change the SSLP register while the SPCR.SPE bit is 1.

40.2.3 RSPI Pin Control Register (SPPCR)

Address(es): RSPI0.SPPCR 000D 0102h, RSPI1.SPPCR 000D 0142h, RSPI2.SPPCR 000D 0302h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	MOIFE	MOIFV	—	—	SPLP2	SPLP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPLP	RSPI Loopback	0: Normal mode 1: Loopback mode (data is inverted for transmission)	R/W
b1	SPLP2	RSPI Loopback 2	0: Normal mode 1: Loopback mode (data is not inverted for transmission)	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	MOIFV	MOSI Idle Fixed Value	0: The level output on the MOSIx pin during MOSI idling corresponds to low 1: The level output on the MOSIx pin during MOSI idling corresponds to high	R/W
b5	MOIFE	MOSI Idle Value Fixing Enable	0: MOSI output value equals final data from previous transfer 1: MOSI output value equals the value set in the MOIFV bit	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Do not change the SPPCR register while the SPCR.SPE bit is 1.

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 MISOx pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIx pin and the shift register if the SPCR.MSTR bit is 0, and connects (inverts) the input path and output path for the shift register (loopback mode).

SPLP2 Bit (RSPI Loopback 2)

The SPLP2 bit selects the mode of the RSPI pins.

When the SPLP2 bit is set to 1, the RSPI shuts off the path between the MISOx pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIx 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 MOSIx 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 MOSIx 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 MOSIx pin. When the MOIFE bit is 1, the RSPI outputs the fixed value set in the MOIFV bit to the MOSIx pin.

40.2.4 RSPI Status Register (SPSR)

Address(es): RSPI0.SPSR 000D 0103h, RSPI1.SPSR 000D 0143h, RSPI2.SPSR 000D 0303h

	b7	b6	b5	b4	b3	b2	b1	b0
	SPRF	SPCF	SPTEF	UDRF	PERF	MODF	IDLNF	OVRF
Value after reset:	0	0	1	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OVRF	Overrun Error Flag	0: No overrun error occurs 1: An overrun error occurs	R/(W) *1
b1	IDLNF	Idle Flag	0: RSPI is in the idle state 1: RSPI is in the transfer state	R
b2	MODF	Mode Fault Error Flag	0: Neither a mode fault error nor an underrun error occurs 1: A mode fault error or an underrun error occurs	R/(W) *1
b3	PERF	Parity Error Flag	0: No parity error occurs 1: A parity error occurs	R/(W) *1
b4	UDRF	Underrun Error Flag	This flag is used with the MODF flag to check the state in terms of mode fault errors and underrun errors. b4 b2 0 0: Neither a mode fault error nor an underrun error occurs 0 1: A mode fault error occurs 1 1: An underrun error occurs	R/(W) *1, *2
b5	SPTEF	Transmit Buffer Empty Flag	0: Transmit buffer has valid data 1: Transmit buffer has no valid data	R*3
b6	SPCF	Communication End Flag	0: Communication does not start or communication is in progress 1: Communication has ended	R/(W) *1
b7	SPRF	Receive Buffer Full Flag	0: Receive buffer has no valid data 1: Receive buffer has valid data	R*3

Note 1. Only 0 can be written to clear the flag after reading 1.
Note 2. Clear the MODF and UDRF flags at the same time.
Note 3. The write value should be 1.

OVRF Flag (Overrun Error Flag)

The OVRF flag indicates the occurrence of an overrun error. In master mode (when the SPCR.MSTR bit is 1) and when the RSPCK clock auto-stop function is enabled (the SPCR2.SCKASE bit is 1), an overrun error does not occur; accordingly this flag does not become 1. For details, refer to section 40.3.10.1, Overrun Error.

[Setting condition]

- When the next data reception ends while the SPCR.TXMD bit is 0 and the receive buffer is full.
- When the next data reception ends while the SPCR.MSTR bit is 0, the SPCR3.RXMD bit is 1, and the receive buffer is full.

[Clearing condition]

- When the SPSR register is read while the OVRF flag is 1, and then 0 is written to the OVRF flag.

IDLNF Flag (Idle Flag)

The IDLNF flag indicates the transfer status of the RSPI.

[Setting condition]

Master mode

- When both of the conditions in master mode under the [Clearing condition] below are not satisfied.

Slave mode

- When the SPCR.SPE bit is set to 1 (enables the RSPI function).

[Clearing condition]

Master mode

- When the SPCR.SPE bit is set to 0 (disables the RSPI function)
- When all of the following conditions are satisfied.
  1. The transmit buffer is empty (the SPTEF flag is 1)
  2. The SPSSR.SPCP[2:0] bits are 000b
  3. The transmission of the last bit has been completed and the time specified by the SSLND.SLNDL[2:0] and SPND.SPNDL[2:0] bits has elapsed

Slave mode

- When the SPCR.SPE bit is set to 0 (disables the RSPI function).

MODF Flag (Mode Fault Error Flag)

Indicates the occurrence of a mode fault error or an underrun error. The UDRF flag indicates whether the error is a mode fault error or an underrun error.

[Setting condition]

Multi-master mode

- When the input level of the SSLxi 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 SSLxi 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.
- When the serial transfer starts while the SPCR.SPE bit is 1 (enables the RSPI function) and the transmit data are not ready for output, the RSPI detects an underrun error.

The active level of the SSLxi signal is determined by the SSLP.SSLiP bit (SSLi signal polarity setting bit).

[Clearing condition]

- When the SPSR register is read while the MODF flag is 1, and then 0 is written to the MODF flag.

PERF Flag (Parity Error Flag)

Indicates the occurrence of a parity error.

[Setting condition]

- When a data reception ends while the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1, the RSPI detects a parity error.
- When a data reception ends while the SPCR.MSTR bit is 0, the SPCR3.RXMD bit is 1, and the SPCR2.SPPE bit is 1, the RSPI detects a parity error.

[Clearing condition]

- When the SPSR register is read while the PERF flag is 1, and then 0 is written to the PERF flag.

UDRF Flag (Underrun Error Flag)

Indicates the occurrence of an underrun error. When this flag becomes 1, the MODF flag becomes 1 too. When the MODF flag is 1 and this flag is 0, the error is a mode fault error.

[Setting condition]

- When the serial transfer starts while the SPCR.MSTR bit is 0 (slave mode), the SPCR3.RXMD bit is 0, the SPCR.SPE bit is 1 (enables the RSPI function), and the transmit data are not ready for output, the RSPI detects an underrun error

[Clearing condition]

- When 0 is written to the UDRF flag after reading the SPSR register while the UDRF flag is 1

SPTEF Flag (Transmit Buffer Empty Flag)

Indicates whether the transmit buffer (SPTX) in the RSPI data register has valid data.

[Setting condition]

- When the SPCR.SPE bit is set to 0 (disables the RSPI function).
- When the number of frames of transmit data specified by the SPDCR.SPFC[1:0] bits have been transferred from the transmit buffer to the shift register.

[Clearing condition]

- When the number of frames of transmit data specified by the SPDCR.SPFC[1:0] bits is written to the SPDR register.

The SPDR register can be set only when the SPTEF flag is 1. The data in the transmit buffer is not updated when the SPDR register is set while the SPTEF flag is 0.

SPCF Flag (Communication End Flag)

This flag indicates a completion of the RSPI communication.

[Setting condition]

Master mode

- When all of the following conditions are satisfied.
  1. The transmit buffer is empty (the SPTEF flag is 1)
  2. The SPSSR.SPCP[2:0] bits are 000b
  3. The transmission of the last bit has been completed and the time specified by the SSLND.SLNDL[2:0] and SPND.SPNDL[2:0] bits has elapsed

Full-duplex or transmit-only simplex communications in slave mode (SPI operation)

- When all of the following conditions are satisfied.
  1. The transmit buffer is empty (the SPTEF flag is 1)
  2. The transmit shift register is empty
  3. The SSLx0 signal has been negated

Full-duplex or transmit-only simplex communications in slave mode (clock synchronous operation)

- When all of the following conditions are satisfied.
  1. The transmit buffer is empty (the SPTEF flag is 1)
  2. The transmit shift register is empty
  3. The last bit of the last data has been received (the last even edge of the RSPCK)

Receive-only simplex communications in slave mode (SPI operation)

- When the SSLx0 signal is negated after the number of frames set in the SPDCR.SPFC[1:0] bits have been received.

Receive-only simplex communications in slave mode (clock synchronous operation)

- When the number of frames set in the SPDCR.SPFC[1:0] bits have been received (the last even edge of the

RSPCK).

[Clearing condition]

Full-duplex or transmit-only simplex communications

- When the next transmit data is written to the transmit buffer
- When 0 is written to the SPCF flag after reading the SPSR register while the SPCF flag is 1

Receive-only simplex communications in SPI operation

- When the assertion of the SSLx0 signal for the next data has been detected
- When 0 is written to the SPCF flag after reading the SPSR register while the SPCF flag is 1

Receive-only simplex communications in slave mode (clock synchronous operation)

- When the first edge of the RSPCK signal for the next data has been detected
- When 0 is written to the SPCF flag after reading the SPSR register while the SPCF flag is 1

SPRF Flag (Receive Buffer Full Flag)

Indicates whether the receive buffer (SPRX) in the RSPI data register has valid data.

[Setting condition]

- When the number of frames of receive data specified by the SPDCR.SPFC[1:0] bits is transferred from shift register to the receive buffer (SPRX) while the SPCR.TXMD bit is 0 (full duplex) and the SPRF flag is 0.
  Note that the SPRF flag does not become 1 when the OVRF flag is 1.
- When the number of frames of receive data specified by the SPDCR.SPFC[1:0] bits is transferred from shift register to the receive buffer (SPRX) while the SPCR.MSTR bit is 0, the SPCR3.RXMD bit is 1, and the SPRF flag is 0.
  Note that the SPRF flag does not become 1 when the OVRF flag is 1.

[Clearing condition]

- When all of the received data are read from the SPDR register.

40.2.5 RSPI Data Register (SPDR)

- When accessing in longword size

Address(es): RSPI0.SPDR 000D 0104h, RSPI1.SPDR 000D 0144h, RSPI2.SPDR 000D 0304h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

- When accessing in word size

Address(es): RSPI0.SPDR.H 000D 0104h, RSPI1.SPDR.H 000D 0144h, RSPI2.SPDR.H 000D 0304h

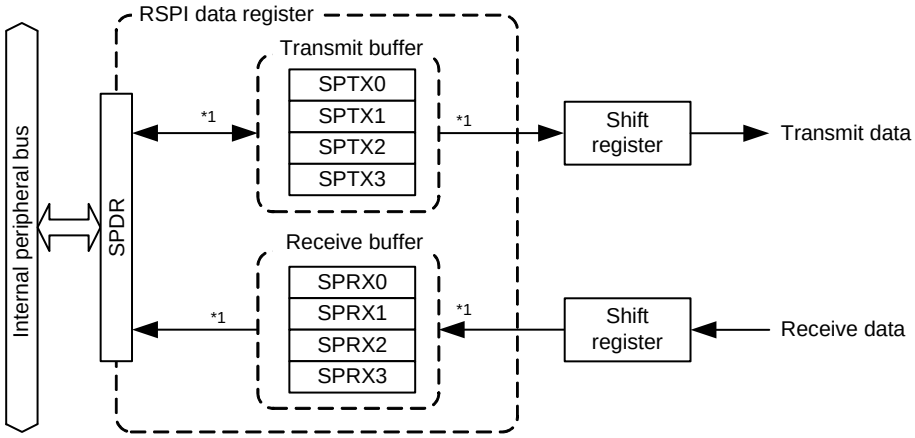
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	H															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- When accessing in byte size

Address(es): RSPI0.SPDR.HH 000D 0104h, RSPI1.SPDR.HH 000D 0144h, RSPI2.SPDR.HH 000D 0304h

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

The SPDR register is the interface with the buffers that hold data for transmission and reception by the RSPI.
When accessing in longwords (the SPLW bit is 1 and the SPBYT bit is 0), access the SPDR register in 32-bit units.
When accessing in words (the SPLW bit is 0 and the SPBYT bit is 0), access the SPDR.H register in 16-bit units.
When accessing in bytes (the SPBYT bit is 1), access the SPDR.HH register in 8-bit units.
The transmit buffer (SPTX) and receive buffer (SPRX) are independent but are both mapped to the SPDR register.
Figure 40.2 shows the Configuration of the SPDR Register.



Note 1. The destination buffer and stage for access is automatically switched by hardware.

Figure 40.2 Configuration of the SPDR Register

The transmit and receive buffers each have four stages. The number of stages to be used is selectable by the number of frames specification bits in the RSPI data control register (SPDCR.SPFC[1:0]). The eight stages of the buffer are all mapped to the single address of the SPDR register.

Data written to the SPDR register are written to a transmit-buffer stage (SPTXn) (n = 0 to 3) and then transmitted from the buffer. The receive buffer holds received data on completion of reception. The receive buffer is not updated if an overrun is generated.

Furthermore, if the data length is other than 32 bits, bits not referred to in SPTXn (n = 0 to 3) are stored in the corresponding bits in SPRXn. For example, if the data length is 9 bits, received data are stored in the SPRXn[8:0] bits and the SPTXn[31:9] bits are stored in the SPRXn[31:9] bits.

(1) Bus Interface

The SPDR register is the interface with 32-bit wide transmit and receive buffers, each of which has four stages, for a total of 32 bytes. In other words, the 32 bytes are mapped to the 4-byte address space for the SPDR register. Furthermore, the unit of access for the SPDR register is selected by the SPDCR.SPLW bit and the SPDCR.SPBYT bit.
Data for transmission should be flush with the LSB end of the register. Received data are stored flush with the LSB end.
Operations involved in writing to and reading from the SPDR register are described below.

(a) Writing

Data written to the SPDR register are written to a transmit buffer (SPTXn). This is not influenced by the value of the SPDCR.SPRDTD bit unlike when reading from the SPDR register.
The transmit buffer includes a transmit buffer write pointer which is automatically updated to indicate the next stage each time data are written to the SPDR register.
Figure 40.3 shows the configuration of the bus interface with the transmit buffer in the case of writing to the SPDR register.

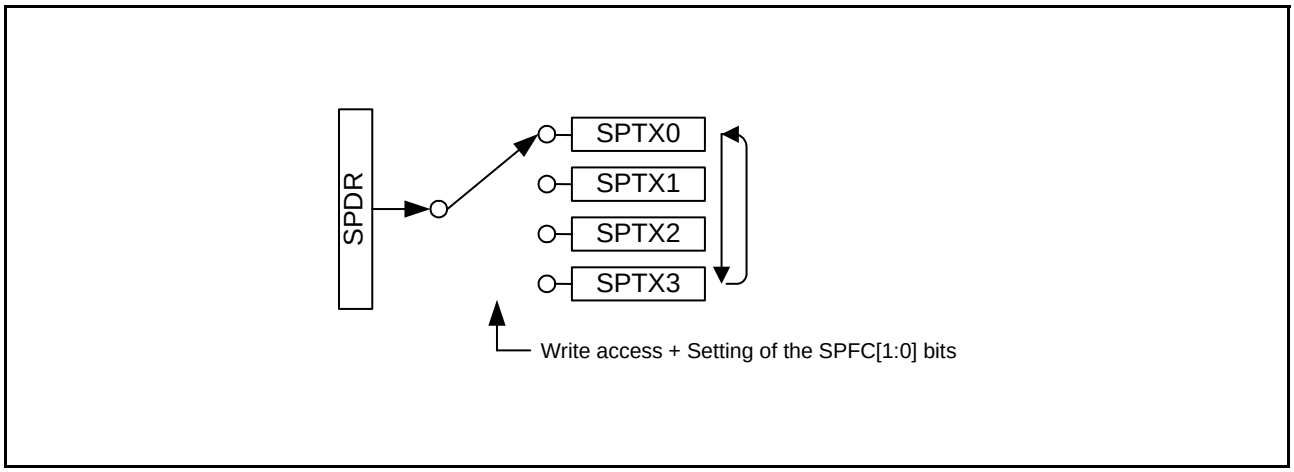


Figure 40.3 Configuration of the SPDR Register (Writing)

The sequence for switching the transmit buffer write pointer differs with the setting of the number of frames specification bits in the RSPI data control register (SPDCR.SPFC[1:0]).

- Settings of the SPFC[1:0] bits and sequence of switching the pointer among SPTX0 to SPTX3.
  When the SPFC[1:0] bits are 00b: SPTX0 → SPTX0 → SPTX0 → …
  When the SPFC[1:0] bits are 01b: SPTX0 → SPTX1 → SPTX0 → SPTX1 → …
  When the SPFC[1:0] bits are 10b: SPTX0 → SPTX1 → SPTX2 → SPTX0 → SPTX1 → …
  When the SPFC[1:0] bits are 11b: SPTX0 → SPTX1 → SPTX2 → SPTX3 → SPTX0 → SPTX1 → …

When 1 is written to the RSPI function enable bit in the RSPI control register (SPCR.SPE) while the bit's current value is 0, SPTX0 will be the destination the next time writing proceeds.
When writing to the transmit buffer (SPTXn) after generation of the transmit buffer empty interrupt (after the SPSR.SPTEF flag becomes 1), write the number of frames set by the number of frames specification bits (SPFC[1:0]) in the RSPI data control register (SPDCR). Even if the number of frames is written to the transmit buffer (SPTXn), the value of the buffer is not updated after completion of the writing and before generation of the next transmit buffer empty interrupt (while the SPSR.SPTEF flag is 0).

(b) Reading

The SPDR register can be read to read the value of a receive buffer (SPRXn) or a transmit buffer (SPTXn). The setting of the RSPI receive/transmit data select bit in the RSPI data control register (SPDCR.SPRDTD) selects whether reading is of the receive or transmit buffer.

The sequence of reading the SPDR register is controlled by independent pointers, receive buffer read pointer and transmit buffer read pointer.

Figure 40.4 shows the configuration of the bus interface with the receive and transmit buffers in the case of reading from the SPDR register.

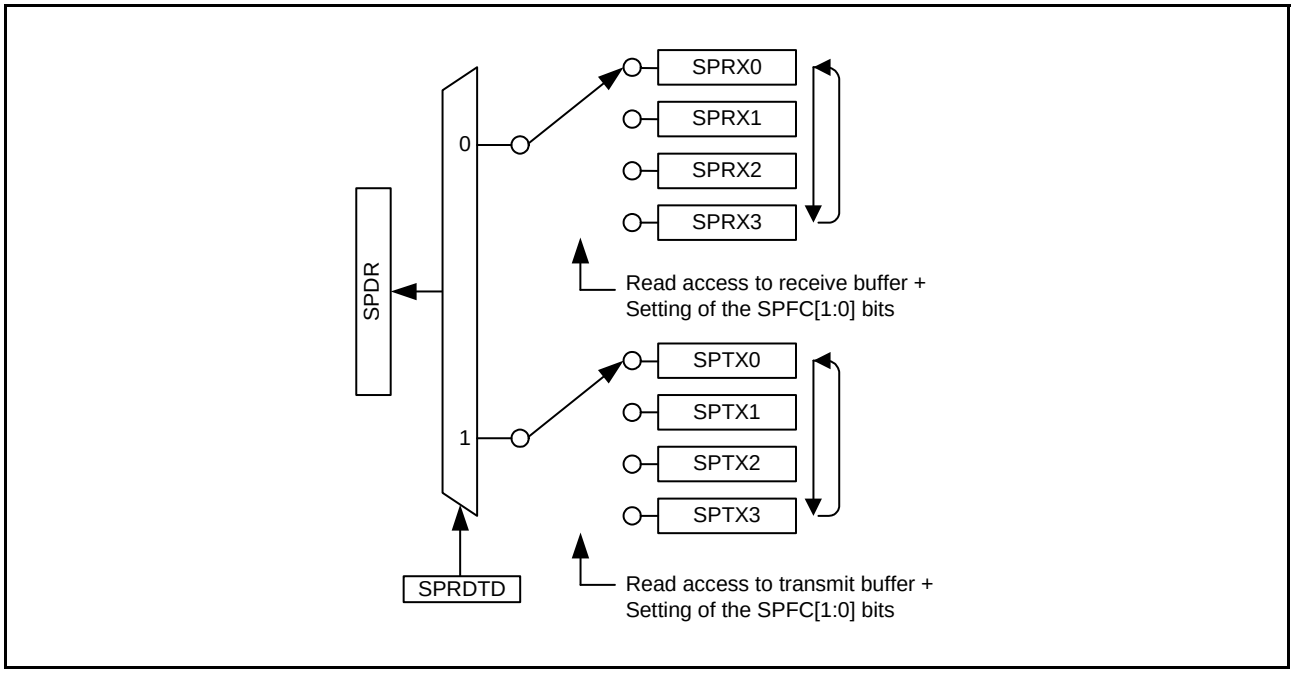


Figure 40.4 Configuration of the SPDR Register (Reading)

Reading the receive buffer switches the receive buffer read pointer to the next buffer automatically.

The sequence of switching the receive buffer read pointer is the same as that for the transmit buffer write pointer. However, when 1 is written to the RSPI function enable bit in the RSPI control register (SPCR.SPE) while the bit's current value is 0, SPRX0 will be indicated by the buffer read pointer the next time reading proceeds.

The transmit buffer read pointer is updated when writing to the SPDR register, and not updated when reading from the transmit buffer. When reading from the transmit buffer, the value most recently written to the SPDR register is read. However, after generation of the transmit buffer empty interrupt, the values read from the transmit buffer are all 0 in the interval after completion of writing the number of frames of data specified in the number of frames specification bits (SPDCR.SPFC[1:0]) and before generation of the next buffer empty interrupt (while the SPSR.SPTEF flag is 0).

40.2.6 RSPI Sequence Control Register (SPSCR)

Address(es): RSPI0.SPSCR 000D 0108h, RSPI1.SPSCR 000D 0148h, RSPI2.SPSCR 000D 0308h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SPSLN[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPSLN[2:0]	RSPI Sequence Length Specification	b2 b0 Sequence Length — Referenced SPCMD0 to SPCMD7 registers (No.) 0 0 0: 1 — 0→0→… 0 0 1: 2 — 0→1→0→… 0 1 0: 3 — 0→1→2→0→… 0 1 1: 4 — 0→1→2→3→0→… 1 0 0: 5 — 0→1→2→3→4→0→… 1 0 1: 6 — 0→1→2→3→4→5→0→… 1 1 0: 7 — 0→1→2→3→4→5→6→0→… 1 1 1: 8 — 0→1→2→3→4→5→6→7→0→… The order in which the SPCMD0 to SPCMD7 registers are to be referenced is changed according to the sequence length that is set in these bits. The relationship among the setting of these bits, sequence length, and the SPCMD0 to SPCMD7 registers referenced by the RSPI is shown above. However, the RSPI in slave mode references the SPCMD0 register.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SPSCR register 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 the 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, the SPCMD0 register is referred.

40.2.7 RSPI Sequence Status Register (SPSSR)

Address(es): RSPI0.SPSSR 000D 0109h, RSPI1.SPSSR 000D 0149h, RSPI2.SPSSR 000D 0309h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SPECM[2:0]			—	SPCP[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPCP[2:0]	RSPI Command Pointer	b2 b0 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
b3	—	Reserved	This bit is read as 0.	R
b6 to b4	SPECM[2:0]	RSPI Error Command	b6 b4 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
b7	—	Reserved	This bit is read as 0.	R

The SPSSR register indicates the sequence control status when the RSPI operates in master mode.
Any writing to the SPSSR register is ignored.

SPCP[2:0] Bits (RSPI Command Pointer)

The SPCP[2:0] bits indicate the SPCMDm register that is currently pointed to by the pointer during sequence control by the RSPI.
For the RSPI's sequence control, refer to section 40.3.12.1, Master Mode Operation.

SPECM[2:0] Bits (RSPI Error Command)

The SPECM[2:0] bits indicate the SPCMDm register that is specified by the SPCP[2:0] bits when an error is detected during sequence control by the RSPI. The RSPI updates the SPECM[2:0] bits only when an error is detected. If both the SPSR.OVRF and SPSR.MODF flags are 0 and there is no error, the values of the SPECM[2:0] bits have no meaning.
For the RSPI's error detection function, refer to section 40.3.10, Error Detection. For the RSPI's sequence control, refer to section 40.3.12.1, Master Mode Operation.

40.2.8 RSPI Bit Rate Register (SPBR)

Address(es): RSPI0.SPBR 000D 010Ah, RSPI1.SPBR 000D 014Ah, RSPI2.SPBR 000D 030Ah

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

The SPBR register sets the bit rate in master mode. Do not change the SPBR register while both the SPCR.MSTR and SPCR.SPE bits are 1.

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 the SPBR register and the SPCMDm.BRDV[1:0] bits (bit rate division setting bits).
The bit rate is determined by combinations of the SPBR register setting and the SPCMDm.BRDV[1:0] bit setting. The equation for calculating the bit rate is given below. In the equation, n denotes the SPBR register setting (0, 1, 2, …, 255), and N denotes the BRDV[1:0] bit setting (0, 1, 2, 3).

$$\text{Bit rate} = \frac{f(PCLK)}{2 \times (n+1) \times 2^N}$$

Table 40.3 lists examples of the relationship among the SPBR register settings, the BRDV[1:0] bit settings, and bit rates.
Use the bit rate that meets electrical characteristics based on the AC specifications of the target device.

Table 40.3 Relationship among SPBR Settings, BRDV[1:0] Settings, and Bit Rates

SPBR (n)	BRDV[1:0] Bits (N)	Division Ratio	Bit Rate							
			PCLK = 32 MHz	PCLK = 36 MHz	PCLK = 40 MHz	PCLK = 50 MHz	PCLK = 60 MHz	PCLK = 80 MHz	PCLK = 100 MHz	PCLK = 120 MHz
0	0	2	16.0 Mbps	18.0 Mbps	20.0 Mbps	25.0 Mbps	30.0 Mbps	40.0 Mbps	—	—
1	0	4	8.00 Mbps	9.00 Mbps	10.0 Mbps	12.5 Mbps	15.0 Mbps	20.0 Mbps	25.0 Mbps	30.0 Mbps
2	0	6	5.33 Mbps	6.00 Mbps	6.67 Mbps	8.33 Mbps	10.0 Mbps	13.3 Mbps	16.7 Mbps	20.0 Mbps
3	0	8	4.00 Mbps	4.50 Mbps	5.00 Mbps	6.25 Mbps	7.50 Mbps	10.0 Mbps	12.5 Mbps	15.0 Mbps
4	0	10	3.20 Mbps	3.60 Mbps	4.00 Mbps	5.00 Mbps	6.00 Mbps	8.00 Mbps	10.0 Mbps	12.0 Mbps
5	0	12	2.67 Mbps	3.00 Mbps	3.33 Mbps	4.16 Mbps	5.00 Mbps	6.67 Mbps	8.33 Mbps	10.0 Mbps
5	1	24	1.33 Mbps	1.50 Mbps	1.67 Mbps	2.08 Mbps	2.50 Mbps	3.33 Mbps	4.17 Mbps	5.00 Mbps
5	2	48	667 kbps	750 kbps	833 kbps	1.04 Mbps	1.25 Mbps	1.67 Mbps	2.08 Mbps	2.50 Mbps
5	3	96	333 kbps	375 kbps	417 kbps	521 kbps	625 kbps	833 kbps	1.04 Mbps	1.25 Mbps
255	3	4096	7.81 kbps	8.80 kbps	9.78 kbps	12.2 kbps	14.6 kbps	19.5 kbps	24.4 kbps	29.3 kbps

40.2.9 RSPI Data Control Register (SPDCR)

Address(es): RSPI0.SPDCR 000D 010Bh, RSPI1.SPDCR 000D 014Bh, RSPI2.SPDCR 000D 030Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SPBYT	SPLW	SPRDTD	—	—	SPFC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	SPFC[1:0]	Number of Frames Specification	b1 b0 0 0: 1 frame 0 1: 2 frames 1 0: 3 frames 1 1: 4 frames	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SPRDTD	RSPI Receive/Transmit Data Select	0: The SPDR values are read from the receive buffer 1: The 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*1	0: The SPDR register is accessed in words 1: The SPDR register is accessed in longwords	R/W
b6	SPBYT	RSPI Byte Access Specification	0: The SPDR register is accessed in words or longwords (the SPLW bit is enabled) 1: The SPDR register is accessed in bytes (the SPLW bit is disabled)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Set the SPBYT bit to 0, when accessing the SPDR register in words or longwords.

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 the SPDR register (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 receive buffer full interrupt, and start of transmission or generation of transmit buffer empty interrupts.

When the number of frames of transmit data specified by SPFC[1:0] bits is written to the SPDR register, the SPSR.SPTEF flag becomes 0 and transmission starts. Then, when the specified number of frames of transmit data has been transferred to the shift register, the SPTEF flag becomes 1 and the RSPI transmit buffer empty interrupt is generated.

When the number of frames specified by the SPFC[1:0] bits are received, the SPSR.SPRF flag becomes 1 and the receive buffer full interrupt is generated.

Table 40.4 lists the frame configurations that can be stored in the SPDR register and examples of combinations of settings for transmission and reception. Do not select the combinations of settings other than those shown in the examples.

Table 40.4 Settable Combinations of SPSLN[2:0] Bits and SPFC[1:0] Bits

Setting	SPSLN[2:0]	SPFC[1:0]	Number of Frames in a Single Sequence	Number of Frames at which Transmit Buffer or Receive Buffer Status Becomes “Has Valid Data”
1-1	000b	00b	1	1
1-2	000b	01b	2	2
1-3	000b	10b	3	3
1-4	000b	11b	4	4
2-1	001b	01b	2	2
2-2	001b	11b	4	4
3	010b	10b	3	3
4	011b	11b	4	4
5	100b	00b	5	1
6	101b	00b	6	1
7	110b	00b	7	1
8	111b	00b	8	1

SPRDTD Bit (RSPI Receive/Transmit Data Select)

The SPRDTD bit selects whether the SPDR register reads values from the receive buffer or from the transmit buffer. If reading is from the transmit buffer, the value written to the SPDR register immediately beforehand is read.
When reading the transmit buffer, do so before writing of the number of frames set in the SPFC[1:0] bits is finished and after generation of the transmit buffer empty interrupt (While the SPSR.SPTEF flag is 1).
For details, refer to section 40.2.5, RSPI Data Register (SPDR).

SPLW Bit (RSPI Longword Access/Word Access Specification)

The SPLW bit specifies the access width for the SPDR register. This bit setting is enabled when the SPBYT bit is 0. Access to the SPDR register in words when the SPLW bit is 0 and in longwords when the SPLW bit is 1.
Also, when the SPLW bit is 0, set the SPCMDm.SPB[3:0] bits (RSPI data length setting bits) to 8 to 16 bits. Do not select 20, 24, or 32 bits.

SPBYT Bit (RSPI Byte Access Specification)

The SPBYT bit specifies the access width for the SPDR register. Access to the SPDR register according to the SPLW bit setting when the SPBYT bit is 0. Access to the SPDR register in bytes when the SPBYT bit is 1.
When the SPBYT bit is 1, set the SPCMDm.SPB[3:0] bits (RSPI data length setting bits) to 8 bits. Do not select 9 to 16, 20,24, or 32 bits.

40.2.10 RSPI Clock Delay Register (SPCKD)

Address(es): RSPI0.SPCKD 000D 010Ch, RSPI1.SPCKD 000D 014Ch, RSPI2.SPCKD 000D 030Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SCKDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SCKDL[2:0]	RSPCK Delay Setting	b2 b1 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

The SPCKD register sets a period from the beginning of SSLxi signal assertion to RSPCK oscillation (RSPCK delay) when the SPCMDm.SCKDEN bit is 1. Do not change the SPCKD register while both the SPCR.MSTR and SPCR.SPE bits are 1.

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.

40.2.11 RSPI Slave Select Negation Delay Register (SSLND)

Address(es): RSPI0.SSLND 000D 010Dh, RSPI1.SSLND 000D 014Dh, RSPI2.SSLND 000D 030Dh

	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

The SSLND register sets a period (SSL negation delay) from the transmission of a final RSPCK edge to the negation of the SSLxi signal during a serial transfer by the RSPI in master mode. Do not change the SSLND register while both the SPCR.MSTR and SPCR.SPE bits are 1.

SLNDL[2:0] Bits (SSL Negation Delay Setting)

The SLNDL[2:0] bits set an SSL negation delay value when the SPCMDm.SLNDEN bit is 1.
When using the RSPI in slave mode, set the SLNDL[2:0] bits to 000b.

40.2.12 RSPI Next-Access Delay Register (SPND)

Address(es): RSPI0.SPND 000D 010Eh, RSPI1.SPND 000D 014Eh, RSPI2.SPND 000D 030Eh

	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 SSLxi signal after termination of a serial transfer when the SPCMDm.SPNDEN bit is 1. Do not change the SPND register while both the SPCR.MSTR and SPCR.SPE bits are 1.

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.

40.2.13 RSPI Control Register 2 (SPCR2)

Address(es): RSPI0.SPCR2 000D 010Fh, RSPI1.SPCR2 000D 014Fh, RSPI2.SPCR2 000D 030Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SCKASE	PTE	SPIIE	SPOE	SPPE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPPE	Parity Enable*1	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	R/W
b1	SPOE	Parity Mode*1	0: Selects even parity for use in transmission and reception 1: Selects odd parity for use in transmission and reception	R/W
b2	SPIIE	Idle Interrupt Enable	0: Disables the generation of idle interrupt requests 1: Enables the generation of idle interrupt requests	R/W
b3	PTE	Parity Self-Diagnosis	0: Disables the self-diagnosis function of the parity circuit 1: Enables the self-diagnosis function of the parity circuit	R/W
b4	SCKASE	RSPCK Auto-Stop Function Enable*1	0: Disables the RSPCK auto-stop function 1: Enables the RSPCK auto-stop function	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Do not change the SPPE, SPOE, and SCKASE bits while the SPCR.SPE bit is 1.

SPPE Bit (Parity Enable)

The SPPE bit enables or disables the parity function.

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 (Idle Interrupt Enable)

The SPIIE bit enables or disables the generation of idle interrupt requests when the RSPI being in the idle state is detected and the SPSR.IDLNF flag is set to 0.

PTE Bit (Parity Self-Diagnosis)

The PTE bit enables the self-diagnosis function of the parity circuit in order to check whether the parity function is operating correctly.

SCKASE Bit (RSPCK Auto-Stop Function Enable)

The SCKASE bit enables or disables the RSPCK auto-stop function. When this function is enabled, the RSPCK clock is stopped before an overrun error occurs when data is received in master mode. For details, refer to section 40.3.10.1, Overrun Error.

40.2.14 RSPI Command Register m (SPCMDm) (m = 0 to 7)

Address(es): RSPI0.SPCMD0 000D 0110h, RSPI0.SPCMD1 000D 0112h, RSPI0.SPCMD2 000D 0114h, RSPI0.SPCMD3 000D 0116h, RSPI0.SPCMD4 000D 0118h, RSPI0.SPCMD5 000D 011Ah, RSPI0.SPCMD6 000D 011Ch, RSPI0.SPCMD7 000D 011Eh, RSPI1.SPCMD0 000D 0150h, RSPI1.SPCMD1 000D 0152h, RSPI1.SPCMD2 000D 0154h, RSPI1.SPCMD3 000D 0156h, RSPI1.SPCMD4 000D 0158h, RSPI1.SPCMD5 000D 015Ah, RSPI1.SPCMD6 000D 015Ch, RSPI1.SPCMD7 000D 015Eh, RSPI2.SPCMD0 000D 0310h, RSPI2.SPCMD1 000D 0312h, RSPI2.SPCMD2 000D 0314h, RSPI2.SPCMD3 000D 0316h, RSPI2.SPCMD4 000D 0318h, RSPI2.SPCMD5 000D 031Ah, RSPI2.SPCMD6 000D 031Ch, RSPI2.SPCMD7 000D 031Eh

	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	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 (burst transfer)	R/W
b11 to b8	SPB[3:0]	RSPI Data Length Setting	b11 b8 0100 to 0111: 8 bits 1 0 0 0: 9 bits 1 0 0 1: 10 bits 1 0 1 0: 11 bits 1 0 1 1: 12 bits 1 1 0 0: 13 bits 1 1 0 1: 14 bits 1 1 1 0: 15 bits 1 1 1 1: 16 bits 0 0 0 0: 20 bits 0 0 0 1: 24 bits 0010, 0011: 32 bits	R/W
b12	LSBF	RSPI LSB First	0: MSB first 1: LSB first	R/W
b13	SPNDEN	RSPI Next-Access Delay Enable	0: A next-access delay of 1 RSPCK + 2 PCLK 1: A next-access delay is equal to the setting of the RSPI next-access delay register (SPND)	R/W
b14	SLNDEN	SSL Negation Delay Setting Enable	0: An SSL negation delay of 1 RSPCK 1: An SSL negation delay is equal to the setting of the RSPI slave select negation delay register (SSLND)	R/W
b15	SCKDEN	RSPCK Delay Setting Enable	0: An RSPCK delay of 1 RSPCK 1: An RSPCK delay is equal to the setting of the RSPI clock delay register (SPCKD)	R/W

x: Don’t care

The 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 the SPCMD0 register is used to set a transfer mode for the RSPI in slave mode. The RSPI in master mode sequentially references the 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.
An SPCMDm register that is referenced by the RSPI in master mode can be checked by means of the SPSSR.SPCP[2:0] bits. Do not change the SPCMDm register while the SPCR.MSTR bit is 0 and the SPCR.SPE bit is 1.

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 the SPBR register (refer to section 40.2.8, RSPI Bit Rate Register (SPBR)). The settings in the SPBR register 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 the SPCMDm register, different BRDV[1:0] bit settings can be specified. This enables execution of serial transfers at a different bit rate for each command.

SSLA[2:0] Bits (SSL Signal Assertion Setting)

The SSLA[2:0] bits control the SSLxi signal assertion when the RSPI performs serial transfers in master mode.
Setting the SSLA[2:0] bits controls the assertion for the SSLxi signal. When an SSLxi signal is asserted, its polarity is determined by the set value in the corresponding SSLP. When the SSLA[2:0] bits are set to 000b in multi-master mode, serial transfers are performed with all the SSL signals in the negated state (as the SSLx0 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 SSLxi signal level for the current command is to be kept or negated between the SSL negation timing associated with the current command and the SSL assertion timing associated with the next command.
Setting the SSLKP bit to 1 enables a burst transfer. For details, refer to section 40.3.12.1, Master Mode Operation (4) Burst Transfer.
When using the RSPI in slave mode, the SSLKP bit should be set to 0.

SPB[3:0] Bits (RSPI Data Length Setting)

The SPB[3:0] bits set a transfer data length for the RSPI in master mode or slave mode. When the SPDCR.SPBYT is 1, set the SPB[3:0] bits to 0100b (8 bits). When the SPBYT bit is 0 and the SPDCR.SPLW bit is 0, set the SPB[3:0] bits to 0100b (8 bits) to 1111b (16 bits).

LSBF Bit (RSPI LSB First)

The LSBF bit sets the data format of the RSPI in master mode or slave mode to MSB first or LSB first.

SPNDEN Bit (RSPI Next-Access Delay Enable)

The SPNDEN bit sets the period from the time the RSPI in master mode terminates a serial transfer and sets the SSLxi signal inactive until the RSPI enables the SSLxi 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 SSLxi 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 SSLxi 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.

40.2.15 RSPI Data Control Register 2 (SPDCR2)

Address(es): RSPI0.SPDCR2 000D 0120h, RSPI1.SPDCR2 000D 0160h, RSPI2.SPDCR2 000D 0320h

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

Bit	Symbol	Bit Name	Description	R/W
b0	BYSW	Byte Swap	0: Byte swapping of SPDR data disabled 1: Byte swapping of SPDR data enabled	R/W
b1	DINV	Transfer Data Invert	0: Data bits in the transmit buffer are transferred to the shift register as they are. Data bits in the shift register are transferred to the receive buffer as they are. 1: Data bits in the transmit buffer are transferred to the shift register with inverting. Data bits in the shift register are transferred to the receive buffer with inverting.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to enable or disable byte swapping and logic inverting of transmit and receive data.
The SPCR.SPE bit should be 0 when rewriting this register.

BYSW Bit (Byte Swap)

On transmit, this bit specifies that data bytes written in the SPDR register will be swapped before being transmitted. On receive, this bit specifies that received bytes will be swapped before the data is transferred to the SPDR register. This bit setting is enabled when the SPDCR.SPBYT bit is 0.
When using byte swap, set the SPCMD.SPB[3:0] bits to 1111b (16 bits), 0010b (32 bits), or 0011b (32 bits). Also, set the SPCR2.SPPE bit to 0 (parity bit not added). For details, refer to sections 40.3.4.3 Byte Swap Transmission and 40.3.4.4 Byte Swap Reception.

DINV Bit (Transfer Data Invert)

This bit is used to invert the logic levels of the data bits when the data is transferred from the transmit buffer to the shift register and from the shift register to the receive buffer.
The parity bit is added for the data in the transmit shift register and the parity is checked for the data in the receive shift register.

40.2.16 RSPI Control Register 3 (SPCR3)

Address(es): RSPI0.SPCR3 000D 0121h, RSPI1.SPCR3 000D 0161h, RSPI2.SPCR3 000D 0321h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SPCIE	—	—	SCKDDIS	RXMD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RXMD	Receive Operating Mode Setting	0: Full-duplex or transmit-only simplex communications (enables the transmitter) 1: Receive-only simplex communications (disables the transmitter)	R/W
b1	SCKDDIS	RSPCK Delay Between Data Byte Disable	0: Inserts delays between data bytes during burst transfer 1: Does not insert delays between data bytes during burst transfer	R/W
b3 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	SPCIE	Communication End Interrupt Enable	0: Disables the generation of communication end interrupt requests 1: Enables the generation of communication end interrupt requests	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R

Note 1. Do not change the RXMD and SCKDDIS bits while the SPCR.SPE bit is 1.

This register is used to set the operating mode of the RSPI.

RXMD Bit (Receive Operating Mode Setting)

This bit is used to stop the transmitter and operate only the receiver. This bit is valid only in slave mode.
When this bit is 1, the setting of the SPCR.TXMD bit is ignored.

SCKDDIS Bit (RSPCK Delay Between Data Byte Disable)

This bit is used to select whether the RSPCK delay between data bytes is to be inserted or not during burst transfer. This bit is valid only when the SPCR.MSTR bit is 1 (master mode) and the SPCMDm.SSLKP bit is 1. Set this bit to 0 in slave mode.

SPCIE Bit (Communication End Interrupt Enable)

This bit is used to enable or disable the generation of the communication end interrupt request.

40.3 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.

40.3.1 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 40.5 lists the relationship between RSPI modes and SPCR settings, and a description of each mode.

Table 40.5 Relationship between RSPI Modes and SPCR Settings and Description of Each Mode

	SPI Operation			**Clock Synchronous Operation**	
Mode	**Slave**	**Single-Master**	**Multi-Master**	**Slave**	**Master**
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
RSPCKx signal	Input	Output	Output/Hi-Z*1	Input	Output
MOSIx signal	Input	Output	Output/Hi-Z*1	Input	Output
MISOx signal	Output/Hi-Z*2	Input	Input	Output	Input
SSLx0 signal	Input	Output	Input	Hi-Z*3	Hi-Z*3
SSLx1 to SSLx3 signals	Hi-Z*3	Output	Output/Hi-Z*1	Hi-Z*3	Hi-Z*3
SSL polarity change function	Supported	Supported	Supported	—	—
Transfer rate	Up to PCLK/4	Up to PCLK/2	Up to PCLK/2	Up to PCLK/4	Up to PCLK/2
Clock source	RSPCK input	On-chip baud rate generator	On-chip baud rate generator	RSPCK input	On-chip baud rate generator
Clock polarity	Two				
Clock phase	Two	Two	Two	One (CPHA = 1)	Two
First transfer bit	MSB/LSB				
Transfer data length	8 to 16, 20, 24, 32 bits				
Burst transfer	Possible (CPHA = 1)	Possible (CPHA = 0,1)	Possible (CPHA = 0,1)	—	—
RSPCK delay control	Not supported	Supported	Supported	Not supported	Supported
SSL negation delay control	Not supported	Supported	Supported	Not supported	Supported
Next-access delay control	Not supported	Supported	Supported	Not supported	Supported
Transfer activation method	SSL input active or RSPCK oscillation	Transmit buffer is written to when a transmit buffer empty interrupt request is generated or the SPTEF flag is 1	Transmit buffer is written to when a transmit buffer empty interrupt request is generated or the SPTEF flag is 1	RSPCK oscillation	Transmit buffer is written to when a transmit buffer empty interrupt request is generated or the SPTEF flag is 1
Sequence control	Not supported	Supported	Supported	Not supported	Supported
Transmit buffer empty detection	Supported*7	Supported	Supported	Supported*7	Supported
Receive buffer full detection	Supported*4				
Overrun error detection	Supported*4	Supported*4, *6	Supported*4, *6	Supported*4	Supported*4, *6
Underrun error detection	Supported*7	Not supported	Not supported	Supported*7	Not supported
Parity error detection	Supported*4, *5				
Mode fault error detection	Supported (MODFEN = 1)	Not supported	Supported	Not supported	Not supported

Note 1. When SSLx0 is asserted by another master device, the pin becomes Hi-Z.
Note 2. When SSLx0 is negated or the SPCR.SPE bit is 0, the pin becomes Hi-Z.
Note 3. This function is not supported in this mode.
Note 4. When the SPCR.TXMD bit is 1, the detections of receiver buffer full, overrun error, and parity error are not performed.
Note 5. When the SPCR2.SPPE bit is 0, the detection of parity error is not performed.

Note 6. When the SPCR2.SCKASE bit is 1, the detection of overrun error is not performed.
Note 7. When the SPCR3.RXMD bit is 1, the detections of transmit buffer empty and underrun error are not performed.

40.3.2 Controlling RSPI 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 40.6.

Table 40.6 MOSI Signal Value Determination during SSL Negation Period

MOIFE Bit	MOIFV Bit	MOSIx Signal Value during SSL Negation Period
0	0, 1	Final data from previous transfer
1	0	Low
1	1	High

40.3.3 RSPI System Configuration Examples

40.3.3.1 Single Master/Single Slave (with This MCU Acting as Master)

Figure 40.5 shows a single-master/single-slave RSPI system configuration example when this MCU is used as a master. In the single-master/single-slave configuration, the SSLx0 to SSLx3 output of this MCU (master) are not used. The SSL input of the SPI slave is fixed to the low level, and the SPI slave is maintained in a select state.*1
This MCU (master) drives the RSPCKx and MOSIx. The SPI slave drives the MISO.

Note 1. In the transfer format corresponding to the case where the SPCMDm.CPHA bit is 0, there are slave devices for which the SSL signal cannot be fixed to the active level. In situations where the SSL signal cannot be fixed, the SSLxi output of this MCU should be connected to the SSL input of the slave device.

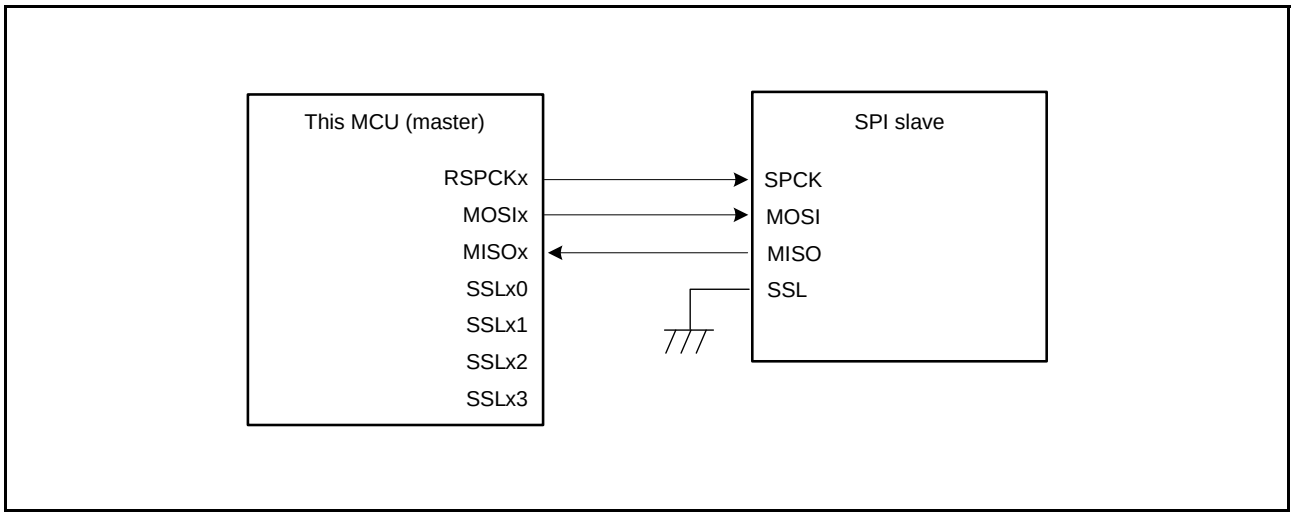


Figure 40.5 Single-Master/Single-Slave Configuration Example (This MCU = Master)

40.3.3.2 Single Master/Single Slave (with This MCU Acting as Slave)

Figure 40.6 shows a single-master/single-slave RSPI system configuration example when this MCU is used as a slave. When this MCU is to operate as a slave, the SSLx0 pin is used as SSL input. The SPI master drives the RSPCK and MOSI. This MCU (slave) drives the MISOx.*1

In the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, the SSLx0 input of this MCU (slave) is fixed to the low level, this MCU (slave) is maintained in a select state, and in this manner it is possible to execute serial transfer (Figure 40.7). In this configuration, the communication end interrupt and the communication end event are not generated.

Note 1. When SSLx0 is at the non-active level, the pin state is Hi-Z.

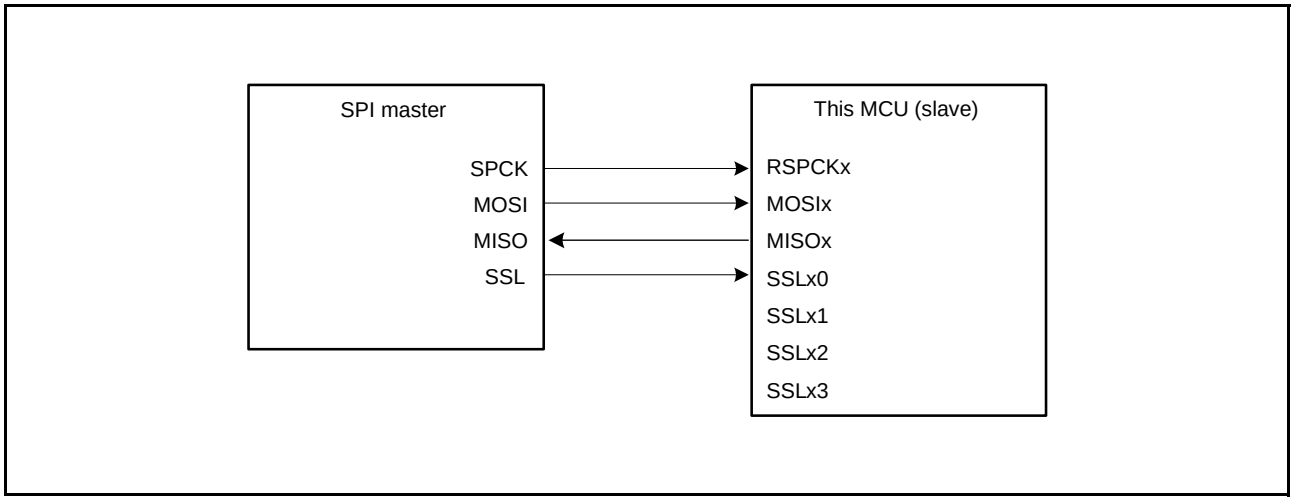


Figure 40.6 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 0)

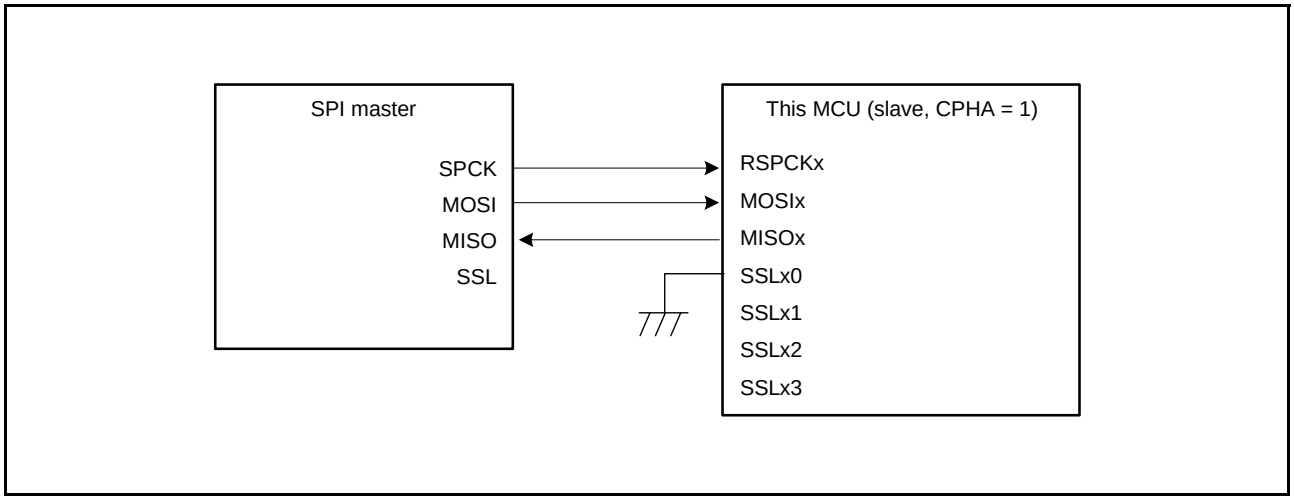


Figure 40.7 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 1)

40.3.3.3 Single Master/Multi-Slave (with This MCU Acting as Master)

Figure 40.8 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 40.8, the RSPI system is comprised of this MCU (master) and four slaves (SPI slave 0 to SPI slave 3).

The RSPCKx and MOSIx outputs of this MCU (master) are connected to the RSPCK and MOSI inputs of SPI slave 0 to SPI slave 3. The MISO outputs of SPI slave 0 to SPI slave 3 are all connected to the MISOx input of this MCU (master). SSLx0 to SSLx3 outputs of this MCU (master) are connected to the SSL inputs of SPI slave 0 to SPI slave 3, respectively.

This MCU (master) drives RSPCKx, MOSIx, and SSLx0 to SSLx3. Of the SPI slave 0 to SPI slave 3, the slave that receives low-level input into the SSL input drives MISO.

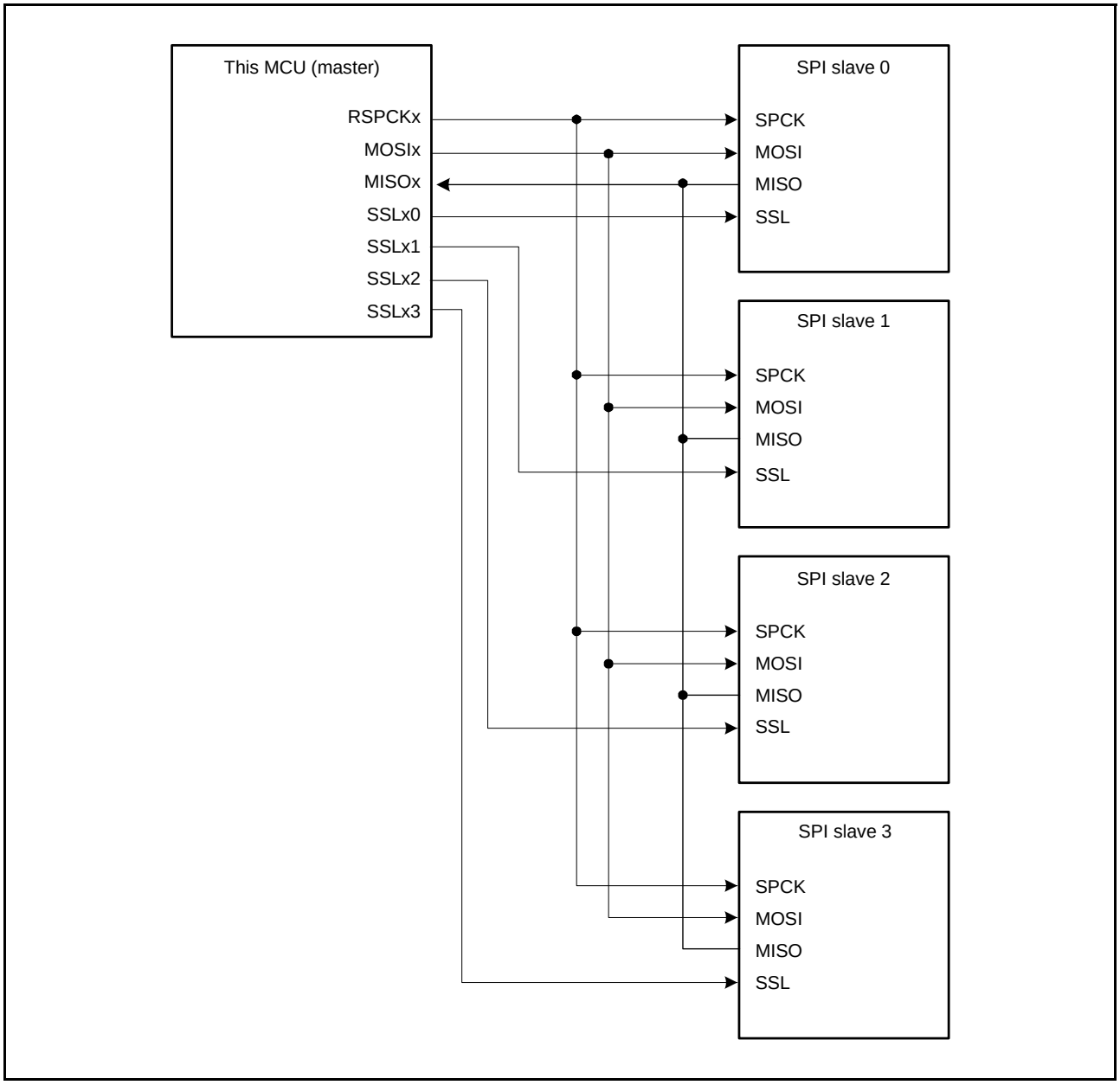


Figure 40.8 Single-Master/Multi-Slave Configuration Example (This MCU = Master)

40.3.3.4 Single Master/Multi-Slave (with This MCU Acting as Slave)

Figure 40.9 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a slave. In the example of Figure 40.9, the RSPI system is comprised of an SPI master and two MCUs (slave X and slave Y).
The SPCK and MOSI outputs of the SPI master are connected to the RSPCKx and MOSIx inputs of the MCUs (slave X and slave Y). The MISOx outputs of the MCUs (slave X and slave Y) are all connected to the MISO input of the SPI master. SSLX and SSLY outputs of the SPI master are connected to the SSLx0 inputs of the MCUs (slave X and slave Y), respectively.
The SPI master drives SPCK, MOSI, SSLX, and SSLY. Of the MCUs (slave X and slave Y), the slave that receives low-level input into the SSLx0 input drives MISOx.

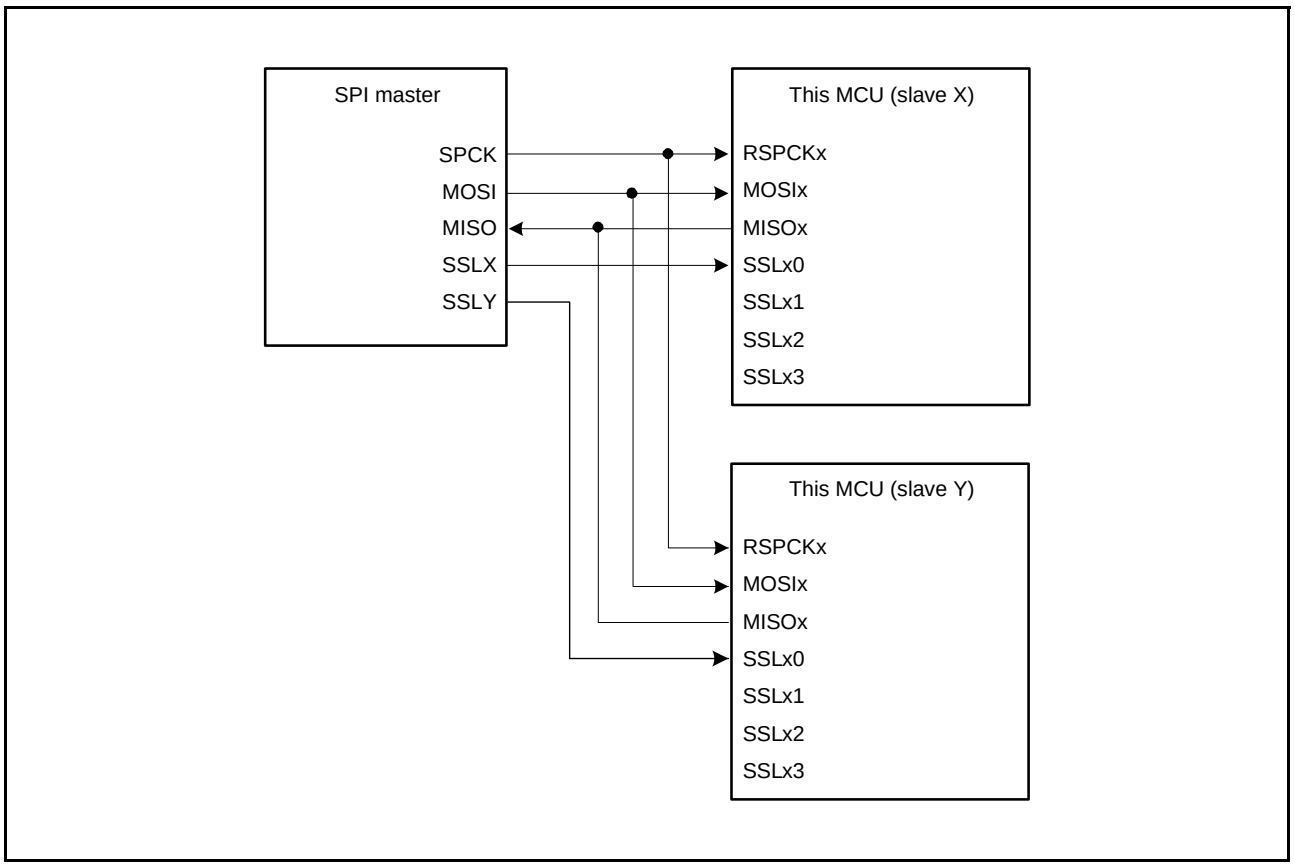


Figure 40.9 Single-Master/Multi-Slave Configuration Example (This MCU = Slave)

40.3.3.5 Multi-Master/Multi-Slave (with This MCU Acting as Master)

Figure 40.10 shows a multi-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 40.10, the RSPI system is comprised of two MCUs (master X and master Y) and two SPI slaves (SPI slave 1 and SPI slave 2).

The RSPCKx and MOSIx outputs of the MCUs (master X and master Y) are connected to the RSPCK and MOSI inputs of SPI slaves 1 and 2. The MISO outputs of SPI slaves 1 and 2 are connected to the MISOx inputs of the MCUs (master X and master Y). Any generic port Y output from this MCU (master X) is connected to the SSLx0 input of this MCU (master Y). Any generic port X output of this MCU (master Y) is connected to the SSLx0 input of this MCU (master X). The SSLx1 and SSLx2 outputs of the MCUs (master X and master Y) are connected to the SSL inputs of the SPI slaves 1 and 2. In this configuration example, since the system can be comprised solely of SSLx0 input, and SSLx1 and SSLx2 outputs for slave connections, the SSLx3 output of this MCU is not required.

This MCU drives RSPCKx, MOSIx, SSLx1, and SSLx2 when the SSLx0 input level is high. When the SSLx0 input level is low, this MCU detects a mode fault error, sets RSPCKx, MOSIx, SSLx1, and SSLx2 to Hi-Z, and releases the RSPI bus right to the other master. Of the SPI slaves 1 and 2, the slave that receives low-level input into the SSL input drives MISO.

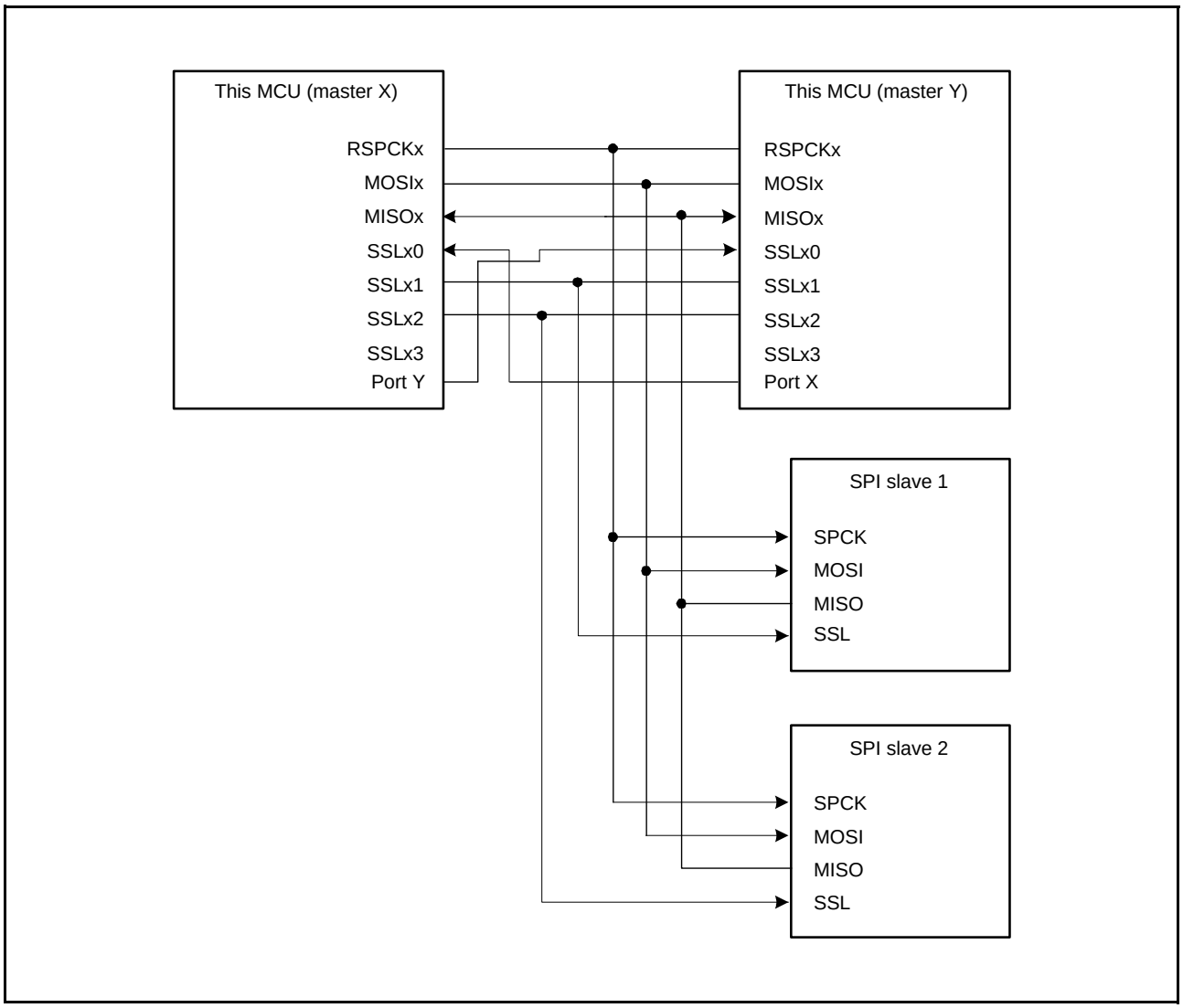


Figure 40.10 Multi-Master/Multi-Slave Configuration Example (This MCU = Master)

40.3.3.6 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Master)

Figure 40.11 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this MCU is used as a master. In the master (clock synchronous operation)/slave (clock synchronous operation) configuration, SSLx0 to SSLx3 of this MCU (master) are not used.
This MCU (master) drives the RSPCKx and MOSIx. The SPI slave drives the MISO.

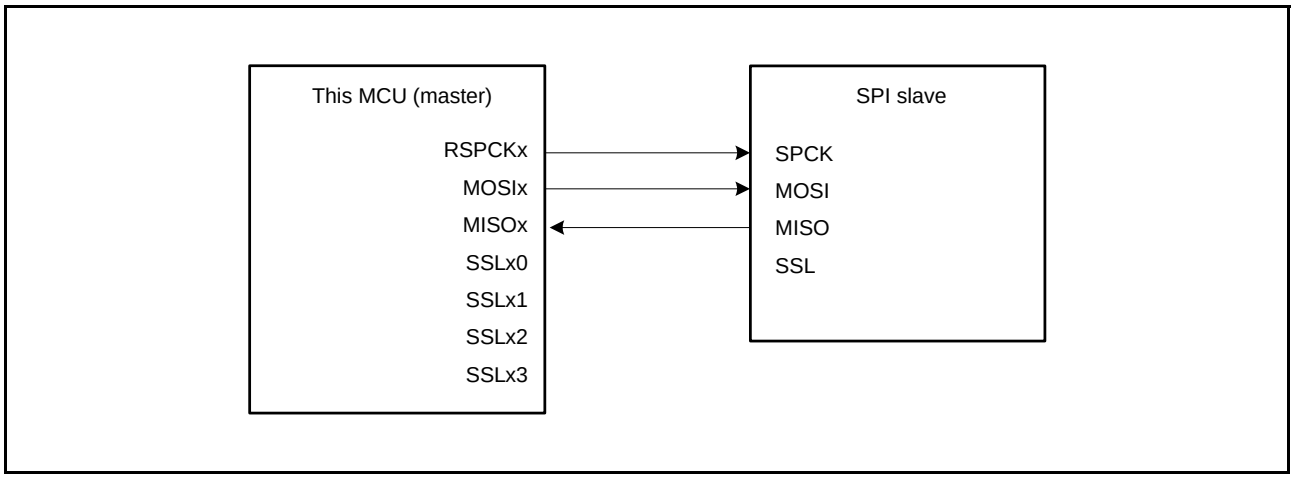


Figure 40.11 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Master)

40.3.3.7 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Slave)

Figure 40.12 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this MCU is used as a slave. When this MCU is to operate as a slave (clock synchronous operation), this MCU (slave) drives the MISOx and the SPI master drives the SPCK and MOSI. In addition, SSLx0 to SSLx3 of this MCU (slave) are not used.
Only in the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, this MCU (slave) can execute serial transfer.

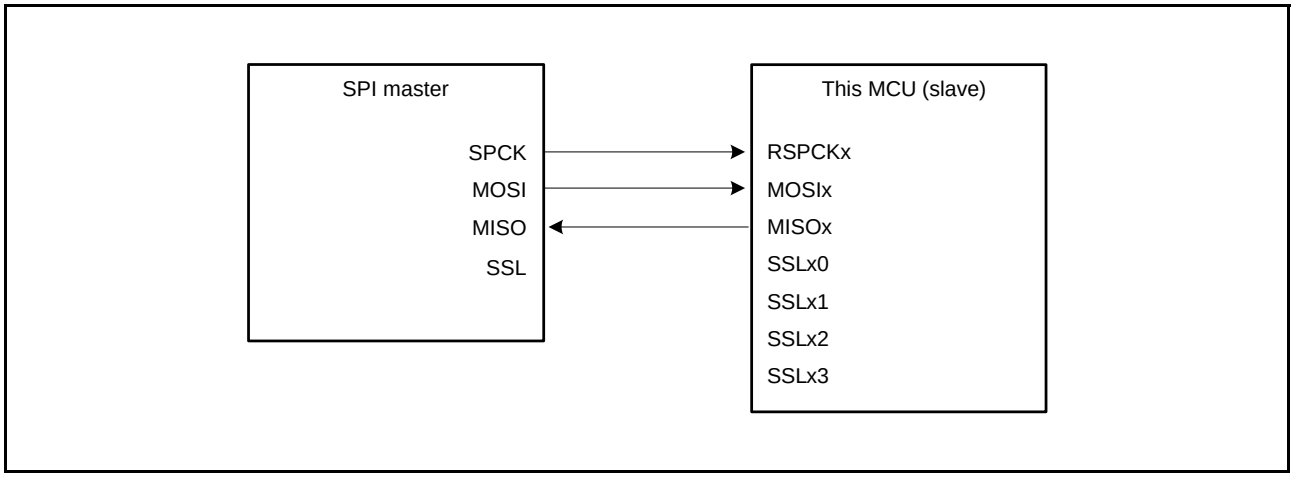


Figure 40.12 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Slave, CPHA = 1)

40.3.4 Data Format

The RSPI's data format depends on the settings in RSPI command register m (SPCMDm) (m = 0 to 7) and the parity enable bit in RSPI control register 2 (SPCR2.SPPE) and RSPI data control register 2 (SPDCR2). Regardless of whether the MSB or LSB is first, the RSPI treats the range from the LSB bit in the RSPI data register (SPDR) to the selected data length as transfer data.
The format of one frame of data before or after transfer is shown below.

(a) With Parity Disabled

When parity is disabled, transmission or reception of data proceeds with the length in bits selected in the RSPI data length setting bits in RSPI command register m (SPCMDm.SPB[3:0]).

(b) With Parity Enabled

When parity is enabled, transmission or reception of data proceeds with the length in bits selected in the RSPI data length setting bits in RSPI command register m (SPCMDm.SPB[3:0]). In this case, however, the last bit is a parity bit.

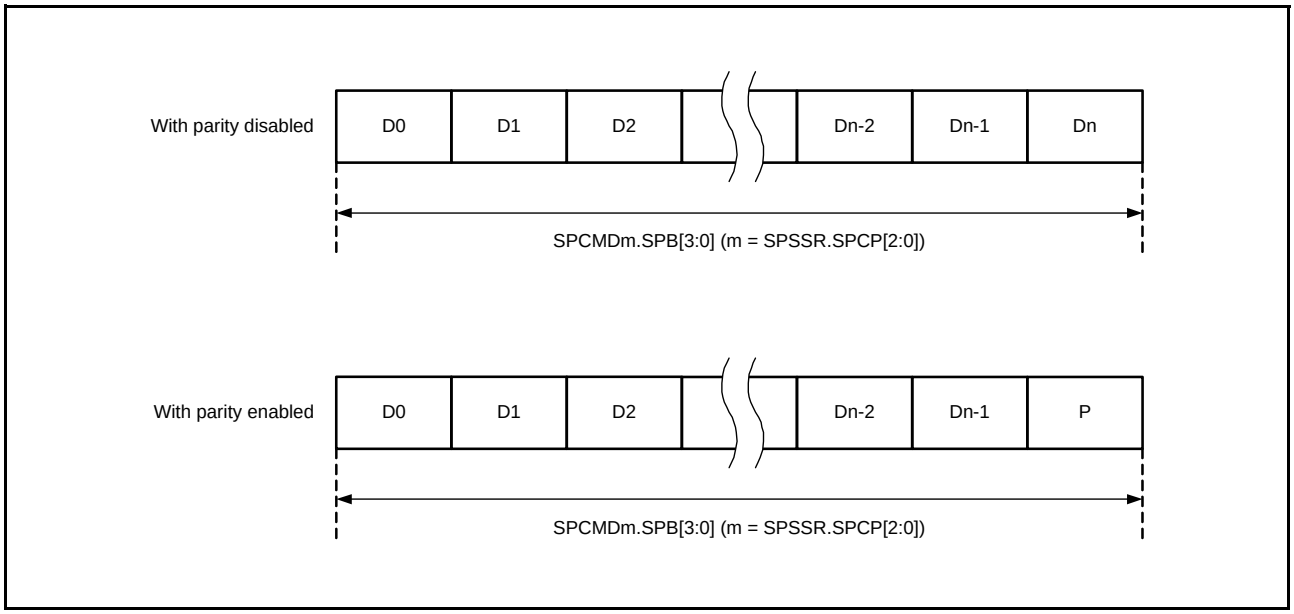


Figure 40.13 Outline of the Data Format (with Parity Disabled/Enabled)

40.3.4.1 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 40.14 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, an RSPI data length of 32 bits, and MSB first selected.
In transmission, bits T31 to T00 from the current stage of the transmit buffer are copied to the shift register. Data for transmission are shifted out from the shift register in order from T31, through T30, and so on to T00.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R31 to R00 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

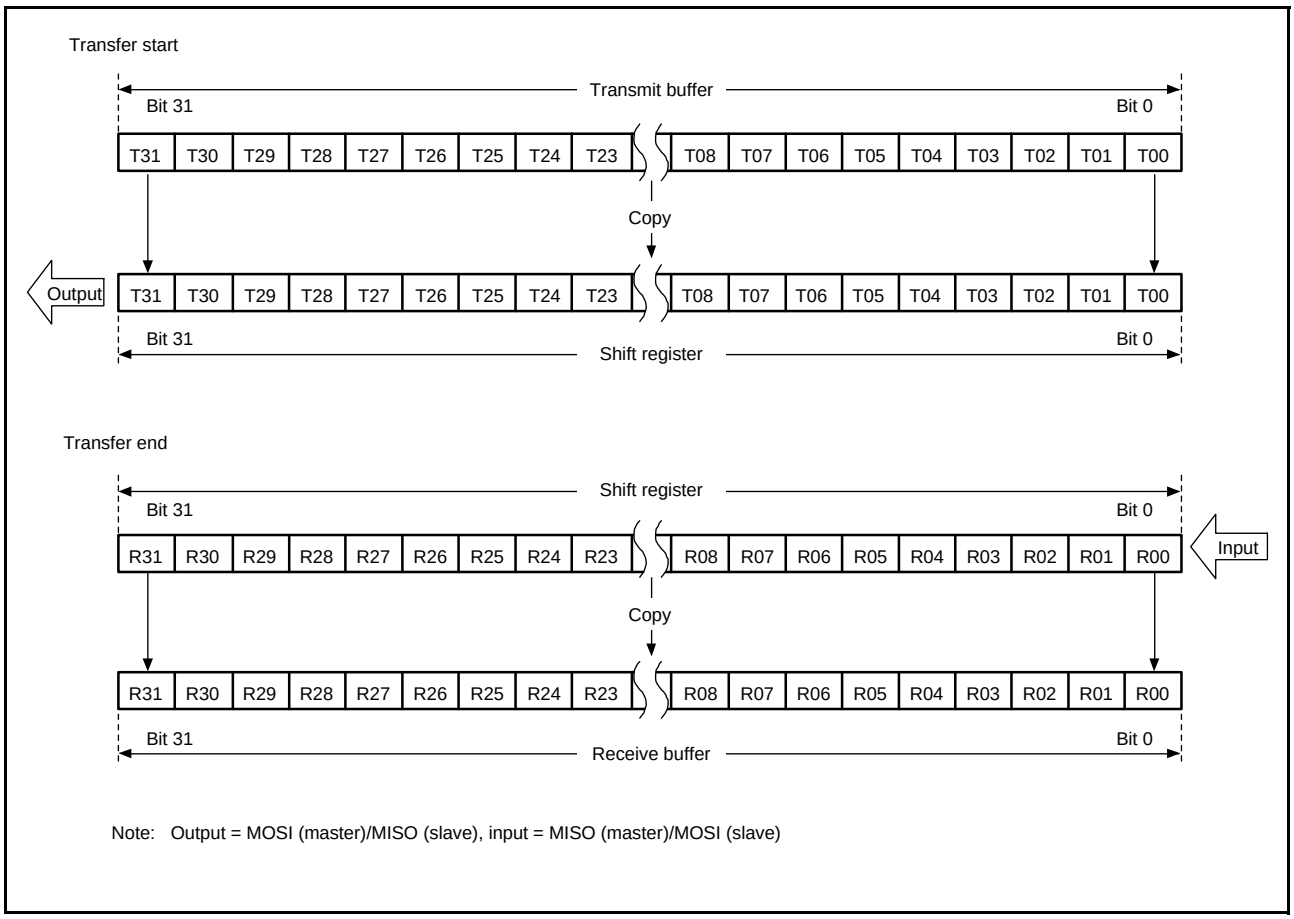


Figure 40.14 MSB First Transfer (32-Bit Data, Parity Disabled)

(2) MSB First Transfer (24-Bit Data)

Figure 40.15 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, 24 bits as the RSPI data length for an example that is not 32 bits, and MSB first selected.
In transmission, the lower 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, if the SPCR3.RXMD bit is 0, the upper 8 bits of the transmit buffer are stored in the upper 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being set in the upper 8 bits of the receive buffer. If the SPCR3.RXMD bit is 1, 00h is stored in the upper 8 bits of the receive buffer.

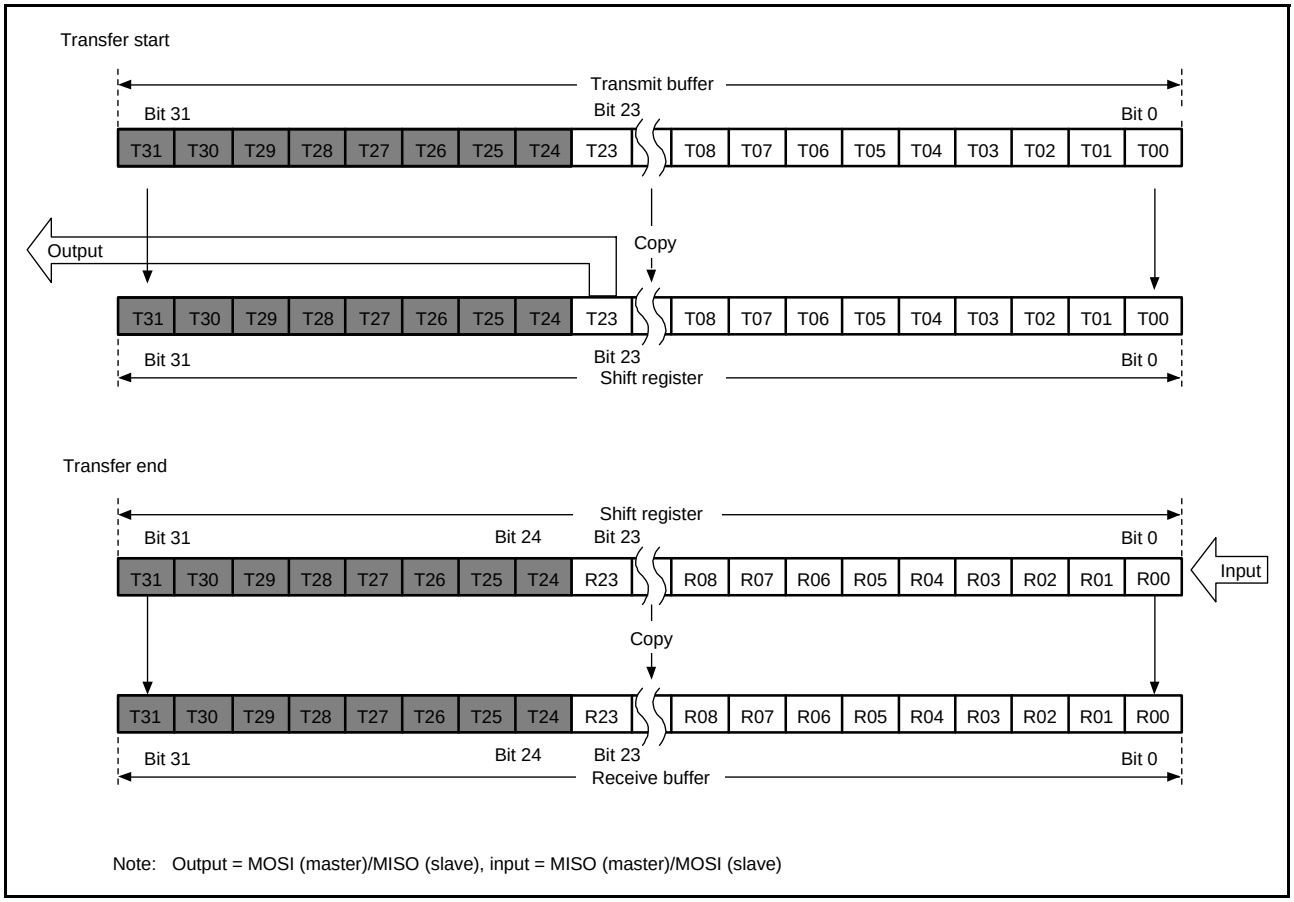


Figure 40.15 MSB First Transfer (24-Bit Data, Parity Disabled)

(3) LSB First Transfer (32-Bit Data)

Figure 40.16 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, an RSPI data length of 32 bits, and LSB first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are reordered bit by bit to obtain the order T00 to T31 for copying to the shift register. Data for transmission are shifted out from the shift register in order from T00, through T01, and so on to T31.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R00 to R31 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

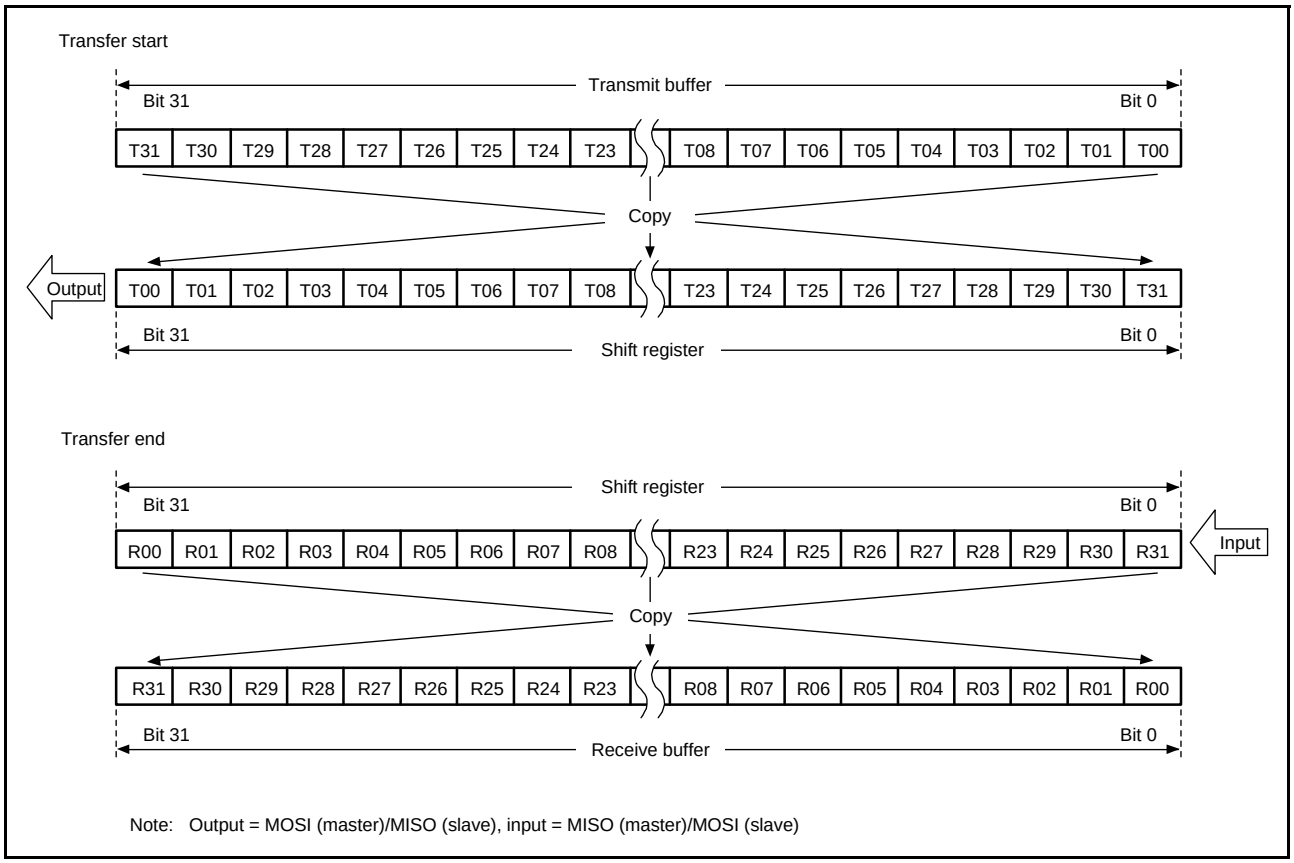


Figure 40.16 LSB First Transfer (32-Bit Data, Parity Disabled)

(4) LSB First Transfer (24-Bit Data)

Figure 40.17 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, 24 bits as the RSPI data length for an example that is not 32 bits, and LSB first selected.

In transmission, the lower 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, if the SPCR3.RXMD bit is 0, the upper 8 bits of the transmit buffer are stored in the upper 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being set in the upper 8 bits of the receive buffer. If the SPCR3.RXMD bit is 1, 00h is stored in the upper 8 bits of the receive buffer.

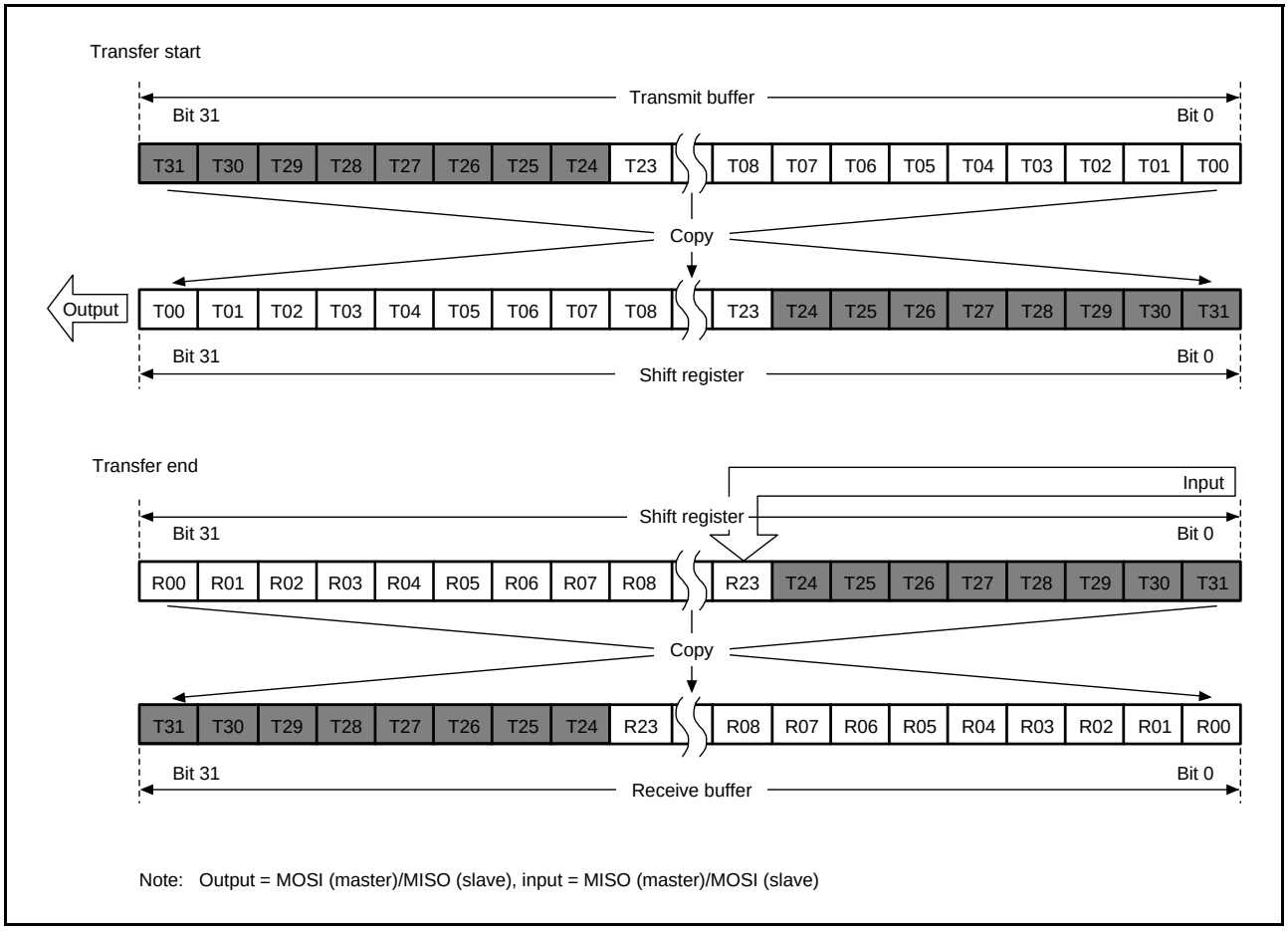


Figure 40.17 LSB First Transfer (24-Bit Data, Parity Disabled)

40.3.4.2 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 40.18 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, an RSPI data length of 32 bits, and MSB first selected.

In transmission, the value of the parity bit (P) is calculated from bits T31 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data are transmitted in the order T31, T30, …, T01, and P.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R31 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R31 to P are checked by judging the parity.

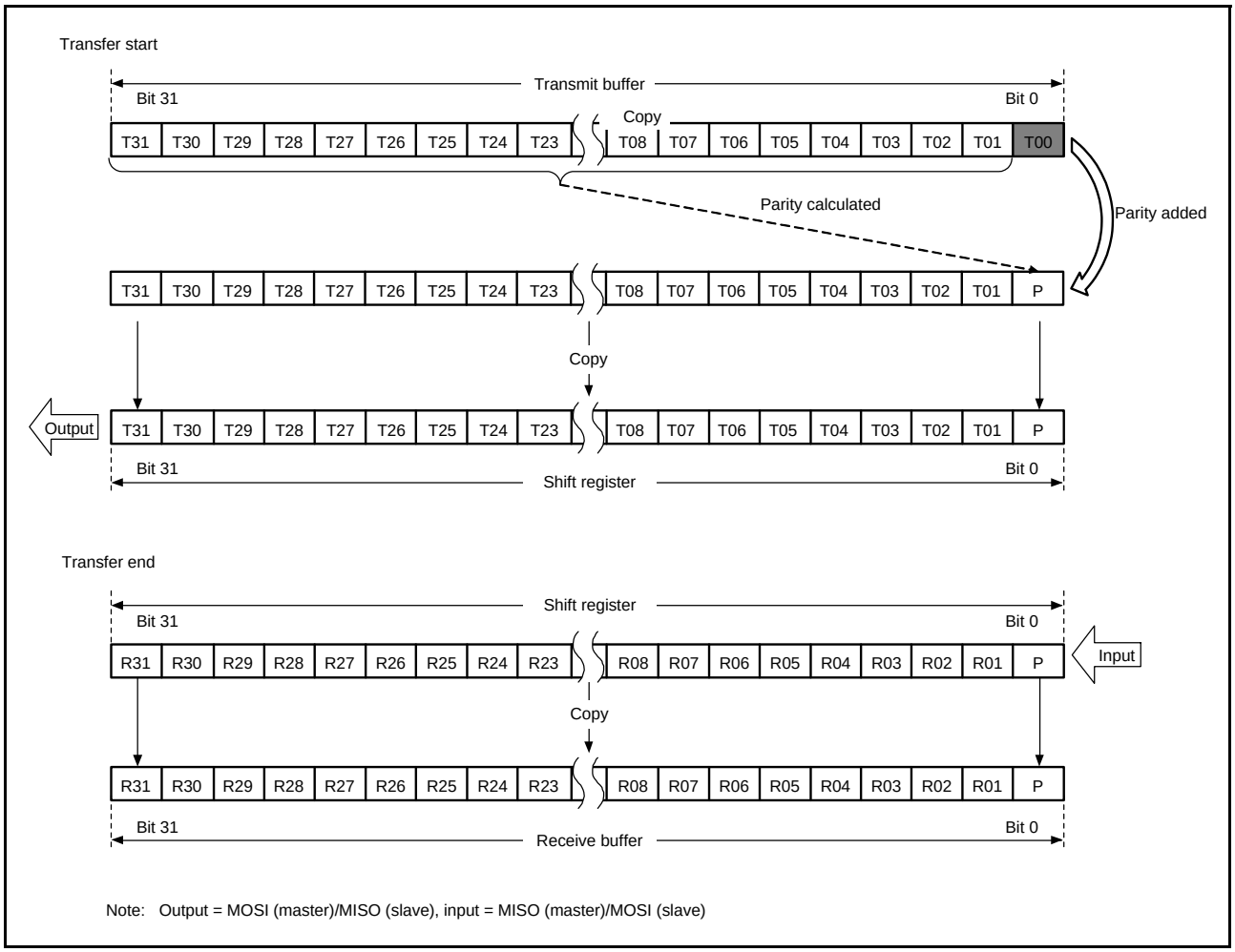


Figure 40.18 MSB First Transfer (32-Bit Data, Parity Enabled)

(2) MSB First Transfer (24-Bit Data)

Figure 40.19 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, 24 bits as the RSPI data length for an example that is not 32 bits, and MSB first selected.
In transmission, the value of the parity bit (P) is calculated from bits T23 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data are transmitted in the order T23, T22, …, T01, and P.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R23 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R23 to P are checked by judging the parity. At this time, if the SPCR3.RXMD bit is 0, the upper 8 bits of the transmit buffer are stored in the upper 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being set in the upper 8 bits of the receive buffer. If the SPCR3.RXMD bit is 1, 00h is stored in the upper 8 bits of the receive buffer.

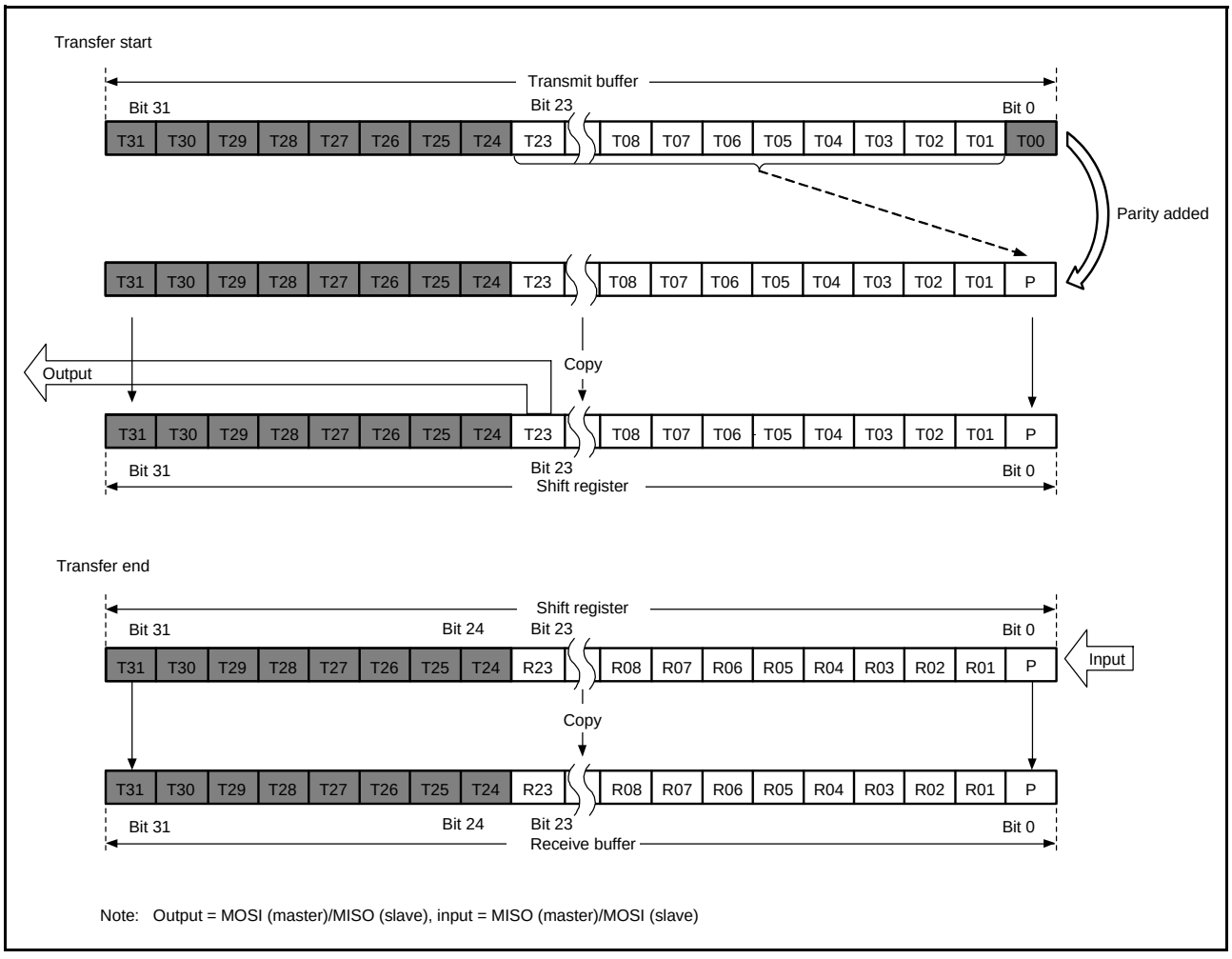


Figure 40.19 MSB First Transfer (24-Bit Data, Parity Enabled)

(3) LSB First Transfer (32-Bit Data)

Figure 40.20 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, an RSPI data length of 32 bits, and LSB first selected.

In transmission, the value of the parity bit (P) is calculated from bits T30 to T00. This replaces the final bit, T31, and the whole is copied to the shift register. Data are transmitted in the order T00, T01, …, T30, and P.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R00 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P are checked by judging the parity.

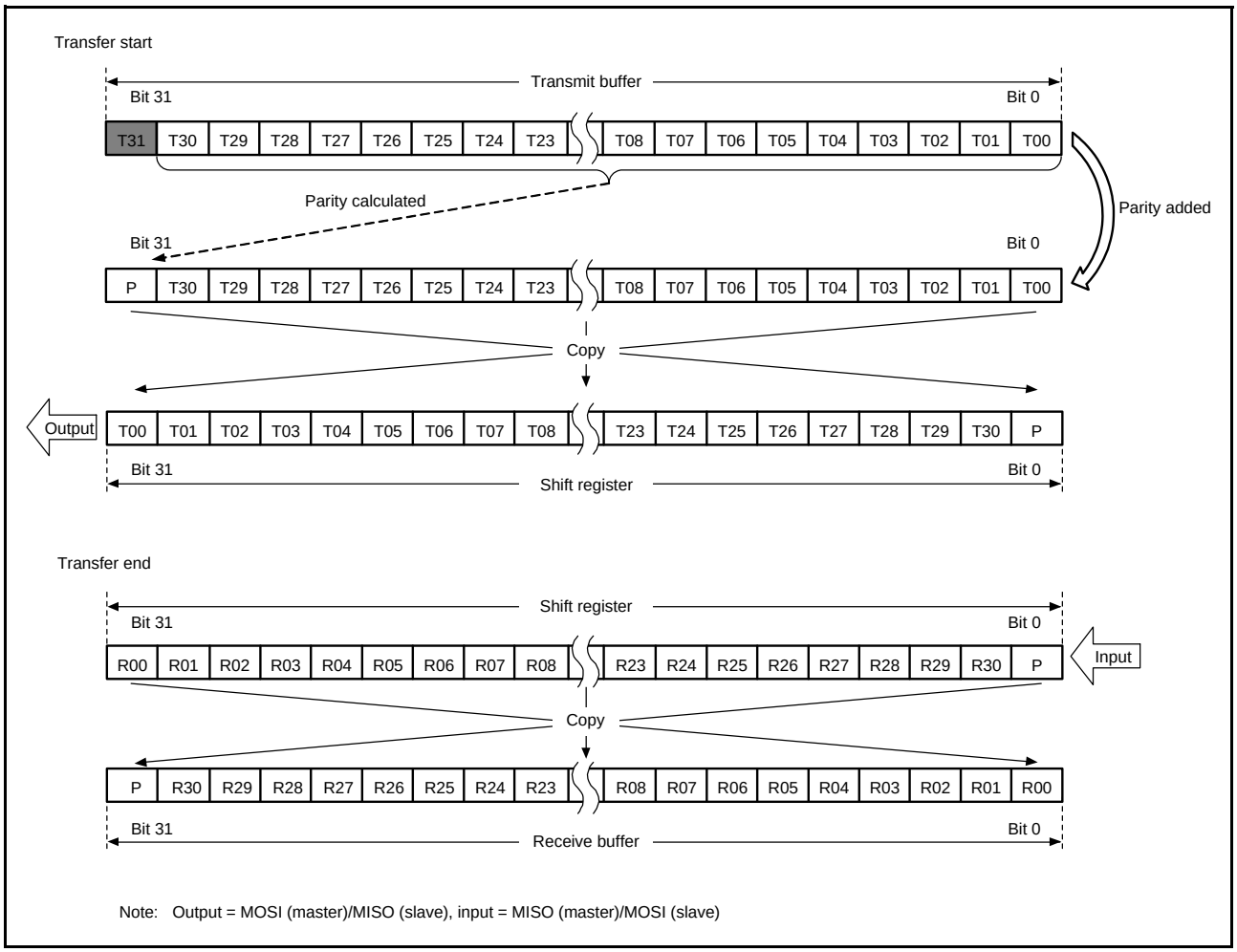


Figure 40.20 LSB First Transfer (32-Bit Data, Parity Enabled)

(4) LSB First Transfer (24-Bit Data)

Figure 40.21 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, 24 bits as the RSPI data length for an example that is not 32 bits, and LSB first selected.
In transmission, the value of the parity bit (P) is calculated from bits T22 to T00. This replaces the final bit, T23, and the whole is copied to the shift register. Data are transmitted in the order T00, T01, …, T22, and P.
In reception, received data are shifted in bit by bit through bit 8 of the shift register. When bits R00 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P are checked by judging the parity. At this time, if the SPCR3.RXMD bit is 0, the upper 8 bits of the transmit buffer are stored in the upper 8 bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being set in the upper 8 bits of the receive buffer. If the SPCR3.RXMD bit is 1, 00h is stored in the upper 8 bits of the receive buffer.

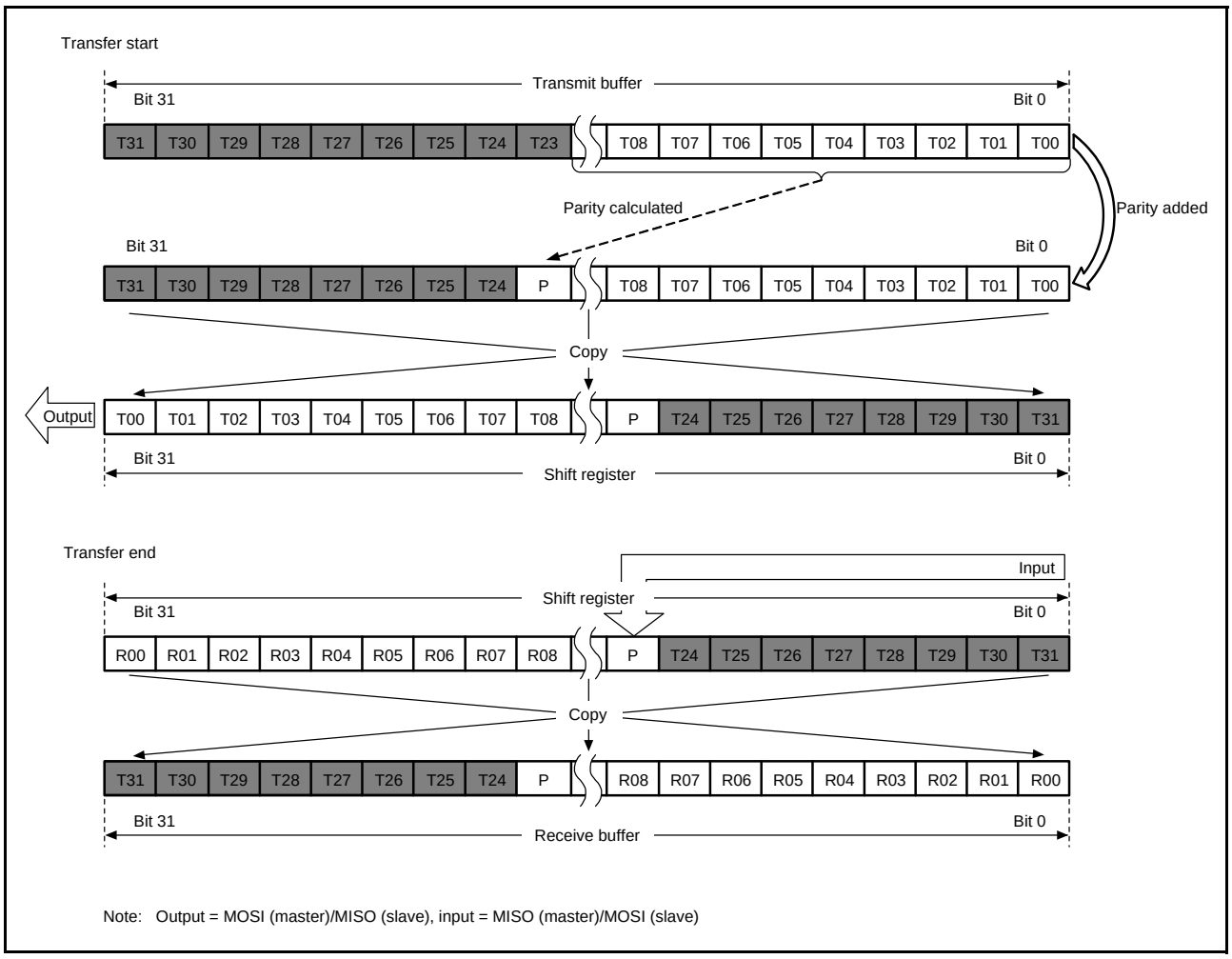


Figure 40.21 LSB First Transfer (24-Bit Data, Parity Enabled)

40.3.4.3 Byte Swap Transmission

When the SPDCR2.BYSW bit is 1 (byte swapping of SPDR data enabled), data bytes written in the transmit buffer (SPDR) will be swapped (in 8 bit units) when the data is transferred to the shift register. Figure 40.22 shows data transfer between the SPDR register and the shift register when data length is 32 bits.

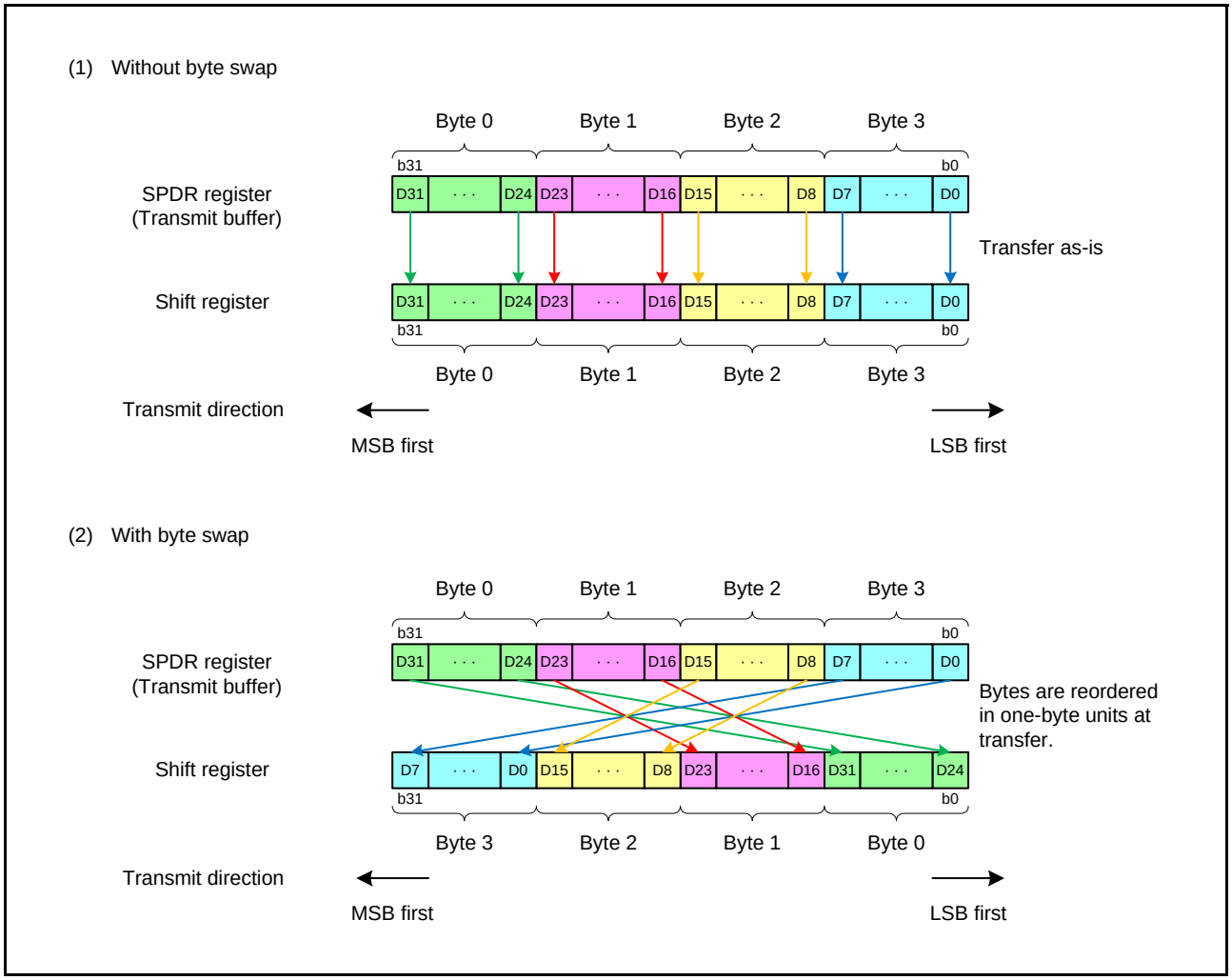


Figure 40.22 Transmit Data Handling in MSB/LSB First with Byte Swapping Enabled/Disabled

40.3.4.4 Byte Swap Reception

When the SPDCR2.BYSW bit is 1 (byte swapping of SPDR data enabled), data bytes in the shift register will be swapped (in 8 bit units) when the data is transferred to the receive buffer (SPDR). Figure 40.23 shows data transfer between the shift register and the SPDR register when data length is 32 bits.

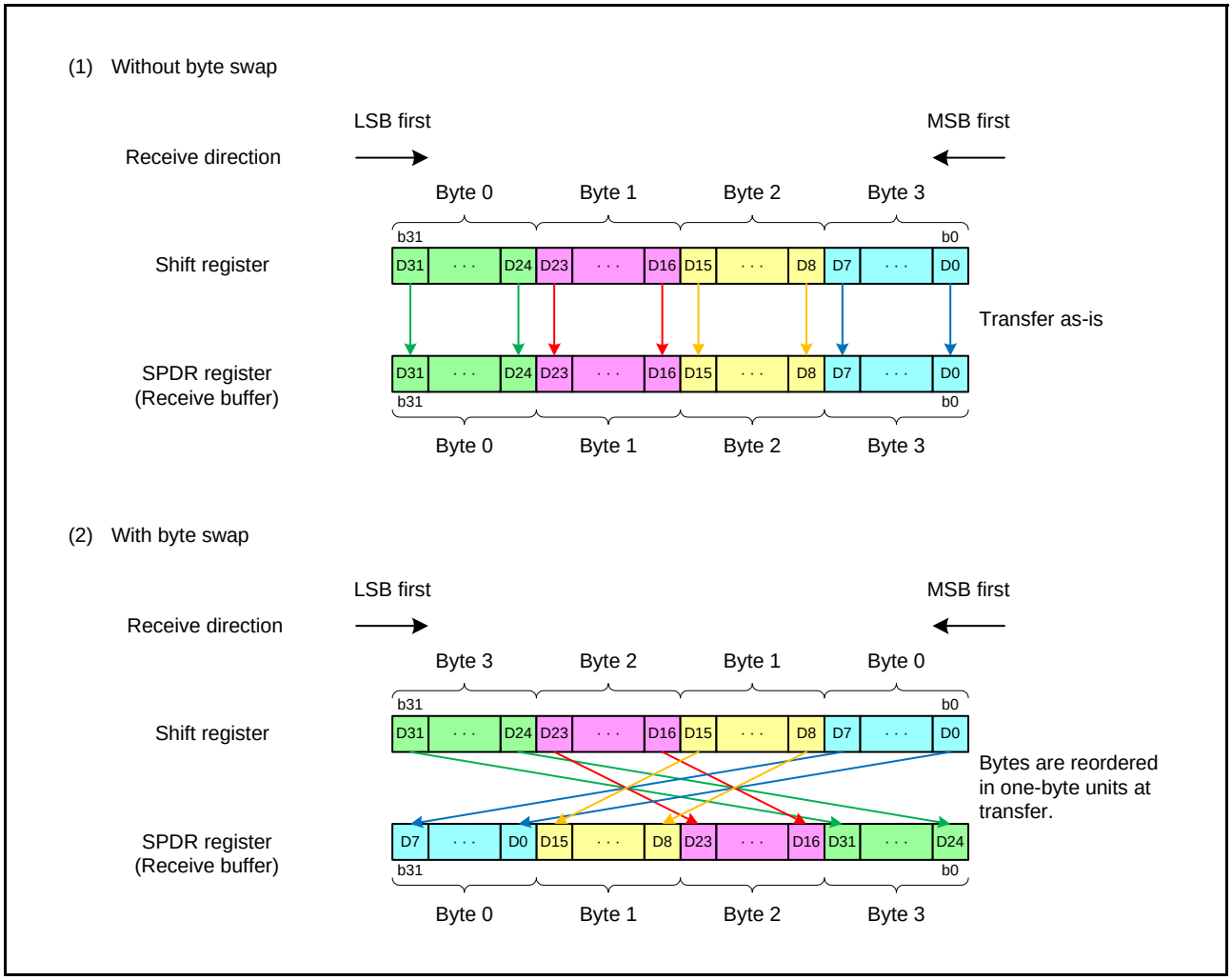


Figure 40.23 Receive Data Handling in MSB/LSB First with Byte Swapping Enabled/Disabled

40.3.5 Transfer Format

40.3.5.1 CPHA = 0

Figure 40.24 shows a sample transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 0. Note that clock synchronous operation (the SPCR.SPMS bit is 1) should not performed when the RSPI operates in slave mode (SPCR.MSTR = 0) and the CPHA bit is 0. In Figure 40.24, RSPCKx (CPOL = 0) indicates the RSPCKx signal waveform when the SPCMDm.CPOL bit is 0; RSPCKx (CPOL = 1) indicates the RSPCKx signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the RSPI settings. For details, refer to section 40.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 0, the driving of valid data to the MOSIx and MISOx signals commences at an SSLxi signal assertion timing. The first RSPCKx signal change timing that occurs after the SSLxi signal assertion becomes the first transfer data fetch timing. After this timing, data is sampled at every 1 RSPCK cycle. The change timing for MOSIx and MISOx signals is 1/2 RSPCK cycles after the transfer data fetch timing. The CPOL bit setting does not affect the RSPCKx signal operation timing; it only affects the signal polarity.

t1 denotes a period from an SSLxi signal assertion to RSPCKx oscillation (RSPCK delay). t2 denotes a period from the termination of RSPCKx oscillation to an SSLxi signal negation (SSL negation delay). t3 denotes a period in which SSLxi signal assertion is suppressed for the next transfer after the end of serial transfer (next-access delay). t1, t2, and t3 are controlled by a master device running on the RSPI system. For a description of t1, t2, and t3 when the RSPI of this MCU is in master mode, refer to section 40.3.12.1, Master Mode Operation.

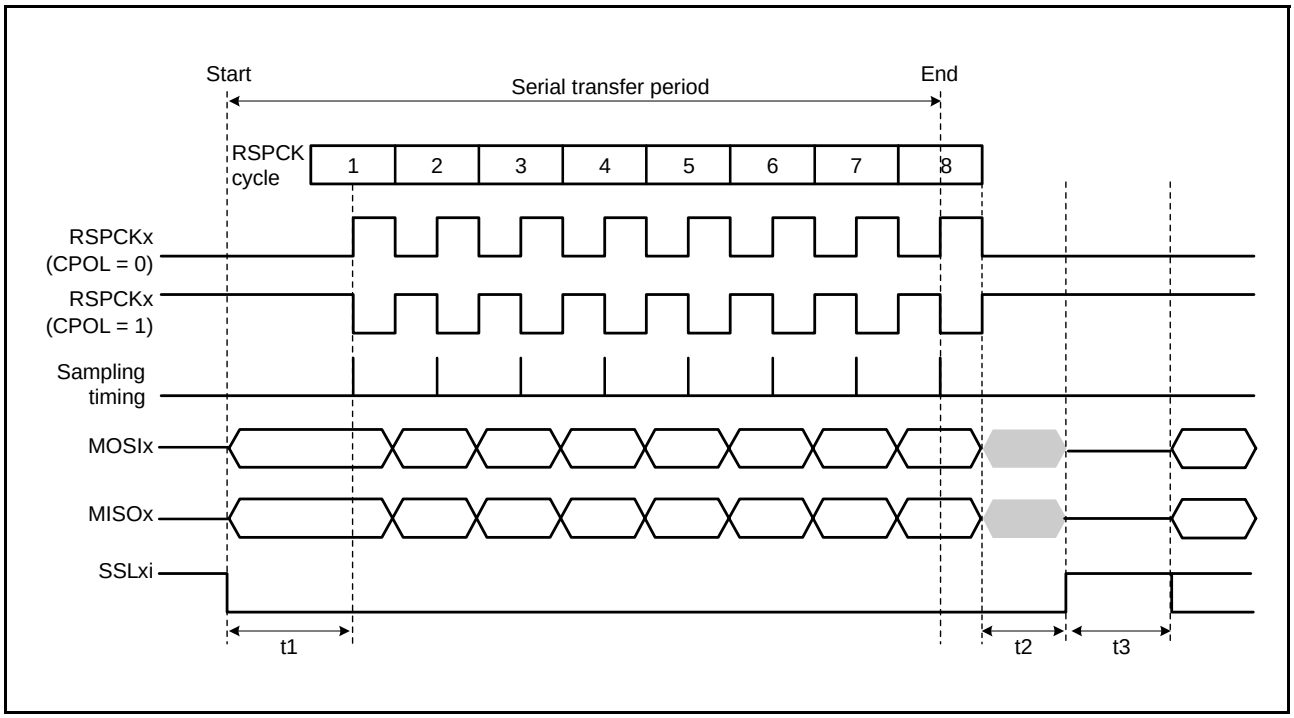


Figure 40.24 RSPI Transfer Format (CPHA = 0)

40.3.5.2 CPHA = 1

Figure 40.25 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 SSLxi signals are not used, and only the three signals RSPCKx, MOSIx, and MISOx handle communications. In Figure 40.25, RSPCK (CPOL = 0) indicates the RSPCKx signal waveform when the SPCMDm.CPOL bit is 0; RSPCKx (CPOL = 1) indicates the RSPCKx signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the RSPI mode (master or slave). For details, refer to section 40.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 1, the driving of invalid data to the MISOx signal commences at an SSLxi signal assertion timing. The output of valid data to the MOSIx and MISOx signals commences at the first RSPCKx signal change timing that occurs after the SSLxi signal assertion. After this timing, data is updated at every 1 RSPCK cycle. The transfer data fetch timing is 1/2 RSPCK cycles after the data update timing. The SPCMDm.CPOL bit setting does not affect the RSPCKx signal operation timing; it only affects the signal polarity.

t1, t2, and t3 are the same as those in the case of CPHA = 0. For a description of t1, t2, and t3 when the RSPI of this MCU is in master mode, refer to section 40.3.12.1, Master Mode Operation.

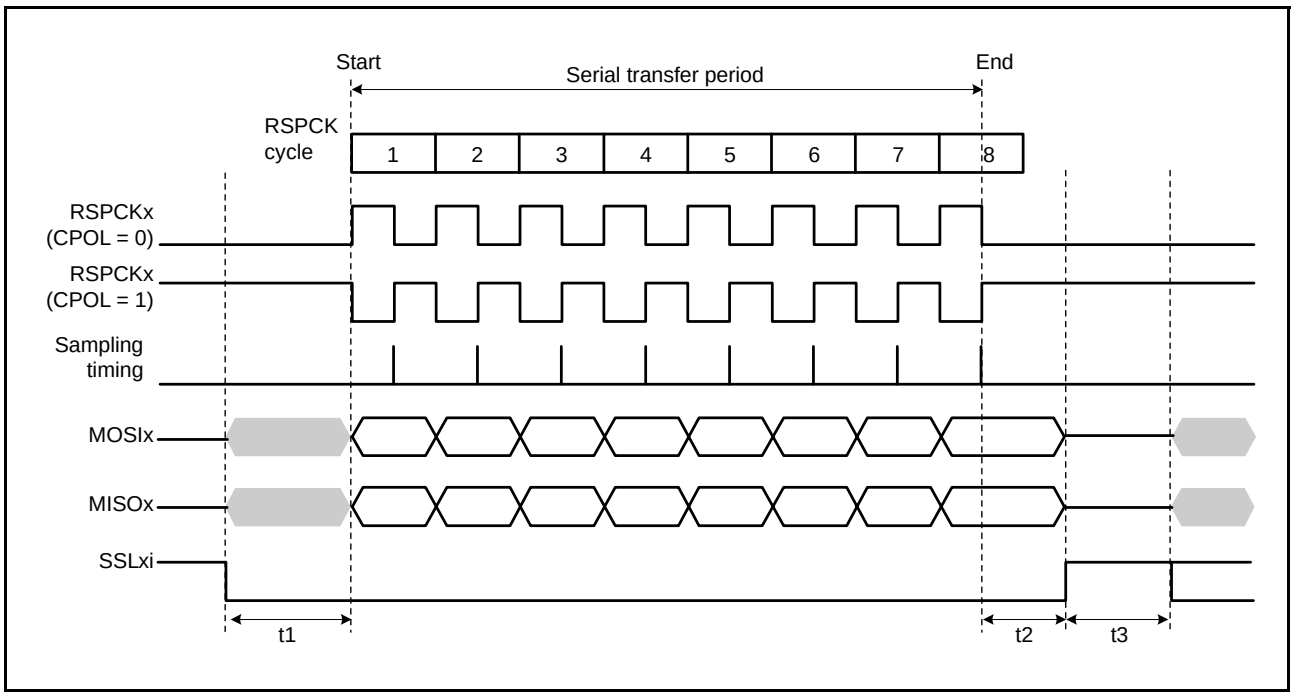


Figure 40.25 RSPI Transfer Format (CPHA = 1)

40.3.6 Communications Operating Mode

Full-duplex communications, transmit-only simplex communications, or receive-only simplex communications can be selected by the SPCR.TXMD and SPCR3.RXMD bits.
'SPDR access' shown in Figure 40.26 and Figure 40.27 indicate the condition of access to the SPDR register, where 'W' denotes a write cycle.

40.3.6.1 Full-Duplex Communications (SPCR.TXMD = 0, SPCR3.RXMD = 0)

Figure 40.26 shows an example of operation when the SPCR.TXMD bit is set to 0 and the SPCR3.RXMD bit is set to 0. In the example in Figure 40.26, 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 RSPCKx waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

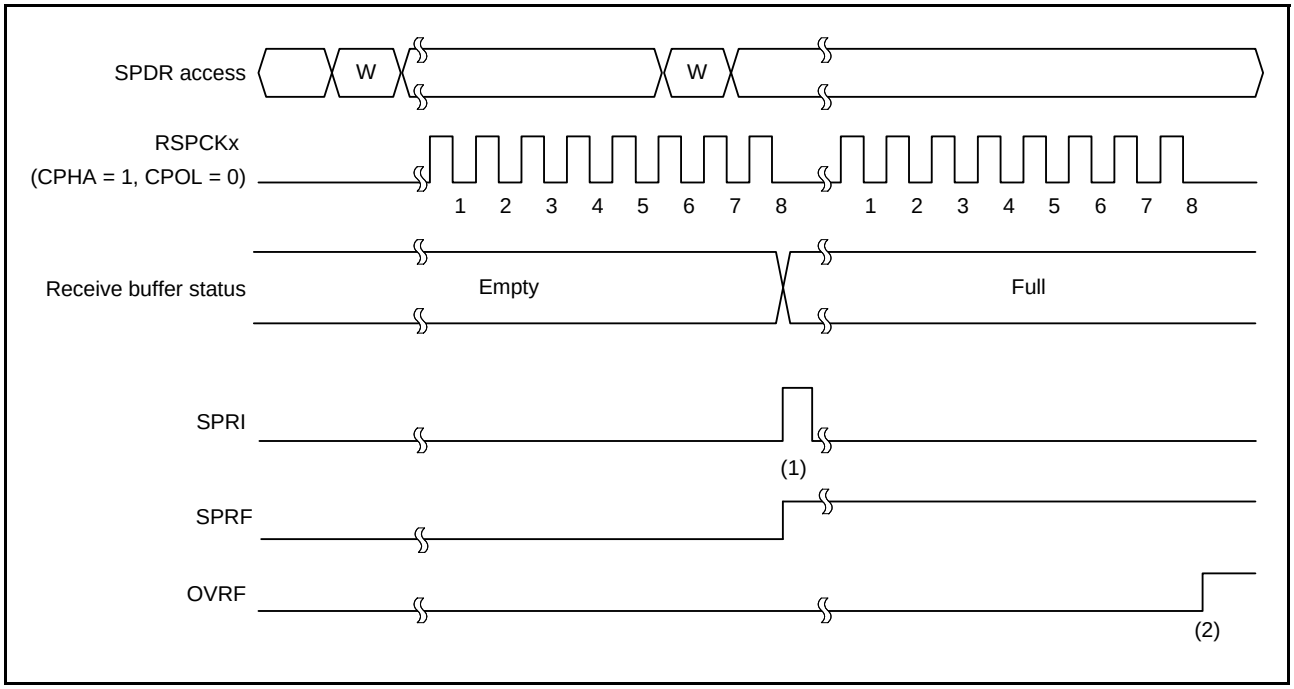


Figure 40.26 Operation Example of SPCR.TXMD = 0 and SPCR3.RXMD = 0

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 the SPDR register empty, the RSPI generates a receive buffer full interrupt request (SPRI) (sets the SPSR.SPRF flag to 1) and copies the received data in the shift register to the receive buffer.

(2) When a serial transfer ends with the receive buffer of the SPDR register holding data that was received in the previous serial transfer, the RSPI sets the SPSR.OVRF flag to 1 and discards the received data in the shift register.

When full-duplex communications (SPCR.TXMD = 0, SPCR3.RXMD = 0) is selected, reception occurs simultaneously with transmit operations. As such, the SPRF and OVRF flags in the SPSR register become 1 at the timing described in (1) and (2), respectively, according to the state of the receive buffer.

40.3.6.2 Transmit-only Simplex Communications (SPCR.TXMD = 1, SPCR3.RXMD = 0)

Figure 40.27 shows an example of operation when the SPCR.TXMD bit is set to 1 and the SPCR3.RXMD bit is set to 0. In the example in Figure 40.27, 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 RSPCKx waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

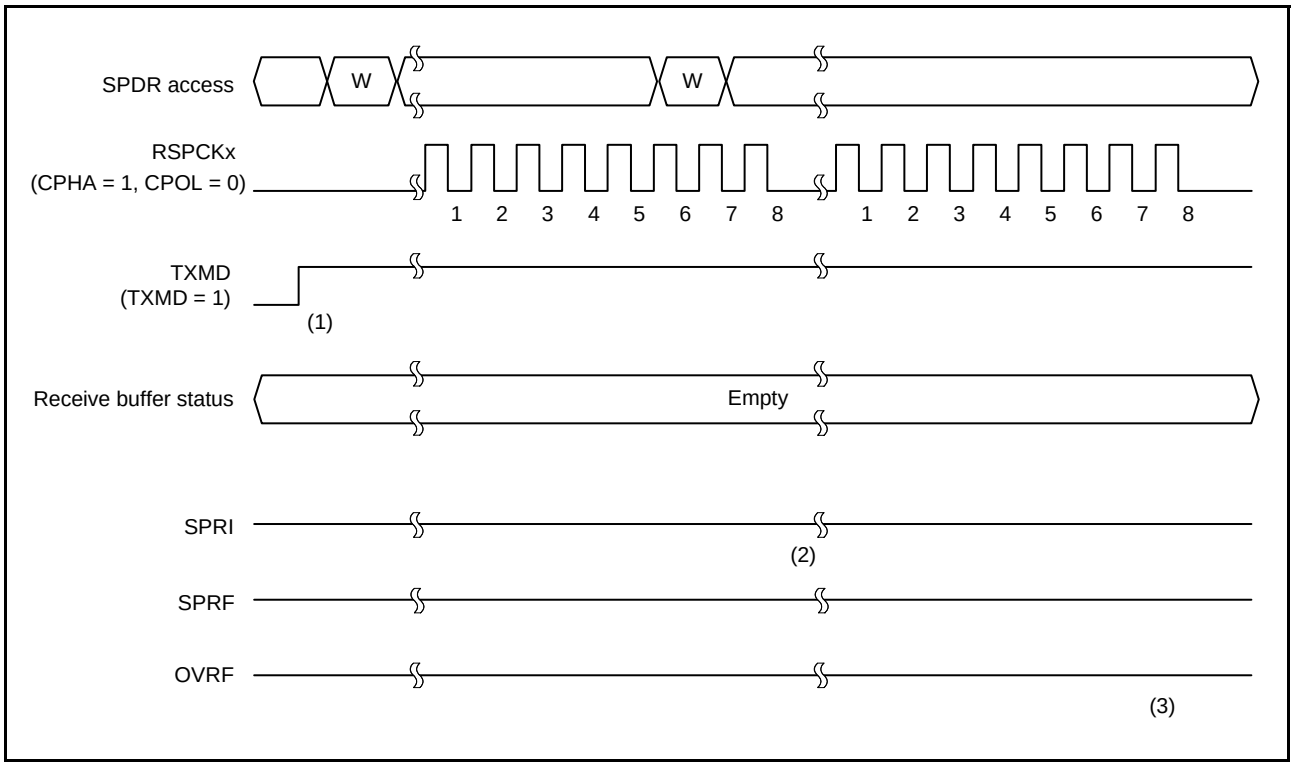


Figure 40.27 Operation Example of SPCR.TXMD = 1 and SPCR3.RXMD = 0

The operation of the flags at timings shown in steps (1) to (3) in the figure is described below.

(1) Make sure there is no data left in the receive buffer and the SPSR.SPRF, OVRF flags are 0 before entering the mode of transmit-only simplex communications (SPCR.TXMD = 1, SPCR3.RXMD = 0).

(2) When a serial transfer ends with the receive buffer of the SPDR register empty, if the mode of transmit-only simplex communications is selected (SPCR.TXMD = 1, SPCR3.RXMD = 0), the SPRF flag remains 0 and the RSPI does not copy the data from the shift register to the receive buffer.

(3) Since the receive buffer of the SPDR register 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-only simplex communications (SPCR.TXMD = 1, SPCR3.RXMD = 0), the RSPI transmits data but does not receive data. Therefore, the SPSR.SPRF, OVRF flags remain 0 at the timings of (1) to (3).

40.3.6.3 Receive-only Simplex Communications (SPCR3.RXMD = 0)

The receive-only simplex communications are valid only when the SPCR.MSTR bit is 0 (slave mode).
Figure 40.28 shows an example of operation when the SPCR3.RXMD bit is set to 1. In the example in Figure 40.28, 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 RSPCKx waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

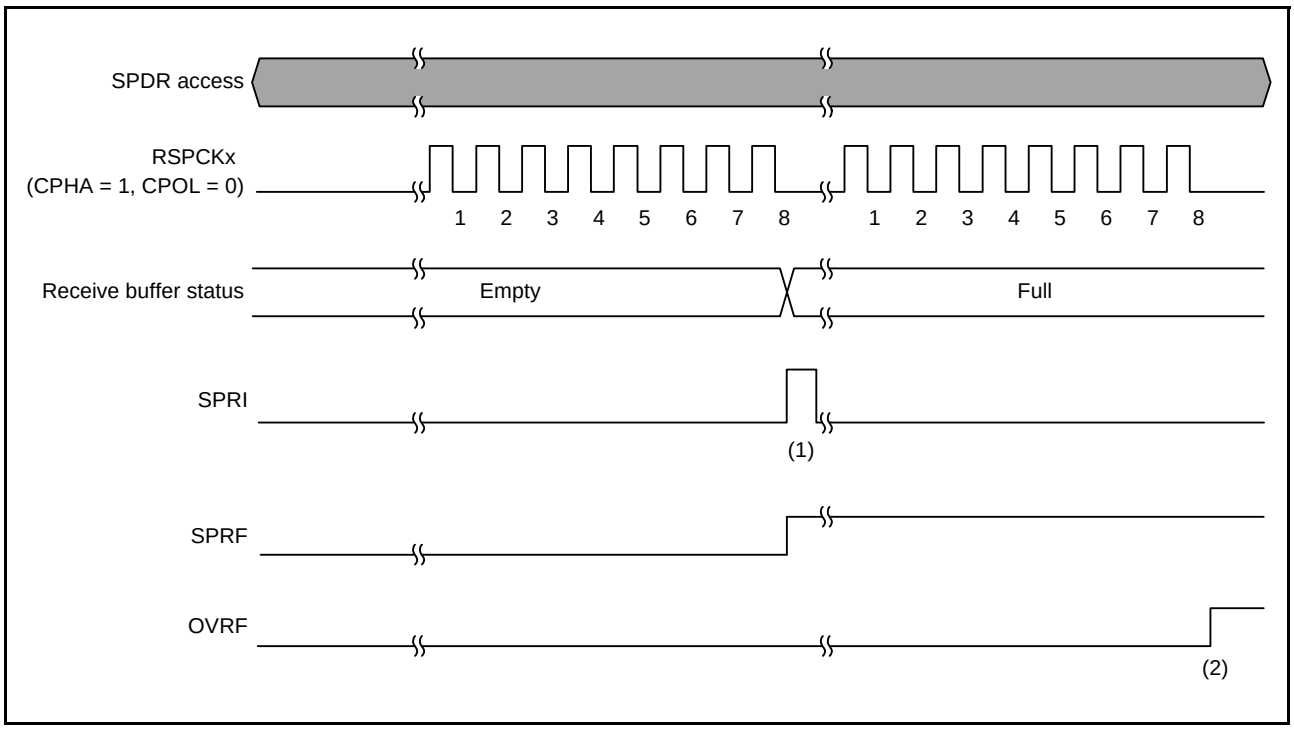


Figure 40.28 Operation Example of SPCR3.RXMD = 1

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 the SPDR register empty, the RSPI generates a receive buffer full interrupt request (SPRI) (sets the SPSR.SPRF flag to 1) and copies the received data in the shift register to the receive buffer.

(2) When a serial transfer ends with the receive buffer of the SPDR register 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.

40.3.7 Transmit Buffer Empty/Receive Buffer Full Interrupts

Figure 40.29 shows an example of operation of the transmit buffer empty interrupt (SPTI) and the receive buffer full interrupt (SPRI). 'SPDR access' shown in Figure 40.29 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 40.29, the RSPI performs an 8-bit serial transfer in which the SPCR.TXMD bit is 0, the SPCR3.RXMD 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 RSPCKx waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

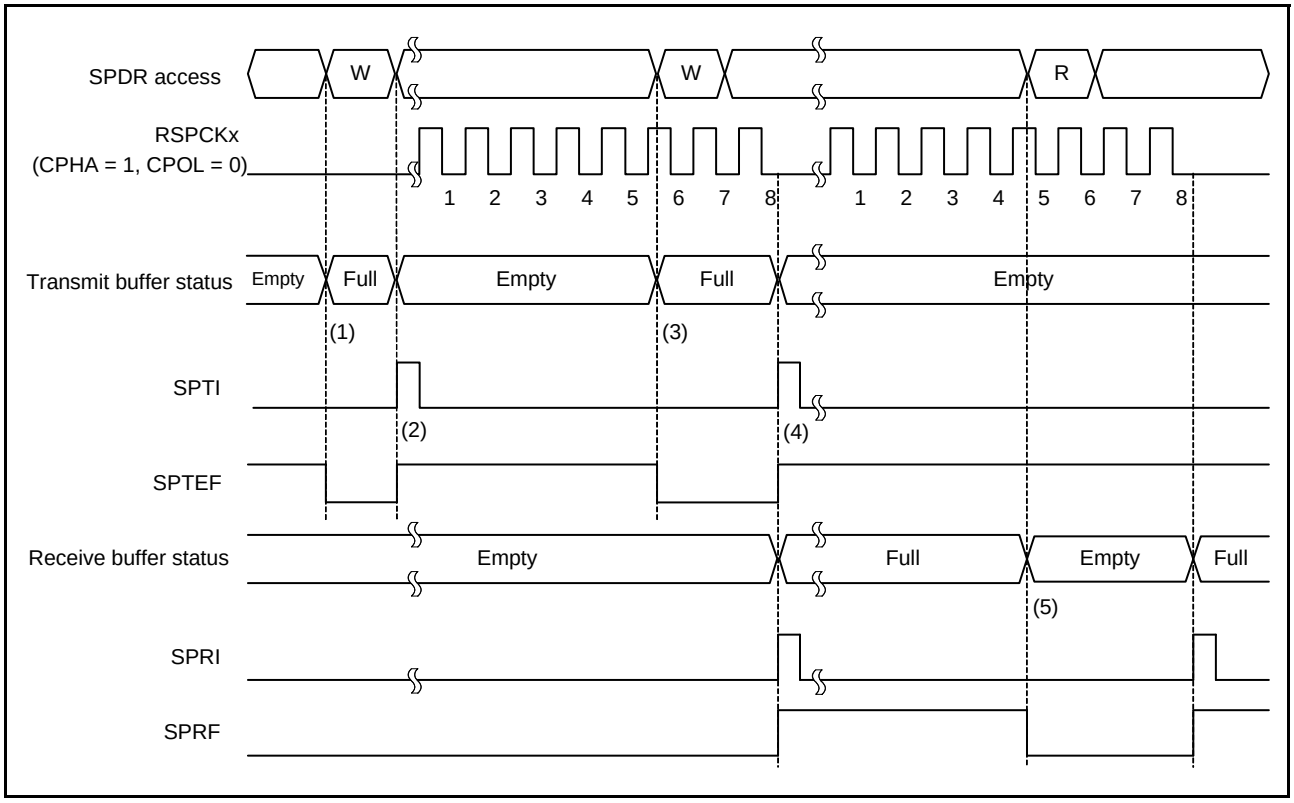


Figure 40.29 Operation Example of SPTI and SPRI Interrupts

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 the SPDR register when the transmit buffer of the SPDR register is empty (data for the next transfer is not set), the RSPI writes data to the transmit buffer and sets the SPSR.SPTEF flag to 0.

(2) If the shift register is empty, the RSPI copies the data from the transmit buffer to the shift register and generates a transmit buffer empty interrupt request (SPTI) and sets the SPSR.SPTEF flag to 1. How a serial transfer is started depends on the mode of the RSPI. For details, refer to section 40.3.12, SPI Operation, and section 40.3.13, Clock Synchronous Operation.

(3) When transmit data is written to the SPDR register in the transmit buffer empty interrupt routine or in the transmit buffer empty detecting process by polling the SPTEF flag, the data is transferred to the transmit buffer and the SPSR.SPTEF flag becomes 0. Because the data being transmitted is stored in the shift register, the RSPI does not copy the data from the transmit buffer to the shift register.

(4) When the serial transfer ends with the receive buffer of the SPDR register being empty, the RSPI copies the receive data from the shift register to the receive buffer, generates a receive buffer full interrupt request (SPRI), and sets the SPSR.SPRF flag to 1. Since the shift register becomes empty upon completion of serial transfer, when the transmit buffer had been full before the serial transfer ended, the RSPI sets the SPSR.SPTEF flag to 1 and copies the data from the transmit buffer to the shift register. Even when received data is not copied from the shift register to the receive buffer in an overrun error status, upon completion of the serial transfer, the RSPI determines that the shift register is empty, thus data transfer from the transmit buffer to the shift register is enabled.

(5) When the SPDR register is read in the receive buffer full interrupt routine or in the receive buffer full detecting process by polling the SPRF flag, the receive data can be read. When the receive data is read, the SPRF flag becomes 0.

If transmit data is written to the SPDR register while the transmit buffer holds data that has not yet been transmitted (the SPTEF flag is 0), the RSPI does not update the data in the transmit buffer. Transmit data should be written to the SPDR register in the transmit buffer empty interrupt request routine or in the transmit buffer empty detecting process by polling the SPTEF flag. To use a transmit buffer empty interrupt, set the SPTIE bit in SPCR to 1.
When setting the SPCR.SPE bit to 0 (RSPI disabled), the SPCR.SPTIE bit should also be set to 0. Otherwise (if the SPCR.SPE bit is 0 and the SPCR.SPTIE is 1), a transmit buffer empty interrupt request will occur.
When serial transfer ends with the receive buffer being full (the SPRF flag is 1), the RSPI does not copy data from the shift register to the receive buffer, and detects an overrun error (refer to section 40.3.10, 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 receive buffer full interrupt, set the SPCR.SPRIE bit to 1.
Transmit and receive interrupts or the corresponding IRn.IR flags (where n is the interrupt vector number) in the ICU can be used to confirm the states of the transmit and receive buffers. Refer to section 15, Interrupt Controller (ICUE), for the interrupt vector numbers. The status of the transmit and receive buffers can be also confirmed by the SPTEF and SPRF flags.

40.3.8 Idle Interrupt

When the SPSR.IDLNF flag becomes 0 while the SPCR2.SPIIE bit is 1, an idle interrupt request (SPII) is generated.
In master mode, the IDLNF flag is 0 before transmission. Therefore, write data in the transmit buffer and set the SPIIE bit to 1 after the IDLNF flag becomes 1 so that an idle interrupt is not generated at this time. When the SSLx0 signal is negated after the transmission is completed and the next data is not supplied until next-access delay time (t3) elapses, the IDLNF flag becomes 0.

40.3.9 Communication End Interrupt

When the SPSR.SPCF flag becomes 1 while the SPCR3.SPCIE bit is 1 or when the SPCIE bit is set to 1 while the SPCR.SPE bit is 1 and SPCF flag is 1, a communication end interrupt request (SPCI) is generated.
The conditions for setting the SPCF flag to 1 differ depending on the operating mode of the RSPI. For details, refer to section 40.2.4, RSPI Status Register (SPSR).

40.3.9.1 In Master Mode

In master mode, the conditions for setting the SPCF flag to 1 are the same in any combination of SPI operation or clock synchronous operation and full-duplex or transmit-only simplex communications.
When the SPSSR.SPCP[2:0] bits becomes 000b and there is no next transmit data, the SPSR.IDLNF flag becomes 0 and the SPCF flag becomes 1.
The SPCF flag is cleared when 0 is written to the SPCF flag after confirming the flag is 1 or when the next data is written to the transmit buffer.

40.3.9.2 Full-duplex or transmit-only simplex communications in slave mode under SPI operation

When the SSLx0 signal is negated while the transmit buffer and the transmit shift register are empty, the SPSR.SPCF flag becomes 1.
The SPCF flag is cleared when 0 is written to the SPCF flag after confirming the flag is 1 or when the next data is written

to the transmit buffer.

40.3.9.3 Receive-only simplex communications in slave mode under SPI operation

When the SSLx0 signal is negated after the number of frames set in the SPDCR.SPFC[1:0] bits have been received, the SPSR.SPCF flag becomes 1.
The SPCF flag is cleared when 0 is written to the SPCF flag after confirming the flag is 1 or when the SSLx0 signal is asserted when the next communication is started.

40.3.9.4 Full-duplex or transmit-only simplex communications in slave mode under clock synchronous operation

When the transmit buffer and the transmit shift register are empty, the SPSR.SPCF flag becomes 1 at the sampling timing of the last bit (the last even edge of the RSPCK).
The SPCF flag is cleared when 0 is written to the SPCF flag after confirming the flag is 1 or when the next data is written to the transmit buffer.

40.3.9.5 Receive-only simplex communications in slave mode under clock synchronous operation

When the number of frames set in the SPDCR.SPFC[1:0] bits have been received (the last even edge of the RSPCK), the SPSR.SPCF flag becomes 1.
The SPCF flag is cleared when 0 is written to the SPCF flag after confirming the flag is 1 or when the RSPCK signal changes when next communication is started.

40.3.10 Error Detection

In the normal RSPI serial transfer, the data written to the transmit buffer of the SPDR register is transmitted, and the received data can be read from the receive buffer of the SPDR register. If access is made to the SPDR register, depending on the status of the transmit/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 underrun error, overrun error, parity error, or mode fault error. Table 40.7 lists the relationship between non-normal transfer operations and the RSPI's error detection function.

Table 40.7 Relationship between Non-Normal Transfer Operations and RSPI Error Detection Function

	Occurrence Condition	RSPI Operation	Error Detection
1	The SPDR register is written when the transmit buffer is full.	• The contents of the transmit buffer are kept. • Missing write data.	None
2	The SPDR register is read when the receive buffer is empty.	If reception has completed, then the received data is output to the bus. Otherwise, data received previously is output instead.	None
3	Serial transfer is started when transmit data is still not loaded on the shift register while the RSPI is set to perform full-duplex or transmit-only simplex communications in slave mode.	• Serial transfer is suspended • Transmit/receive data is missing • The MISO signal output is disabled • RSPI function is disabled	Underrun error
4	Serial transfer terminates when the receive buffer is full.	• The contents of the receive buffer are kept. • Missing receive data.	Overrun error
5	An incorrect parity bit is received when performing full-duplex or receive-only simplex communications with the parity function enabled.	The parity error flag is set.	Parity error
6	The SSLx0 input signal is asserted when the serial transfer is idle in multi-master mode.	• Driving of the RSPCKx, MOSIx, SSLx1 to SSLx3 output signals is stopped. • RSPI function is disabled.	Mode fault error
7	The SSLx0 input signal is asserted during serial transfer in multi-master mode.	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the RSPCKx, MOSIx, SSLx1 to SSLx3 output signals is stopped. • RSPI function is disabled.	Mode fault error
8	The SSLx0 input signal is negated during serial transfer in slave mode.	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the MISOx output signal is stopped. • RSPI function is disabled.	Mode fault error

On operation 1 described in Table 40.7, the RSPI does not detect an error. To prevent data omission during the writing to the SPDR register, the SPDR register should be written when a transmit buffer empty interrupt request occurs or while the SPSR.SPTEF flag is 1.

Likewise, the RSPI does not detect an error on operation 2. To prevent extraneous data from being read, the SPDR register should be read when a receive buffer full interrupt request occurs or while the SPSR.SPRF flag is 1.

An underrun error shown in 3 is described in section 40.3.10.4, Underrun Error. An overrun error shown in 4 is described in section 40.3.10.1, Overrun Error. A parity error shown in 5 is described in section 40.3.10.2, Parity Error. A mode fault error shown in 6 to 8 is described in section 40.3.10.3, Mode Fault Error.

For the transmit and receive interrupts, refer to section 40.3.7, Transmit Buffer Empty/Receive Buffer Full Interrupts.

40.3.10.1 Overrun Error

If a serial transfer ends when the receive buffer of the SPDR register is full, the RSPI detects an overrun error, and sets the SPSR.OVRF flag to 1. When the OVRF flag is 1, the RSPI does not copy data from the shift register to the receive buffer so that the data prior to the occurrence of the error is retained in the receive buffer. To set the OVRF flag to 0, write 0 to the OVRF flag after the CPU has read the SPSR register with the OVRF flag set to 1.
Figure 40.30 shows an example of operations of the SPRF and OVRF flags. 'SPSR access' and 'SPDR access' shown in Figure 40.30 indicate the condition of accesses to the SPSR and SPDR registers, respectively, where 'W' denotes a write cycle, and 'R' a read cycle. In the example in Figure 40.30, 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 RSPCKx waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

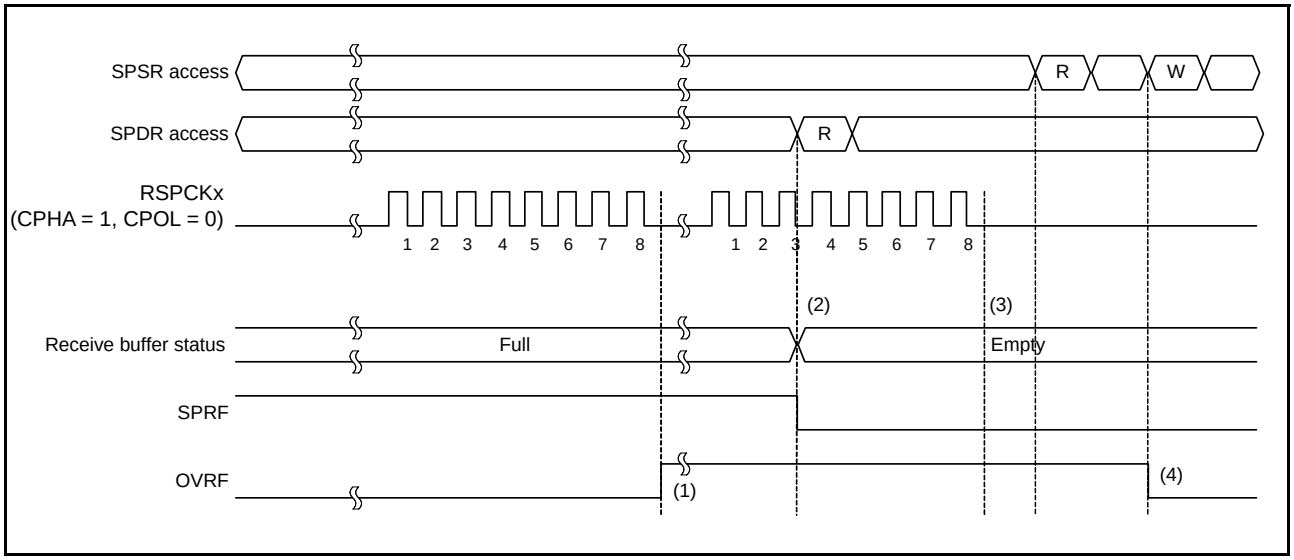


Figure 40.30 Operation Example of the SPRF and OVRF Flags

The operation of the flags at the timing shown in steps (1) to (4) in the figure is described below.

(1) If a serial transfer terminates with the receive buffer full (the SPRF flag is 1), the RSPI detects an overrun error, and sets the OVRF flag to 1. The RSPI does not copy the data in the shift register to the receive buffer. Even if the SPPE bit is 1, parity errors are not detected. In master mode, the RSPI copies the pointer value to the SPCMDm register to the SPSSR.SPECM[2:0] bits.

(2) When the SPDR register is read, the RSPI outputs the data in the receive buffer. At this time the SPRF flag becomes 0. Even if the receive buffer becomes empty, the OVRF flag does not become 0.

(3) If the serial transfer ends with the OVRF flag being 1 (an overrun error occurs), the RSPI does not copy the data in the shift register to the receive buffer (the SPRF flag remains 0). A receive buffer full interrupt is not generated. Even if the SPPE bit is 1, parity errors are not detected. When in master mode, the RSPI does not update the SPSSR.SPECM[2:0] bits. When in an overrun error state and the RSPI does not copy the received data from the shift register to the receive buffer, upon termination of the serial transfer, the RSPI determines that the shift register is empty; in this manner, data transfer from the transmit buffer to the shift register is enabled.

(4) If 0 is written to the OVRF flag after the SPSR register is read when the OVRF flag is 1, the OVRF flag is set to 0.

The occurrence of an overrun can be checked either by reading the SPSR register or by using an error interrupt and reading the SPSR register. When executing a serial transfer, measures should be taken to ensure the early detection of overrun errors, such as reading the SPSR register immediately after the SPDR register is read. When the RSPI is used in master mode, the pointer value to the SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.
If an overrun error occurs and the OVRF flag is set to 1, normal reception operations cannot be performed until the OVRF flag is set to 0.

When the RSPCK auto-stop function is enabled in master mode, an overrun error does not occur. Figure 40.31 and Figure 40.32 show the clock stop waveform when a serial transfer continues while the receive buffer is full in master mode.

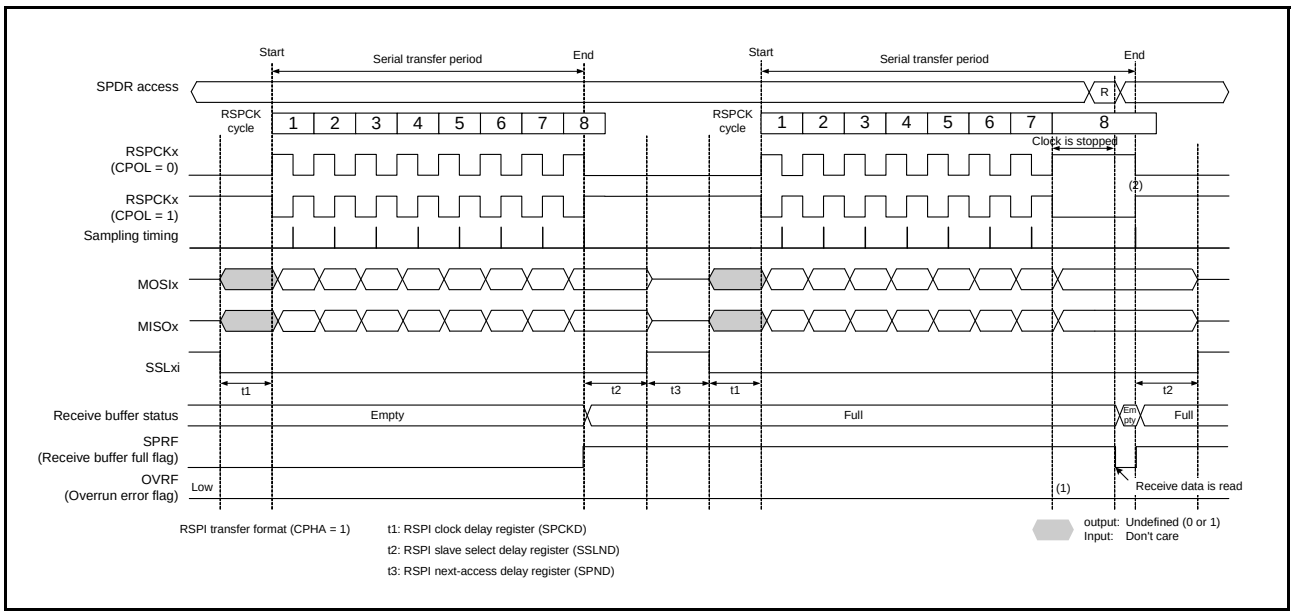


Figure 40.31 Clock Stop Waveform When a Serial Transfer Continues While the Receive Buffer is Full in Master Mode (CPHA = 1)

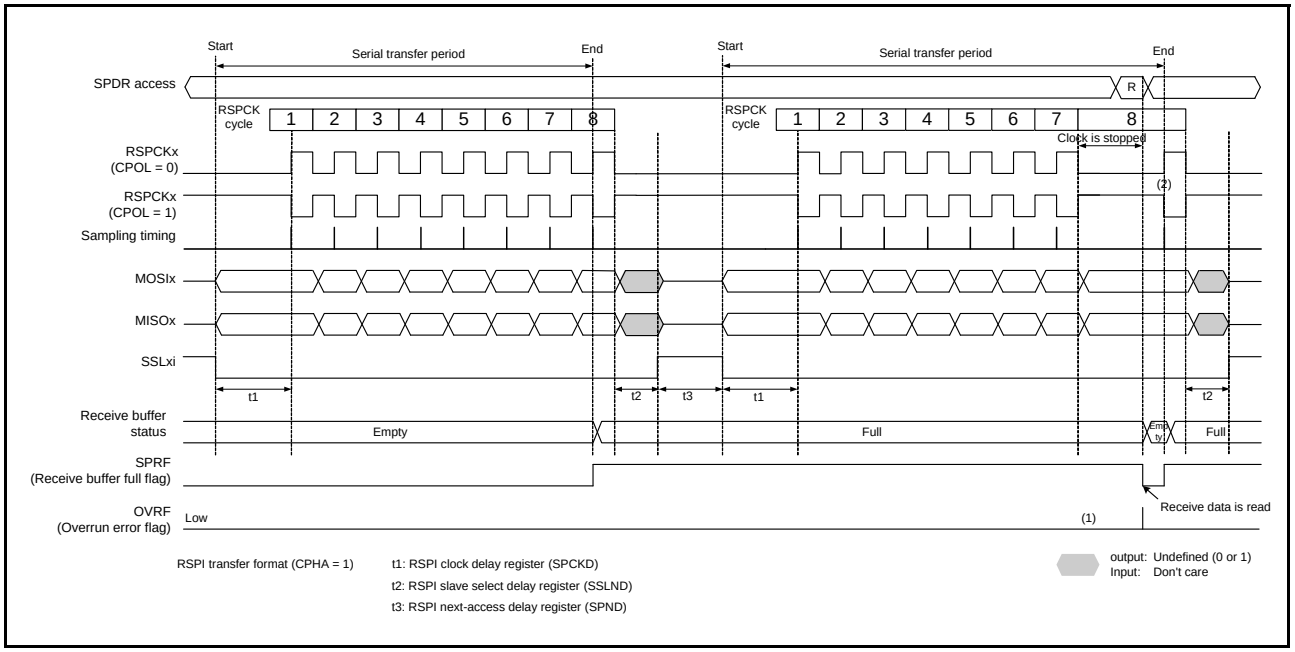


Figure 40.32 Clock Stop Waveform When a Serial Transfer Continues While the Receive Buffer is Full in Master Mode (CPHA = 0)

The operation of the flags at the timings shown in steps (1) and (2) in the figure is described below.

(1) When the receive buffer is full, an overrun error does not occur because the RSPCK clock is stopped.

(2) If the SPDR register is read while the clock is stopped, data in the receive buffer can be read. The RSPCK clock restarts after reading the receive buffer (after the SPRF flag becomes 0).

40.3.10.2 Parity Error

If full-duplex communication or receive-only simplex communication is performed with the SPCR.TXMD bit set to 0 and the SPCR2.SPPE bit set to 1, when serial transfer ends, the RSPI checks whether there are parity errors. Upon detecting a parity error in the received data, the RSPI sets the SPSR.PERF flag to 1. Since the RSPI does not copy the data in the shift register to the receive buffer when the SPSR.OVRF flag is set to 1, parity error detection is not performed for the received data. To set the PERF flag to 0, write 0 to the PERF flag after the SPSR register is read with the PERF flag set to 1.

Figure 40.33 shows an example of operation of the OVRF and PERF flags. 'SPSR access' shown in Figure 40.33 indicates the condition of access to the SPSR register, where 'W' denotes a write cycle, and 'R' a read cycle. In the example of Figure 40.33, full-duplex communication or receive-only simplex communication 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 RSPCKx waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

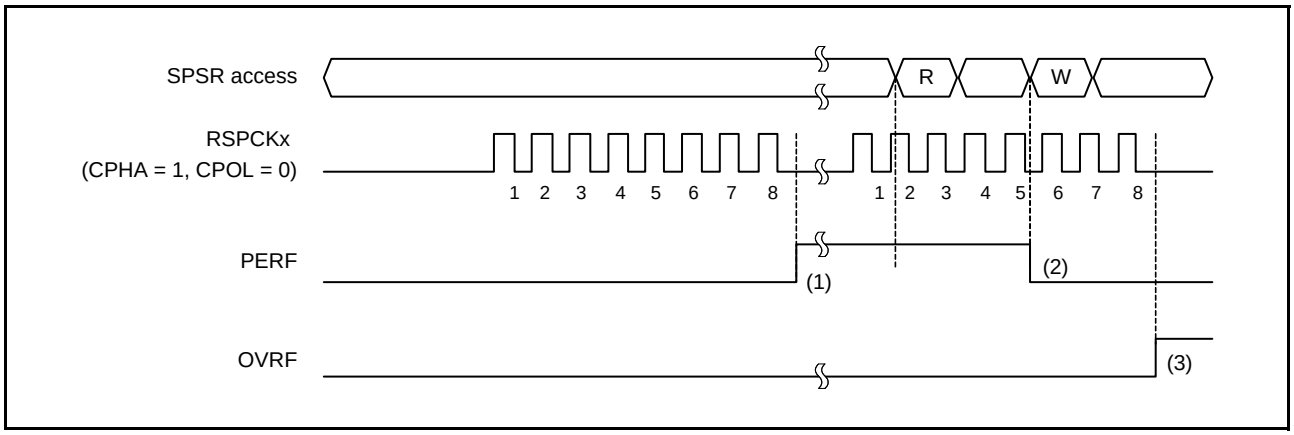


Figure 40.33 Operation Example of PERF Flag

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 the SPCMDm register to the SPSSR.SPECM[2:0] bits.

(2) If 0 is written to the PERF flag after the SPSR register is read when the PERF flag is 1, the PERF flag is set to 0.

(3) When the RSPI detects an overrun error and serial transfer is terminated, the data in the shift register is not copied to the receive buffer. The RSPI does not perform parity error detection at this timing.

The occurrence of a parity error can be checked either by reading the SPSR register or by using an error interrupt and reading the SPSR register. When executing a serial transfer, measures should be taken to ensure the early detection of parity errors, such as reading the SPSR register. When the RSPI is used in master mode, the pointer value to the SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

40.3.10.3 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 SSLx0 input signal of the RSPI in multi-master mode, the RSPI detects a mode fault error irrespective of the status of the serial transfer, and sets the SPSR.MODF flag to 1. Upon detecting the mode fault error, the RSPI copies the value of the pointer to the SPCMDm register to the SPSSR.SPECM[2:0] bits. The active level of the SSLx0 signal is determined by the SSLP.SSL0P bit.

When the MSTR bit is 0, the RSPI operates in slave mode. The RSPI detects a mode fault error if the MODFEN bit of the RSPI in slave mode is 1, and the SPMS bit is 0, and if the SSLx0 input signal is negated during the serial transfer period (from the time the driving of valid data is started to the time the final valid data is fetched).

Upon detecting a mode fault error, the RSPI stops driving of the output signals and clears the SPCR.SPE bit to 0 (refer to section 40.3.11, 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 the SPSR register or by using an error interrupt and reading the SPSR register. Detecting mode fault errors without utilizing the error interrupt requires polling of the SPSR register. When using the RSPI in master mode, the pointer value to the SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

When the MODF flag is 1, writing of the value 1 to the SPE bit is ignored by the RSPI. To enable the RSPI function after the detection of a mode fault error, set the MODF flag to 0. When setting the MODF flag to 0, the SPE bit becomes 1.

40.3.10.4 Underrun Error

If a serial transfer is started when the SPCR.SPE bit is 1 (RSPI function is enabled) and transmit data is still not loaded on the shift register while RSPI operates in slave mode (the SPCR.MSTR bit is 0) and transmitter is enabled (the SPCR3.RXMD bit is 0), the RSPI detects an underrun error, and sets the SPSR.MODF and SPSR.UDRF flags to 1. Upon detecting an underrun error, the RSPI stops driving of the output signals and clears the SPCR.SPE bit to 0. Clearing of the SPE bit disables the RSPI function. (Refer to section 40.3.11, Initializing RSPI). The occurrence of an underrun error can be checked either by reading the SPSR register or by using an error interrupt and reading the SPSR register. Detecting underrun errors without utilizing the error interrupt requires polling of the SPSR register. When the MODF flag is 1, writing of the value 1 to the SPE bit is ignored by the RSPI. To enable the RSPI function after the detection of an underrun error, set the MODF flag to 0. When setting the MODF flag to 0, the SPE bit becomes 1.

40.3.11 Initializing RSPI

If 0 is written to the SPCR.SPE bit or the RSPI sets the SPE bit to 0 because of the detection of a mode fault error or an underrun error, the RSPI disables the RSPI function, and initializes some of the module functions. When a system reset is generated, the RSPI initializes all of the module functions. The following describes initialization by the clearing of the SPCR.SPE bit and initialization by a system reset.

40.3.11.1 Initialization by Clearing the SPE Bit

When the SPCR.SPE bit is set to 0, the RSPI performs the following initialization:

- Aborting the transmission and reception that is being executed
- Stopping the driving of output signals (Hi-Z) in slave mode
- Initializing the internal state of the RSPI
- Initializing the transmit buffer of the RSPI (Set the SPTEF flag to 1)

Initialization by the clearing of the SPE bit does not initialize the control bits of the RSPI. For this reason, the RSPI can be started in the same transfer mode as prior to the initialization if the SPE bit is set to 1 again.
The SPSR.SPRF, SPCF, UDRF, PERF, MODF, and OVRF flags are not initialized, nor is the value of the RSPI sequence status register (SPSSR) initialized. For this reason, even after the RSPI is initialized, data from the receive buffer can be read and the communication status that was being executed before the initialization and the status of error occurrence during the RSPI transfer can be checked.
The transmit buffer is initialized to an empty state (the SPTEF flag is 1). Therefore, if the SPCR.SPTIE bit is set to 1 after RSPI initialization, a transmit buffer empty interrupt is generated. When the RSPI is initialized, in order to disable any transmit buffer empty interrupt, 0 should be written to the SPTIE bit simultaneously with the writing of 0 to the SPE bit.

40.3.11.2 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 40.3.11.1, Initialization by Clearing the SPE Bit.

40.3.12 SPI Operation

40.3.12.1 Master Mode Operation

The only difference between single-master mode operation and multi-master mode operation lies in mode fault error detection (refer to section 40.3.10, Error Detection). When operating in single-master mode, the RSPI does not detect mode fault errors whereas the RSPI running in multi-master mode does detect mode fault errors. This section explains operations that are common to single-master mode and multi-master mode.

(1) Starting a Serial Transfer

The RSPI updates the data in the transmit buffer (SPTX) when data is written to the RSPI data register (SPDR) with the RSPI transmit buffer being empty (the SPTEF flag is 1 and data for the next transfer is not set). When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits are written to the SPDR register, the RSPI copies data from the transmit buffer to the shift register and starts serial transfer. Upon copying transmit data to the shift register, the RSPI changes the status of the shift register to “full”, and upon termination of serial transfer, it changes the status of the shift register to “empty”. The status of the shift register cannot be referenced.
For details on the RSPI transfer format, refer to section 40.3.5, Transfer Format. The polarity of the SSLxi output pins depends on the SSLP register settings.

(2) Terminating a Serial Transfer

Irrespective of the SPCMDm.CPHA bit, the RSPI terminates the serial transfer after transmitting an RSPCKx edge corresponding to the final sampling timing. If free space is available in the receive buffer (SPRX) (the SPRF flag is 0), upon termination of serial transfer, the RSPI copies data from the shift register to the receive buffer of the SPDR register. It should be noted that the final sampling timing varies depending on the bit length of transfer data. In master mode, the RSPI data length depends on the SPCMDm.SPB[3:0] bit setting. The polarity of the SSLxi output pin depends on the SSLP register settings.
For details on the RSPI transfer format, refer to section 40.3.5, Transfer Format.

(3) Sequence Control

The transfer format that is employed in master mode is determined by the SPSCR, SPCMDm, SPBR, SPCKD, SSLND, and SPND registers.

The SPSCR register 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 the SPCMDm register: SSLxi pin output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCK polarity/phase, whether the SPCKD register is to be referenced, whether the SSLND register is to be referenced, and whether the SPND register is to be referenced. The SPBR register holds some of the bit rate settings; the SPCKD register, an RSPI clock delay value; the SSLND register, an SSL negation delay; and the SPND register, a next-access delay value.

According to the sequence length that is assigned to the SPSCR register, the RSPI makes up a sequence comprised of a part or all of the 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 the SPCMD0 register, 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 the SPCMD0 register, and in this manner the sequence is executed repeatedly.

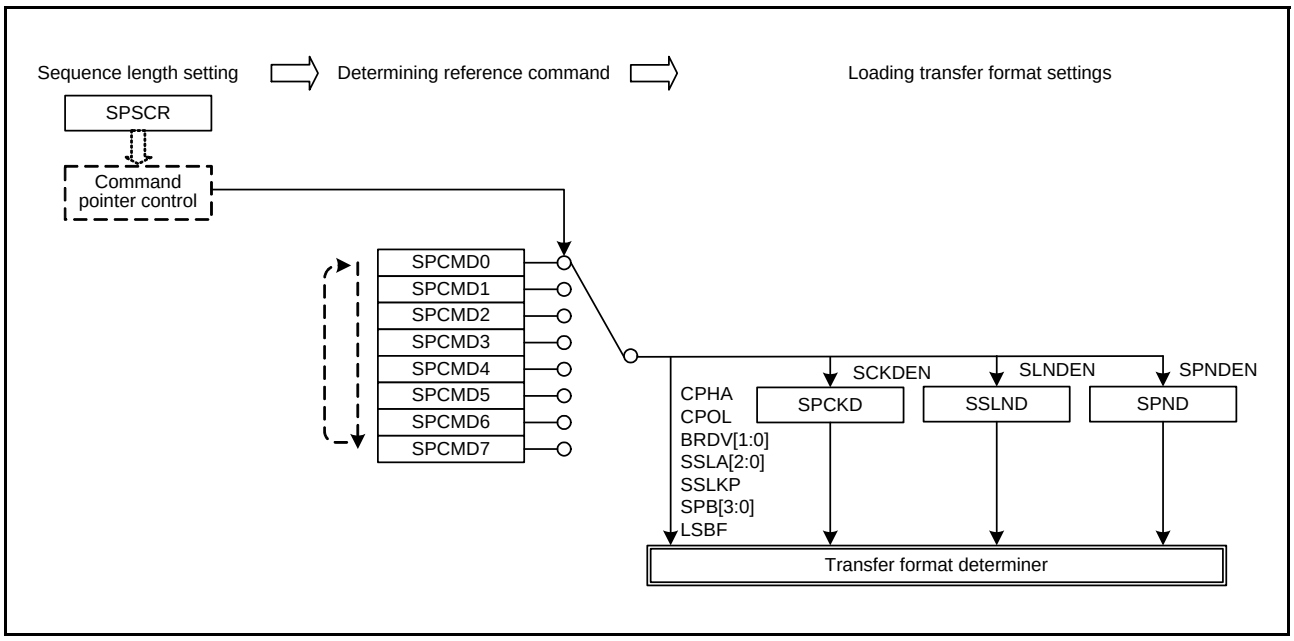


Figure 40.34 Procedure for Determining the Form of Serial Transfer in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

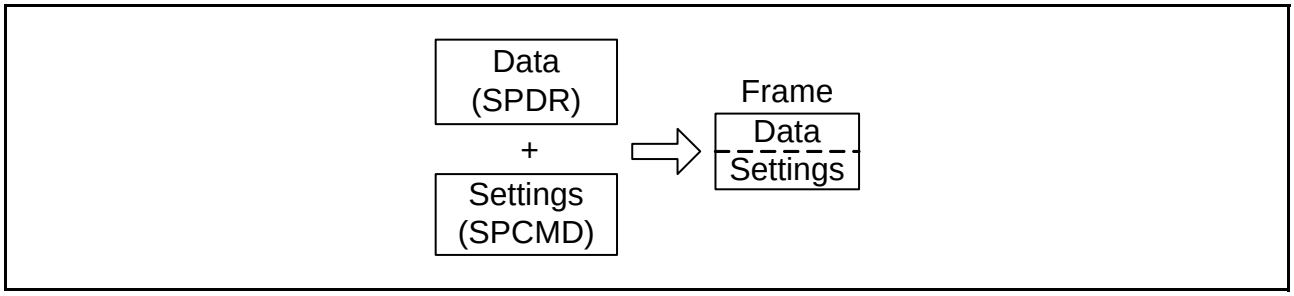


Figure 40.35 Concept of a Frame

Figure 40.36 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 40.4.

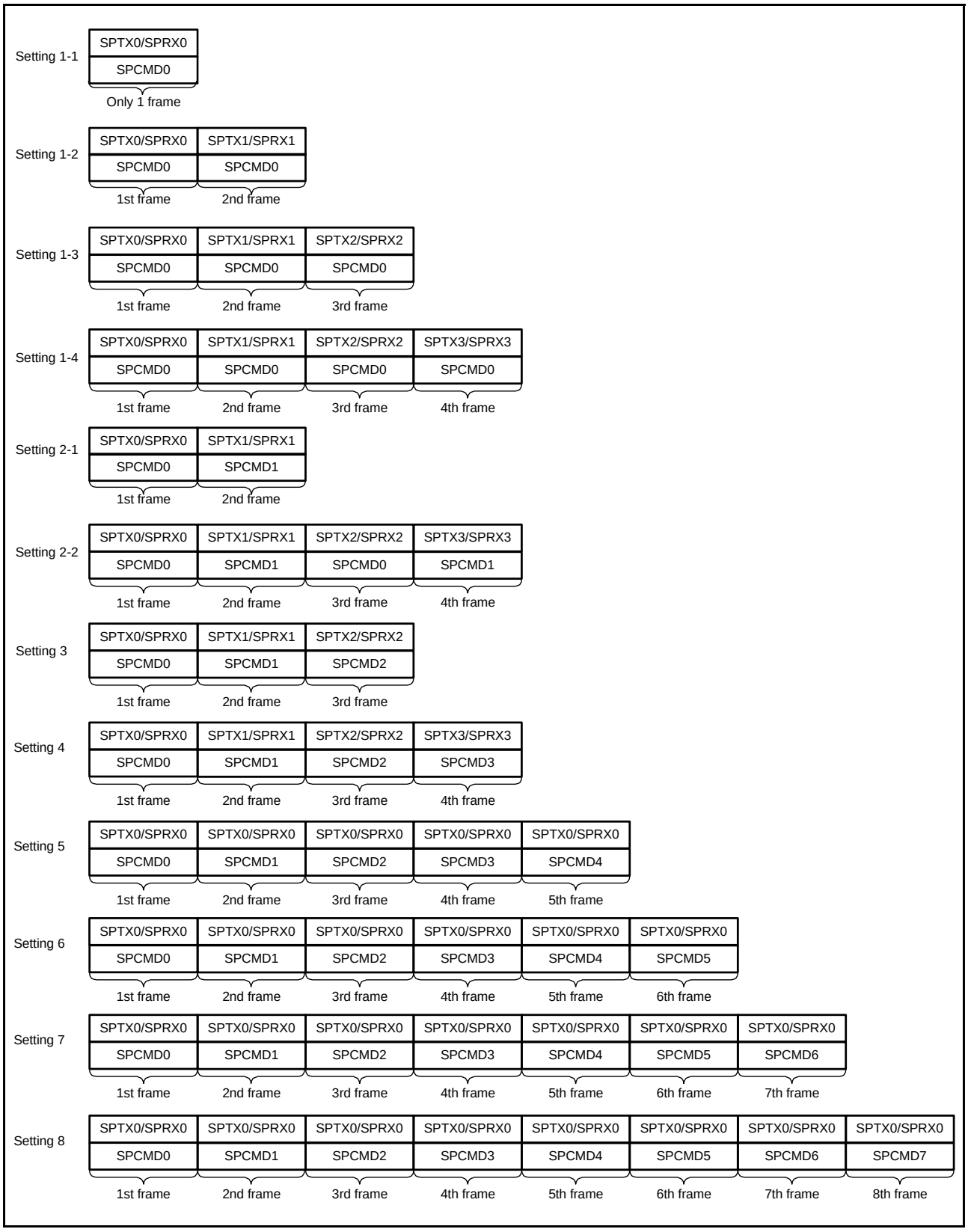


Figure 40.36 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations

(4) Burst Transfer

If the SPCMDm.SSLKP bit that the RSPI references during the current serial transfer is 1, the RSPI keeps the SSLxi signal level during the serial transfer until the beginning of the SSLxi signal assertion for the next serial transfer. If the SSLxi signal level for the next serial transfer is the same as the SSLxi signal level for the current serial transfer, the RSPI can execute continuous serial transfers while keeping the SSLxi signal assertion status (burst transfer).
Figure 40.37 shows an example of an SSLxi 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 40.37. It should be noted that the polarity of the SSLxi output signal depends on the SSLP register settings.

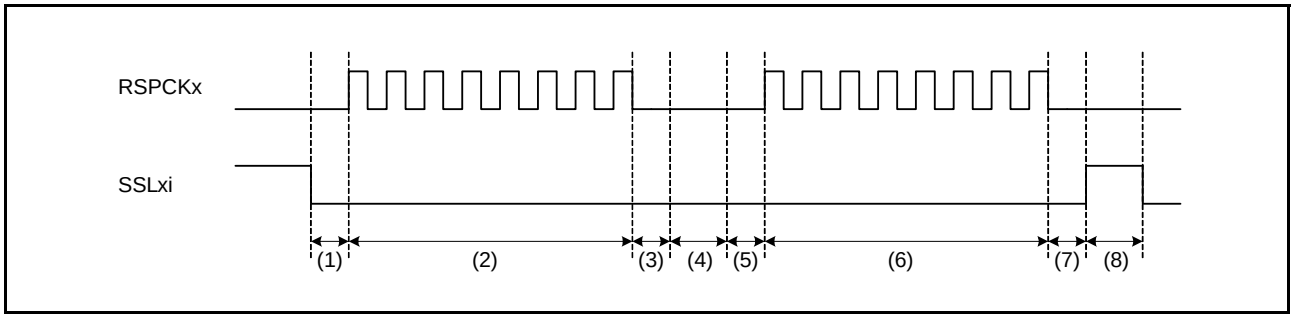


Figure 40.37 Example of Burst Transfer Operation Using SSLKP Bit (CPHA = 1, CPOL = 0)

(1) Based on the SPCMD0 register, the RSPI asserts the SSLxi signal and inserts RSPCK delays.
(2) The RSPI executes serial transfers according to the SPCMD0 register.
(3) The RSPI inserts SSL negation delays.
(4) Since the SPCMD0.SSLKP bit is 1, the RSPI keeps the SSLxi signal value on the SPCMD0 register. This period is sustained, at the shortest, for a period equal to the next-access delay of the SPCMD0 register. 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 the SPCMD1 register, the RSPI inserts RSPCK delays.
(6) The RSPI executes serial transfers according to the SPCMD1 register.
(7) The RSPI inserts SSL negation delays.
(8) Because the SPCMD1.SSLKP bit is 0, the RSPI negates the SSLxi signal. In addition, a next-access delay is inserted according to the SPCMD1 register.

If the SSLxi signal output settings in the SPCMDm register in which 1 is assigned to the SSLKP bit are different from the SSLxi signal output settings in the SPCMDm register to be used in the next transfer, the RSPI switches the SSLxi signal status to SSLxi signal assertion ((5) in Figure 40.37) corresponding to the command for the next transfer. Note that if such an SSLxi signal switching occurs, the slaves that drive the MISOx signal compete, and collision of signal levels may occur.
The RSPI in master mode references the SSLxi 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 SSLxi signal assertion for the next transfer that is detected internally.
When the SPCR3.SCKDDIS bit is set to 1 (Does not insert delays between data bytes during burst transfer), the delays described in (3) to (5) above are not inserted and only delay of 0.5 cycles of RSPCK is inserted.

(5) RSPCK Delay (t1)

The RSPCK delay value 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 the SPCKD register, as listed in Table 40.8. For a definition of RSPCK delay, refer to section 40.3.5, Transfer Format.
When the SPCMDm.SSLKP bit is set to 1 (burst transfer) and the SPCR3.SCKDDIS bit is set to 1 (Does not insert delays between data bytes during burst transfer), the RSPCK delay is inserted only in the first frame.

Table 40.8 Relationship among SCKDEN Bit, SPCKD, and RSPCK Delay Value

SPCMDm.SCKDEN Bit	SPCKD.SCKDL[2:0] Bits	RSPCK Delay Value
0	000b to 111b	1 RSPCK
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

(6) SSL Negation Delay (t2)

The SSL negation delay value 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 the SSLND register, as listed in Table 40.9. For a definition of SSL negation delay, refer to section 40.3.5, Transfer Format.
When the SPCMDm.SSLKP bit is set to 1 (burst transfer) and the SPCR3.SCKDDIS bit is set to 1 (Does not insert delays between data bytes during burst transfer), the SSL negation delay is inserted only in the last frame.

Table 40.9 Relationship among SLNDEN Bit, SSLND, and SSL Negation Delay Value

SPCMDm.SLNDEN Bit	SSLND.SLNDL[2:0] Bits	SSL Negation Delay Value
0	000b to 111b	1 RSPCK
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

(7) Next-Access Delay (t3)

The next-access delay value 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 40.10. For a definition of next-access delay, refer to section 40.3.5, Transfer Format.

When the SPCMDm.SSLKP bit is set to 1 (burst transfer) and the SPCR3.SCKDDIS bit is set to 1 (Does not insert delays between data bytes during burst transfer), the next-access delay is inserted only in the last frame.

Table 40.10 Relationship among SPNDEN Bit, SPND, and Next-Access Delay Value

SPCMDm.SPNDEN Bit	SPND.SPNDL[2:0] Bits	Next-Access Delay Value
0	000b to 111b	1 RSPCK + 2 PCLK
1	000b	1 RSPCK + 2 PCLK
	001b	2 RSPCK + 2 PCLK
	010b	3 RSPCK + 2 PCLK
	011b	4 RSPCK + 2 PCLK
	100b	5 RSPCK + 2 PCLK
	101b	6 RSPCK + 2 PCLK
	110b	7 RSPCK + 2 PCLK
	111b	8 RSPCK + 2 PCLK

(8) Initialization Flowchart

Figure 40.38 is a flowchart illustrating an example of initialization in SPI operation when the RSPI is used in master mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

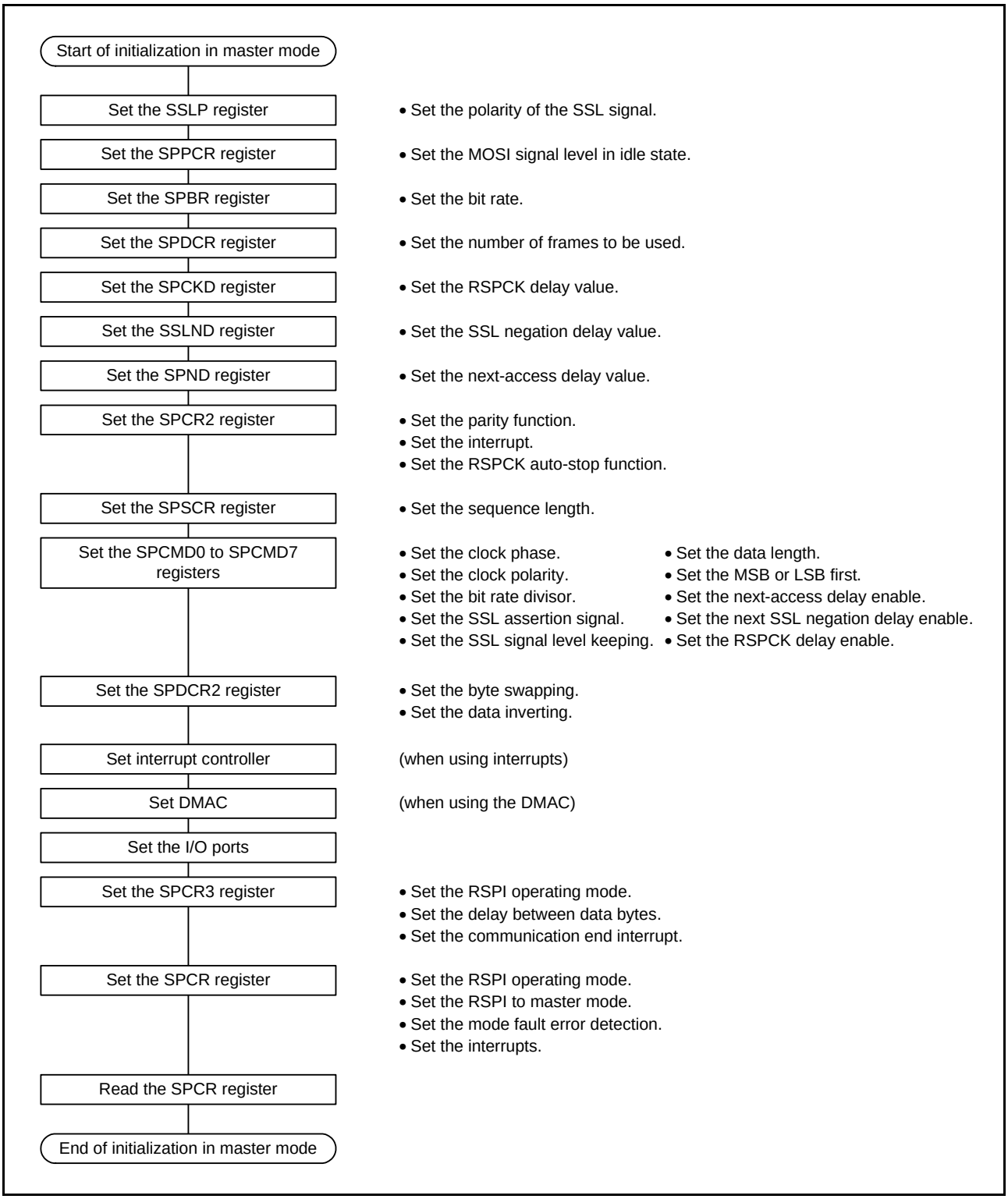


Figure 40.38 Example of Initialization Flowchart in Master Mode (SPI Operation)

(9) Software Processing Flow

Figure 40.40 to Figure 40.42 show examples of the flow of software processing.

(a) Communication Preprocessing Flow

Before starting communications, clear the error flags and disable the idle interrupt and communication end interrupt. Then enable the RSPI function and the required interrupts at the same time.

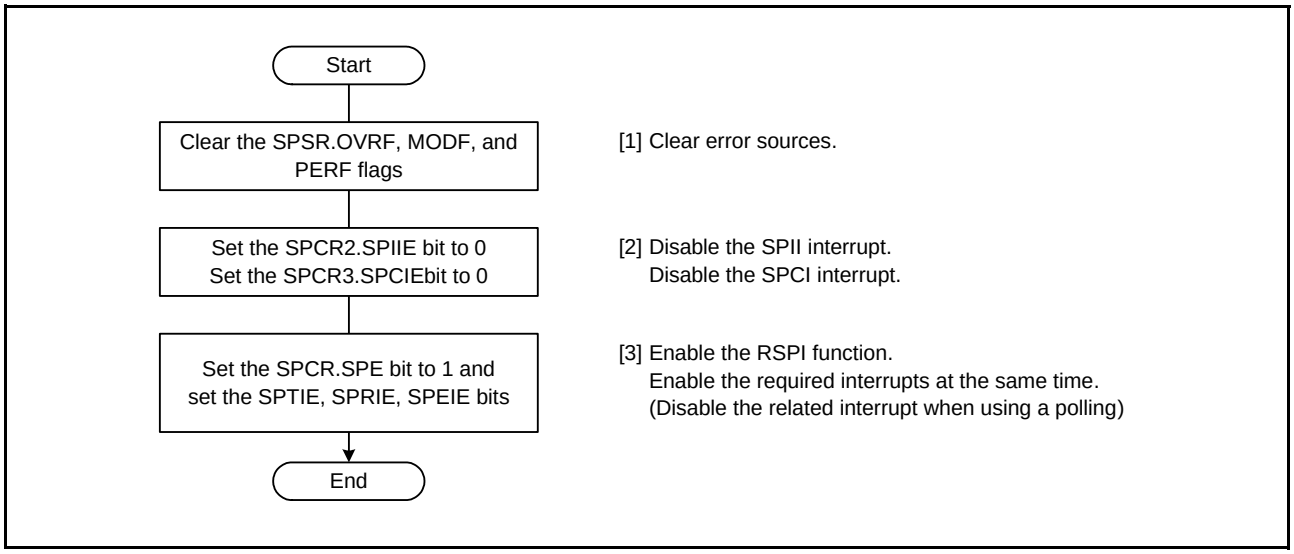


Figure 40.39 Flowchart in Master Mode (Communication Preprocess)

(b) Transmit Processing Flow

When transmitting data, the CPU will be notified of the completion of data transmission by enabling the SPII or SPCI interrupt after the last writing of data for transmission.

The completion of data transmission can also be checked by polling to see if the SPSR.IDLNF flag has become 0 or the SPCF flag has become 1, instead of using the SPII or SPCI interrupt. However, one cycle of PCLK is required for the time from when data for transmission is written in the SPDR register to when the IDLNF flag becomes 1 and the SPCF flag becomes 0. After the last data is written in the SPDR register, discard the value of the SPSR register once not to judge the condition with the IDLNF flag which has not yet become 1 or the SPCF flag which has not yet become 0, and read and use the value of the IDLNF or SPCF flag to confirm the completion of data transmission.

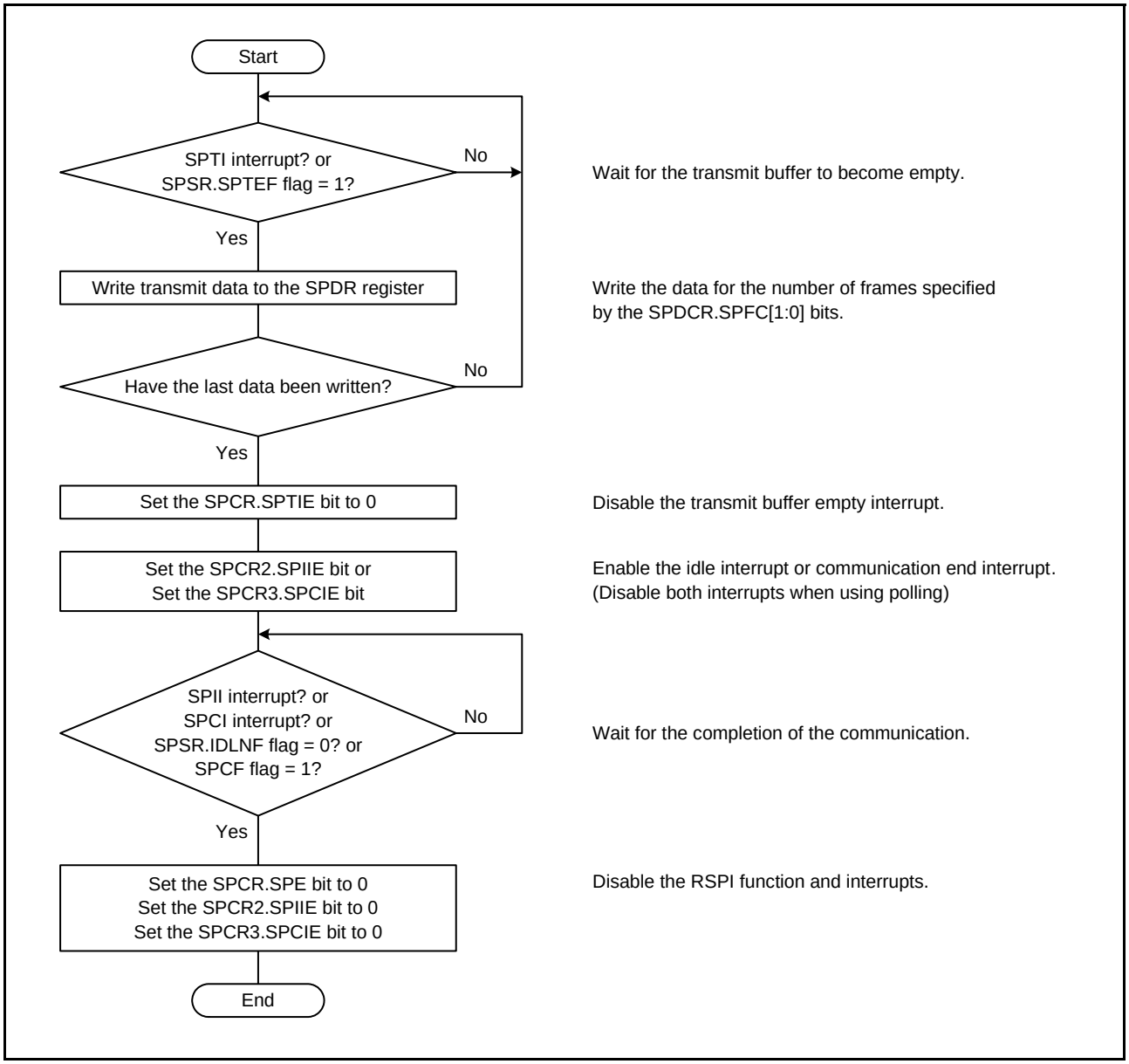


Figure 40.40 Flowchart in Master Mode (Transmission)

(c) Receive Processing Flow

The RSPI does not support receive-only simplex communications in master mode, so processing for transmission is required.

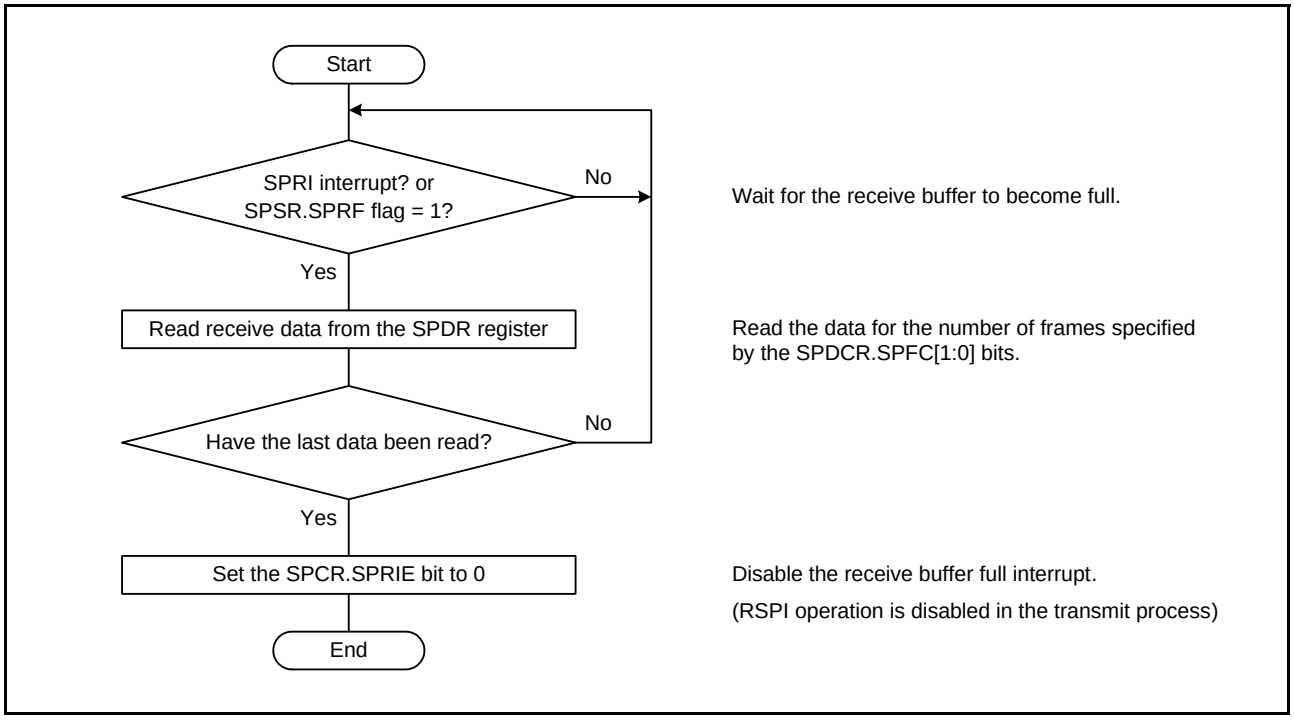


Figure 40.41 Flowchart in Master Mode (Reception)

(d) Flow of Error Processing

When a mode fault error is generated, the SPCR.SPE bit is automatically cleared, stopping operations for transmission and reception. For errors from other sources, however, the SPCR.SPE bit is not cleared and operations for transmission and reception continue; accordingly, we recommend clearing of the SPCR.SPE bit to stop operations in the case of errors other than mode fault errors. Not doing so will lead to updating of the SPSSR.SPECM[2:0] bits.
When interrupts are used and an error occurs, if the ICU.IRn.IR flag for the SPTI or SPRI interrupt request is set to 1, clear the ICU.IRn.IR flag in the error processing routine. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPI.

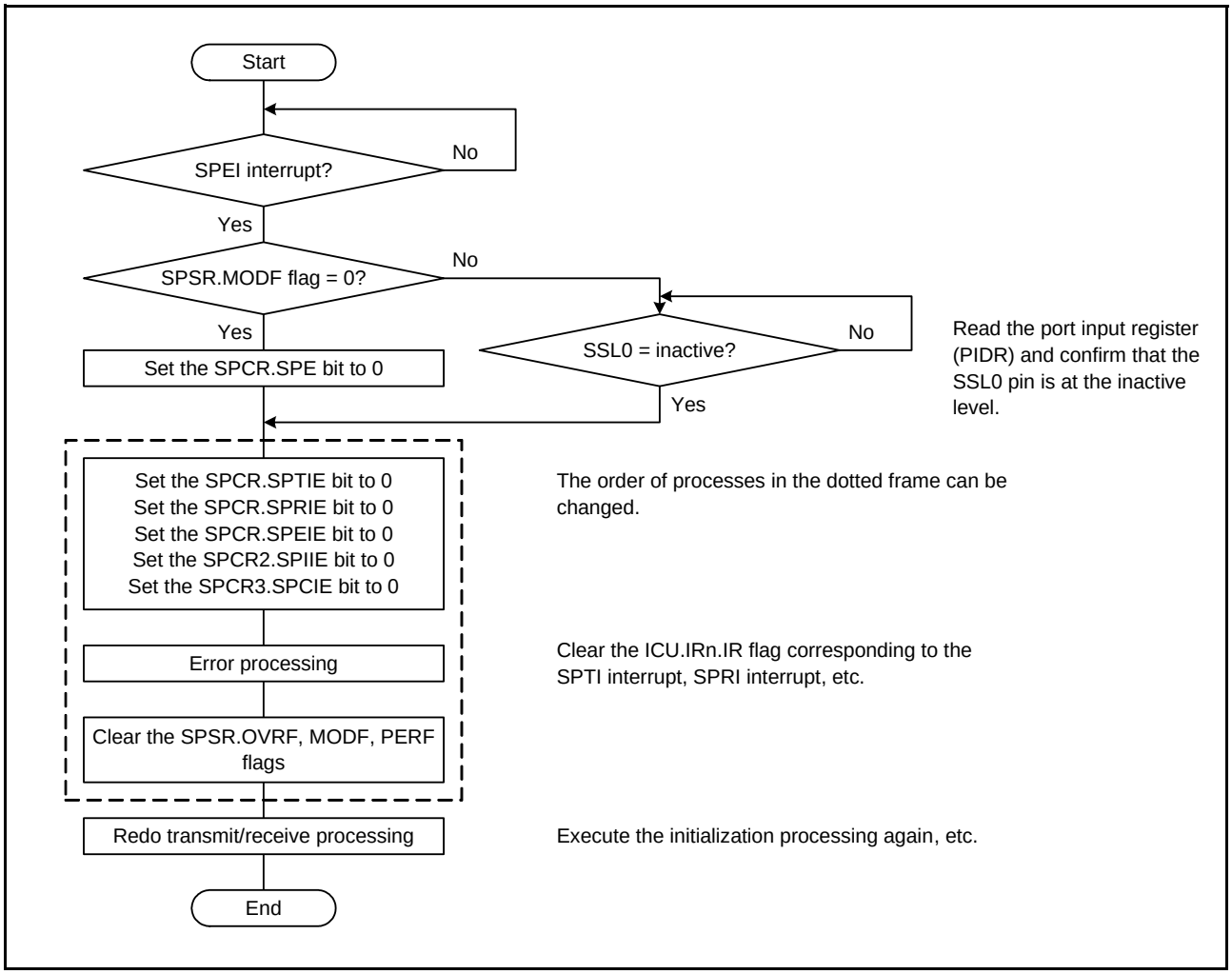


Figure 40.42 Flowchart for Master Mode (Error Processing)

40.3.12.2 Slave Mode Operation

(1) Starting a Serial Transfer

If the SPCMD0.CPHA bit is 0, when detecting an SSLx0 input signal assertion, the RSPI needs to start driving valid data to the MISOx output signal. For this reason, when the CPHA bit is 0, the assertion of the SSLx0 input signal triggers the start of a serial transfer.

If the CPHA bit is 1, when detecting the first RSPCKx edge in an SSLx0 signal asserted condition, the RSPI needs to start driving valid data to the MISOx output signal. For this reason, when the CPHA bit is 1, the first RSPCKx edge in an SSLx0 signal asserted condition triggers the start of a serial transfer.

Irrespective of the CPHA bit setting, the timing at which the RSPI starts driving of the MISOx output signal is the SSLx0 signal assertion timing. The data which is output by the RSPI is either valid or invalid, depending on the CPHA bit setting.

For details on the RSPI transfer format, refer to section 40.3.5, Transfer Format. The polarity of the SSLx0 input signal depends on the setting of the SSLP.SSL0P bit.

(2) Terminating a Serial Transfer

Irrespective of the SPCMD0.CPHA bit, the RSPI terminates the serial transfer after detecting an RSPCKx edge corresponding to the final sampling timing. When free space is available in the receive buffer (the SPRF flag is 0), upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the SPDR register. Upon termination of a serial transfer the RSPI changes the status of the shift register to “empty”, regardless of the receive buffer state. A mode fault error occurs if the RSPI detects an SSLx0 input signal negation from the beginning of serial transfer to the end of serial transfer (refer to section 40.3.10, 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 SSLx0 input signal depends on the SSLP.SSL0P bit setting.

For details on the RSPI transfer format, refer to section 40.3.5, 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 SSLx0 input signal. In the type of configuration shown in Figure 40.7 as an example, if the RSPI is used in single-slave mode, the SSLx0 signal is fixed at the active state. Therefore, when the CPHA bit is set to 0, the RSPI cannot correctly start a serial transfer. To correctly execute transmit/receive operations by the RSPI in slave mode in a configuration in which the SSLx0 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 SSLx0 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 SSLx0 input signal. If the CPHA bit is 1, the period from the first RSPCKx edge to the sampling timing for the reception of the final bit in an SSLx0 signal active state corresponds to a serial transfer period. Even when the SSLx0 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 40.43 is a flowchart illustrating an example of initialization in SPI operation when the RSPI is used in slave mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

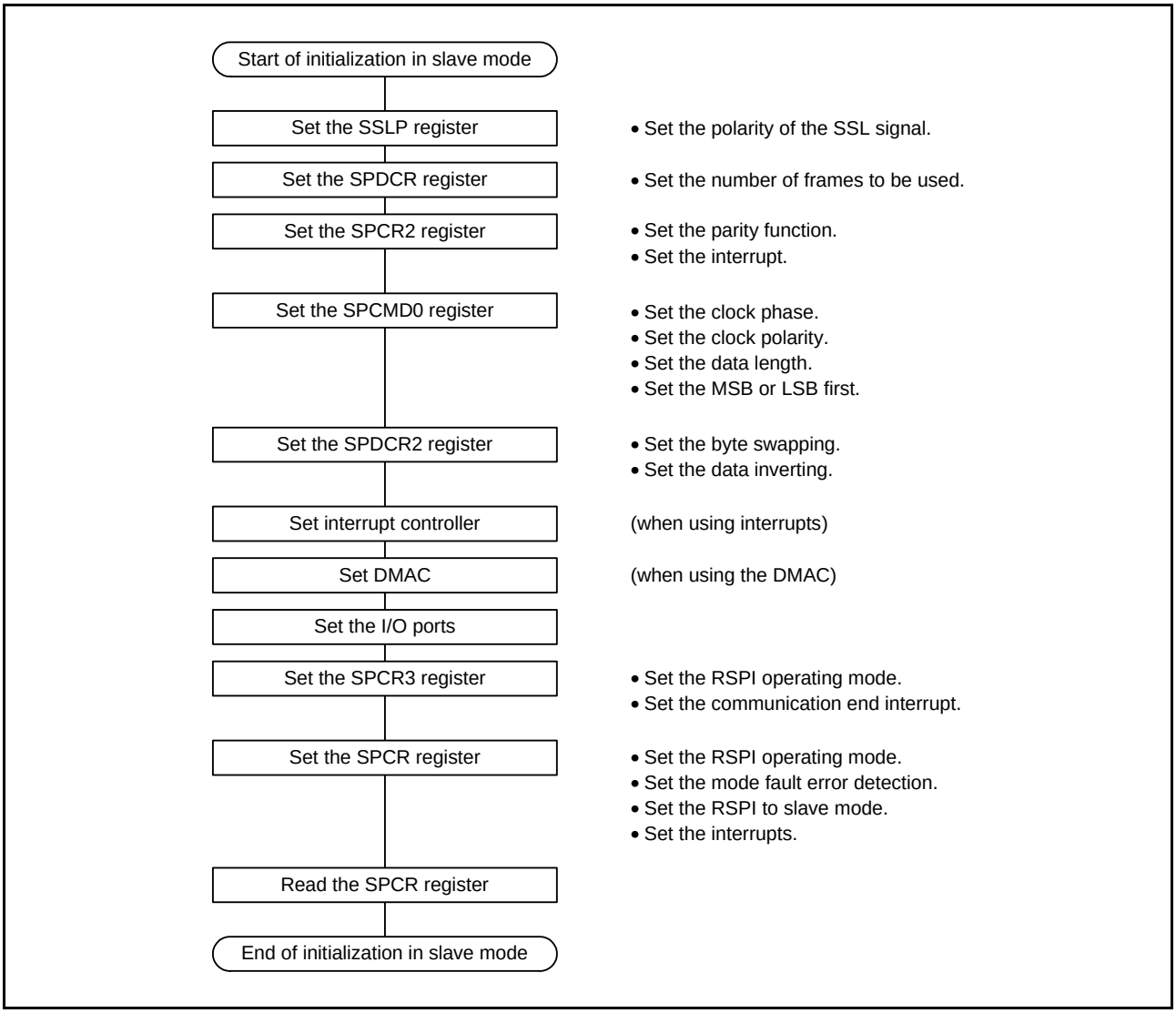


Figure 40.43 Example of Initialization Flowchart in Slave Mode (SPI Operation)

(6) Software Processing Flow

Figure 40.45 to Figure 40.47 show examples of the flow of software processing.

(a) Communication Preprocessing Flow

Before starting communications, clear the error flags and disable the idle interrupt and communication end interrupt. Then enable the RSPI function and the required interrupts at the same time.

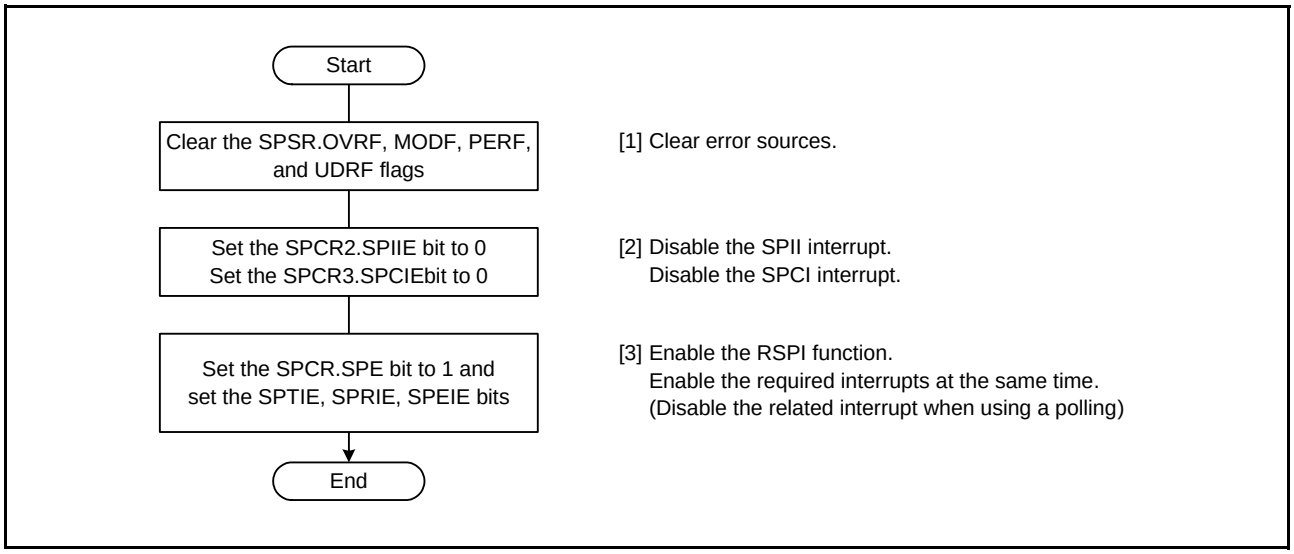


Figure 40.44 Flowchart in Slave Mode (Communication Preprocess)

(b) Transmit Processing Flow

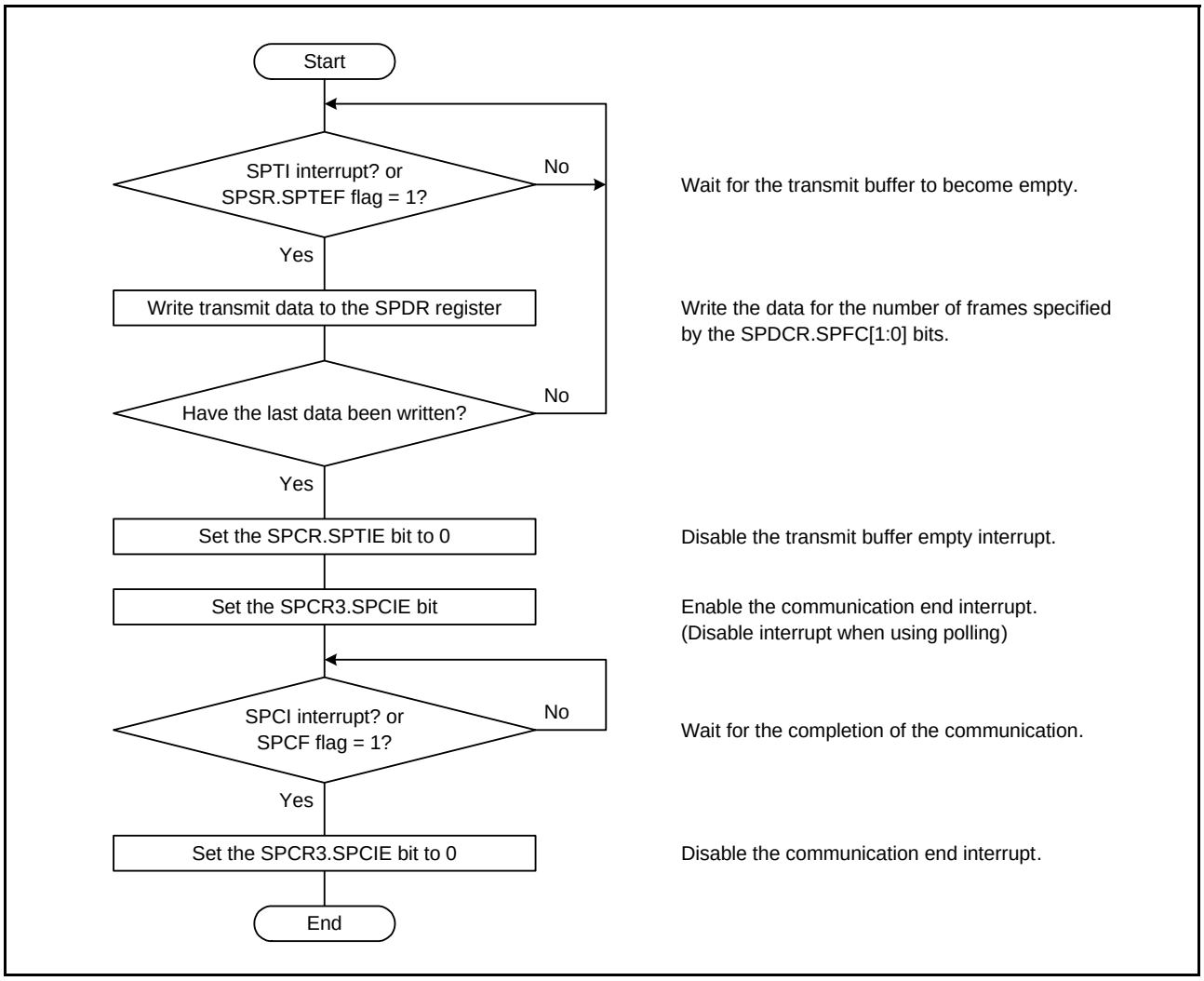


Figure 40.45 Flowchart in Slave Mode (Transmission)

(c) Receive Processing Flow

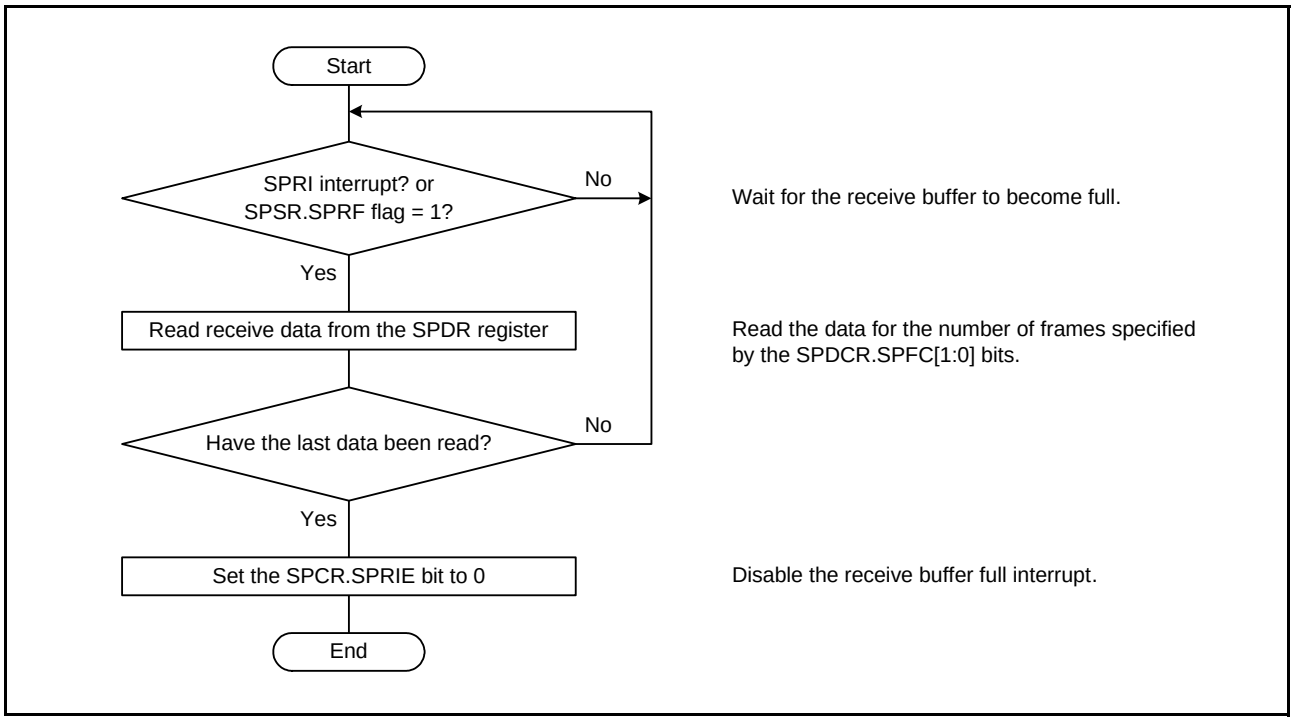


Figure 40.46 Flowchart in Slave Mode (Reception)

(d) Flow of Error Processing

In slave mode, even when a mode fault error is generated, the SPSR.MODF flag can be cleared regardless of the status of the SSLx0 pin.

When interrupts are used and an error occurs, if the ICU.IRn.IR flag for the SPTI or SPRI interrupt request is set to 1, clear the ICU.IRn.IR flag in the error processing routine. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPI.

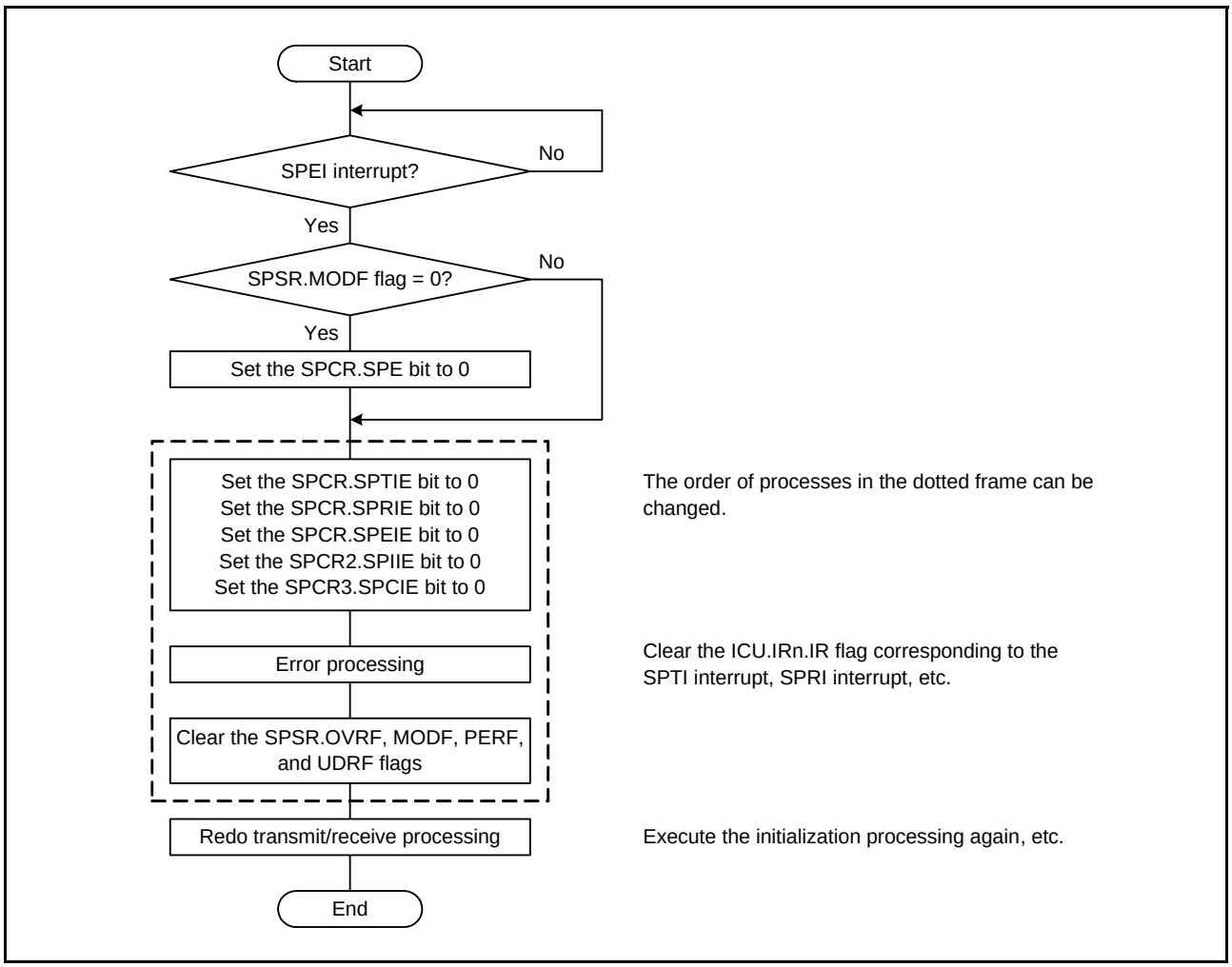


Figure 40.47 Flowchart for Slave Mode (Error Processing)

40.3.13 Clock Synchronous Operation

Setting the SPCR.SPMS bit to 1 selects clock synchronous operation of the RSPI. In clock synchronous operation, the SSLxi pin is not used, and the three pins of RSPCKx, MOSIx, and MISOx handle communications. The SSLxi pin is available as I/O port pins.
Although clock synchronous operation does not require use of the SSLxi pin, operation of the module is the same as in SPI operation. That is, in both master and slave mode operations, communications can be performed with the same flow as in SPI operation. However, mode fault errors are not detected because the SSLxi pin is not used.
Furthermore, do not set the SPCMDm.CPHA bit to 0 if clock synchronous operation is to proceed in slave mode (SPCR.MSTR = 0).

40.3.13.1 Master Mode Operation

(1) Starting a Serial Transfer

The RSPI updates the data in the transmit buffer (SPTX) of the SPDR register when data is written to the SPDR register with the transmit buffer being empty (the SPTEF flag is 1 and data for the next transfer is not set). When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits are written to the SPDR register, the RSPI copies data from the transmit buffer to the shift register and starts serial transmission. Upon copying transmit data to the shift register, the RSPI changes the status of the shift register to “full”, and upon termination of serial transfer, it changes the status of the shift register to “empty”. The status of the shift register cannot be referenced.
For details on the RSPI transfer format, refer to section 40.3.5, Transfer Format.
However, transfer in clock synchronous operation is conducted without the SSLx0 output signal.

(2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after transmitting an RSPCKx edge corresponding to the sampling timing. If free space is available in the receive buffer (SPRX) (the SPRF flag is 0), upon termination of serial transfer, the RSPI copies data from the shift register to the receive buffer of the RSPI data register (SPDR).
It should be noted that the final sampling timing varies depending on the bit length of transfer data. In master mode, the RSPI data length depends on the SPCMDm.SPB[3:0] bit setting.
For details on the RSPI transfer format, refer to section 40.3.5, Transfer Format.
However, transfer in clock synchronous operation is conducted without the SSLx0 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 SSLxi signals are not output in clock synchronous operation, these settings are valid.
The SPSCR register 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 the SPCMDm register: SSLxi output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCKx polarity/phase, whether the SPCKD register is to be referenced, whether the SSLND register is to be referenced, and whether the SPND register is to be referenced. The SPBR register holds some of the bit rate settings; the SPCKD register, an RSPI clock delay value; the SSLND register, an SSL negation delay; and the SPND register, a next-access delay value.
According to the sequence length that is assigned to the SPSCR register, the RSPI makes up a sequence comprised of a part or all of the 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 the 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 the SPCMD0 register, and in this manner the sequence is executed repeatedly.

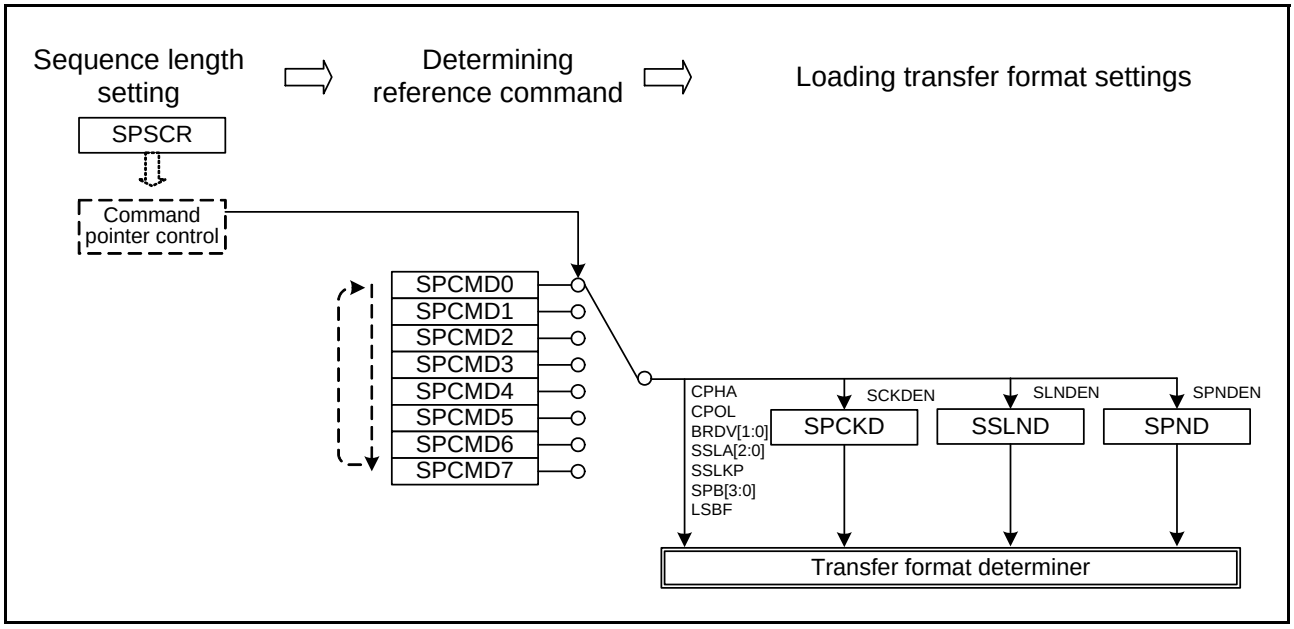


Figure 40.48 Procedure for Determining the Form of Serial Transmission in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

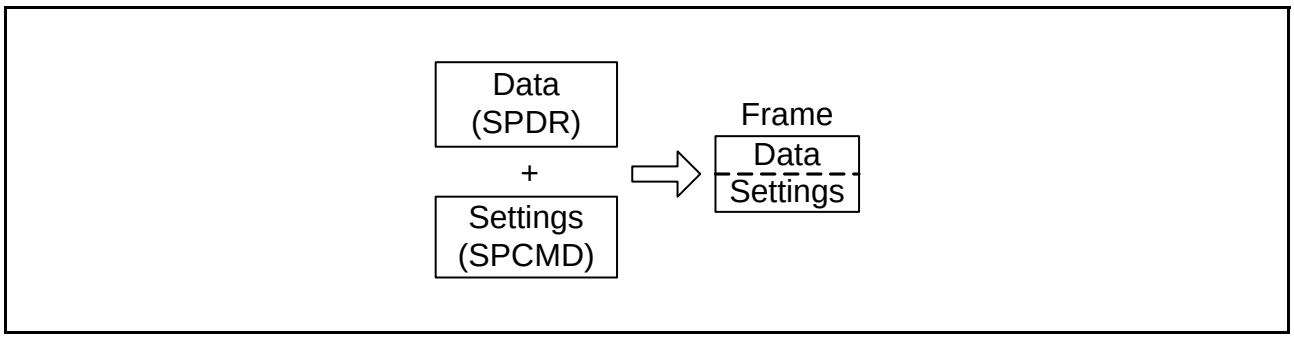


Figure 40.49 Concept of a Frame

Figure 40.50 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 40.4.

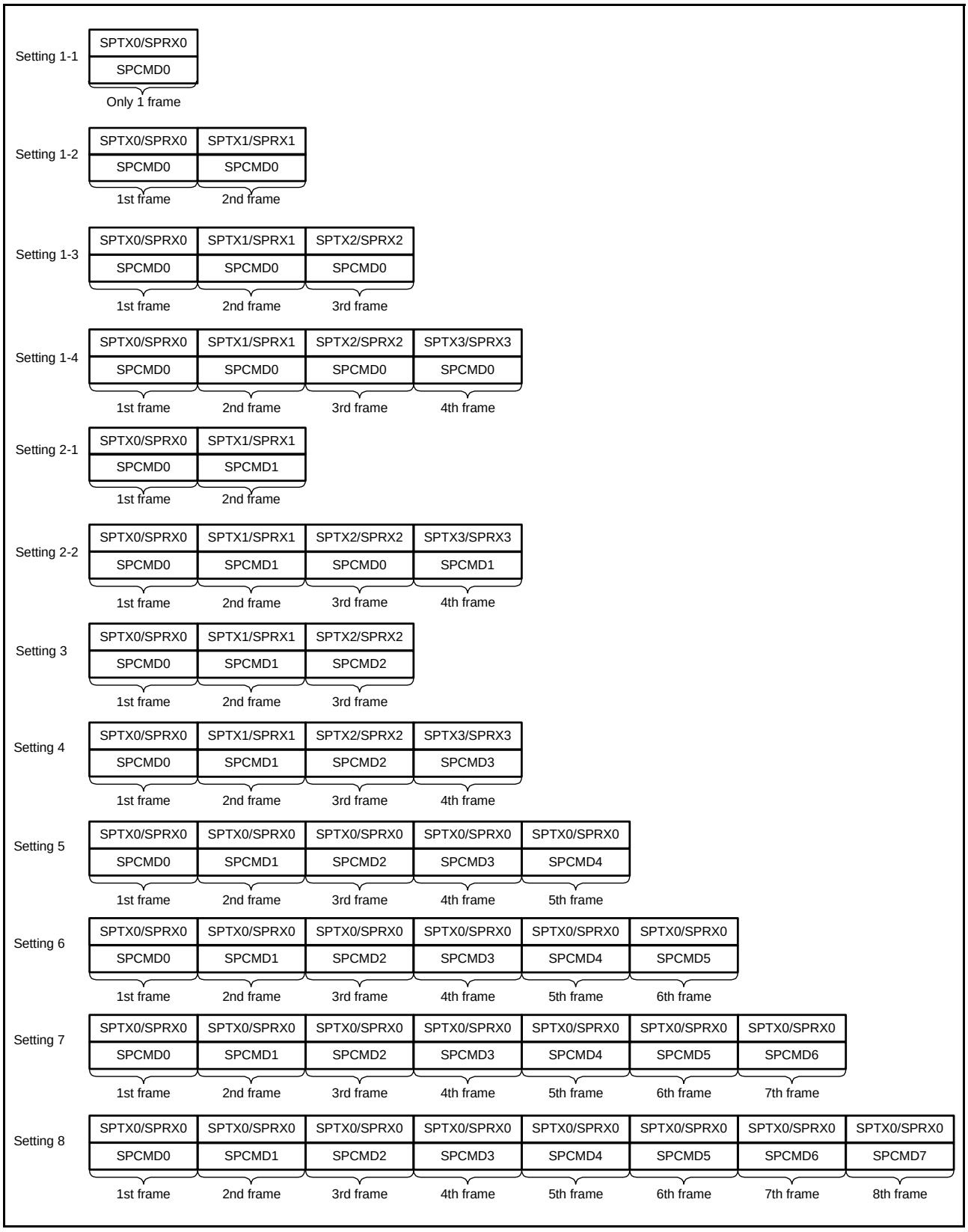


Figure 40.50 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations

(4) Initialization Flowchart

Figure 40.51 is a flowchart illustrating an example of initialization in clock synchronous operation when the RSPI is used in master mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

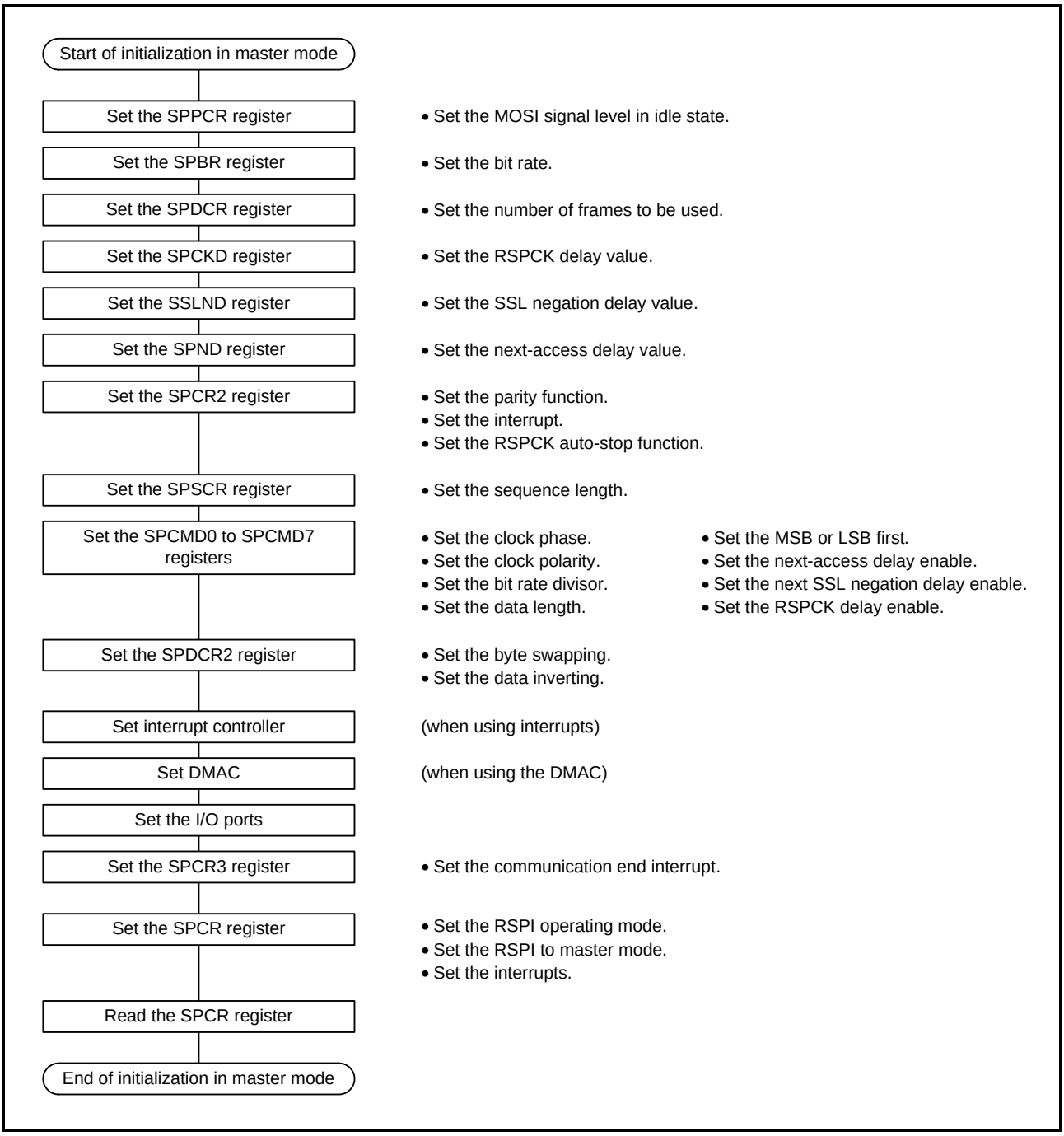


Figure 40.51 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation)

(5) Flow of Software Processing

Software processing during clock-synchronous operation when the RSPI is used in master mode is the same as that for SPI master mode operation. For details, refer to section 40.3.12.1, (9) Software Processing Flow. Note that mode fault errors will not occur.

40.3.13.2 Slave Mode Operation

(1) Starting a Serial Transfer

When the SPCR.SPMS bit is 1, the first RSPCKx edge triggers the start of a serial transfer in the RSPI.
When the SPMS bit is 1, the RSPI drives the MISOx output signal.
For details on the RSPI transfer format, refer to section 40.3.5, Transfer Format.
It should be noted that the SSLx0 input signal is not used in clock synchronous operation.

(2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after detecting an RSPCKx edge corresponding to the final sampling timing. When free space is available in the receive buffer (the SPRF flag is 0), upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the SPDR register. Upon termination of a serial transfer the RSPI changes the status of the shift register to “empty” regardless of the receive buffer status. The final sampling timing changes depending on the bit length of transfer data. In slave mode, the RSPI data length depends on the SPCMD0.SPB[3:0] bit setting.
For details on the RSPI transfer format, refer to section 40.3.5, Transfer Format.

(3) Initialization Flowchart

Figure 40.52 is a flowchart illustrating an example of initialization in clock synchronous operation when the RSPI is used in slave mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

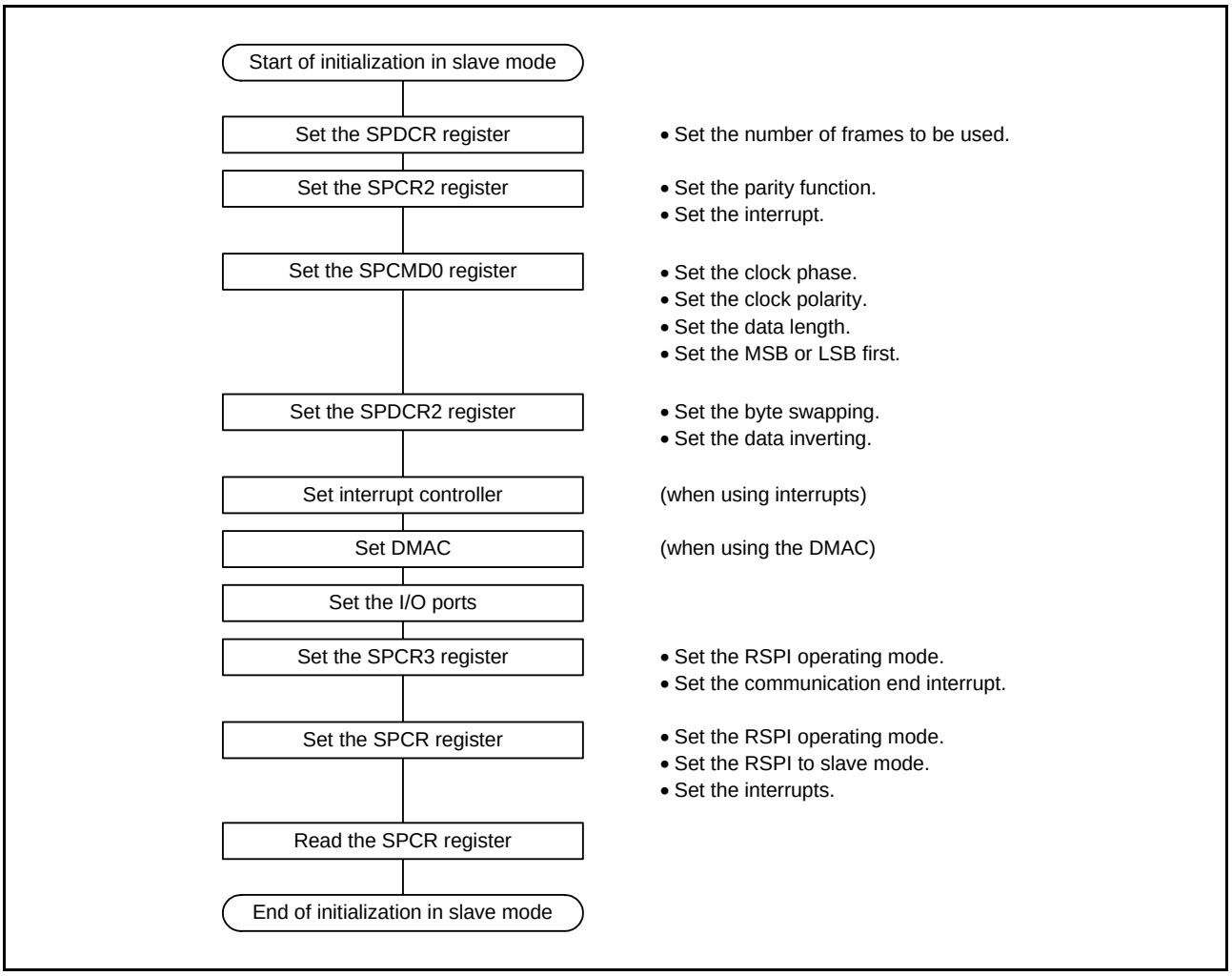


Figure 40.52 Example of Initialization Flowchart in Slave Mode (Clock Synchronous Operation)

(4) Flow of Software Processing

Software processing during clock-synchronous operation when the RSPI is used in slave mode is the same as that for SPI slave mode operation. For details, refer to section 40.3.12.2, (6) Software Processing Flow. Note that mode fault errors will not occur.

40.3.14 Loopback Mode

When 1 is written to the SPPCR.SPLP2 bit or SPPCR.SPLP bit, the RSPI shuts off the path between the MISOx pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSIx 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 MOSIx pin and the shift register if the SPCR.MSTR bit is 1, and between the MISOx 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 40.11 lists the relationship among the SPLP2 and SPLP bits and the received data. Figure 40.53 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 40.11 SPLP2 and SPLP Bit Settings and Received Data

SPPCR.SPLP2 Bit	SPPCR.SPLP Bit	Received Data
0	0	Input data from the MOSIx pin or MISOx pin
0	1	Inverted transmit data
1	0	Transmit data
1	1	Transmit data

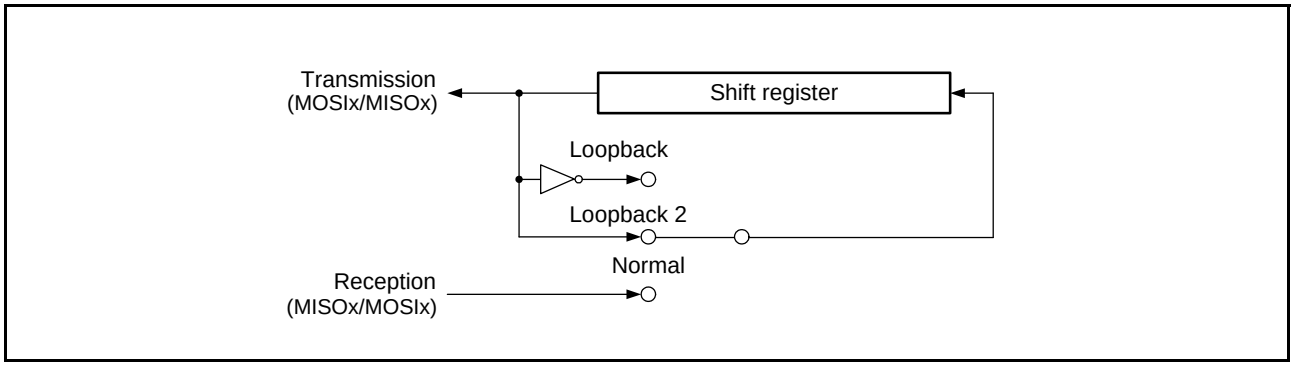


Figure 40.53 Configuration of Shift Register I/O Paths in Loopback Mode (Master Mode)

40.3.15 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 40.54.

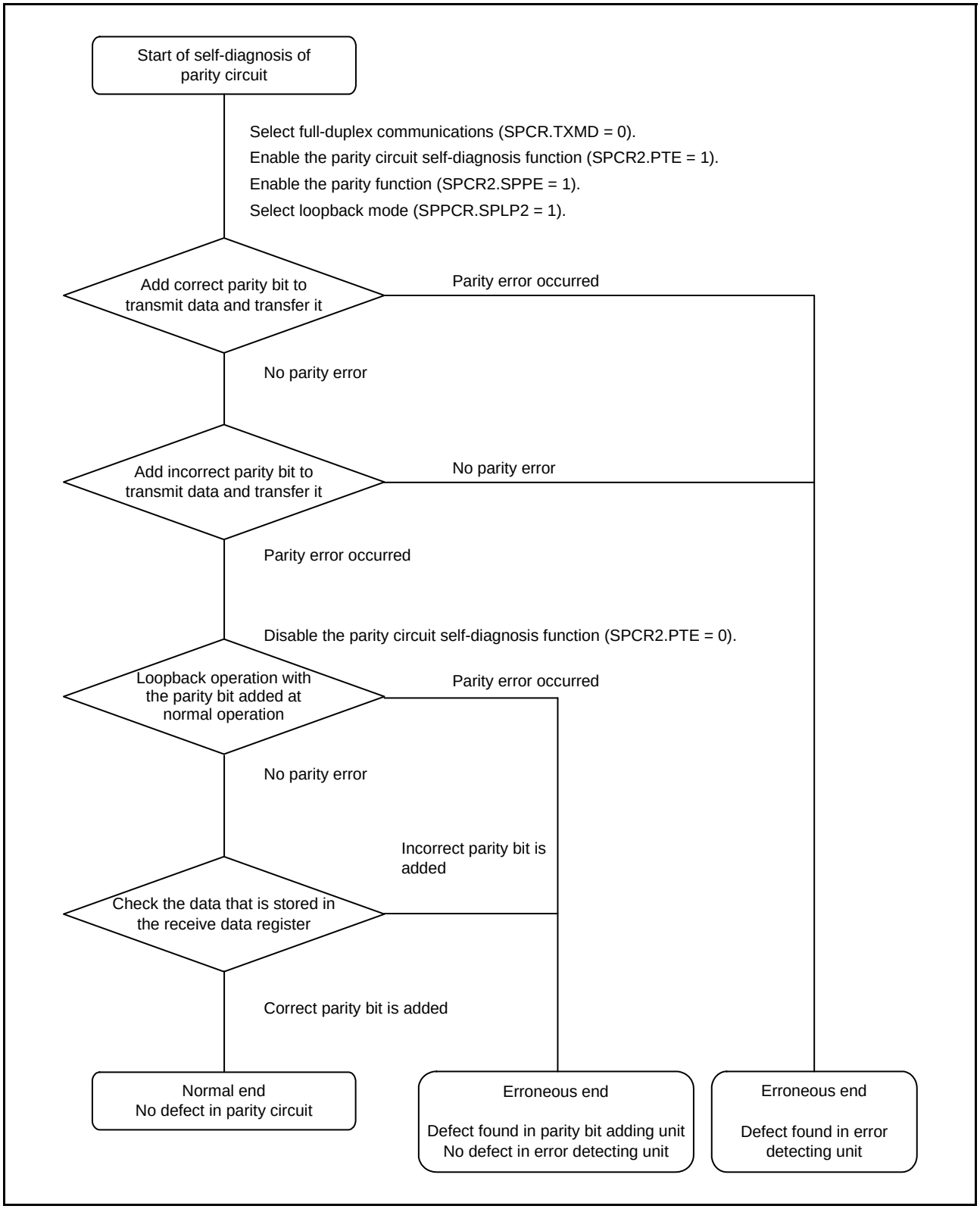


Figure 40.54 Flowchart for Self-Diagnosis of Parity Circuit

40.3.16 Interrupt Sources

The RSPI has interrupt sources of receive buffer full, transmit buffer empty, error (mode fault, underrun, overrun, and parity error), idle, and communication end. 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, underrun, overrun, and parity errors, the actual interrupt source must be determined from the flags. Interrupt sources for the RSPI are listed in Table 40.12. An interrupt is generated on satisfaction of an interrupt condition in Table 40.12. Clear the receive buffer full and transmit buffer empty sources through data transfer.

When using the DTC or DMAC to perform data transmission/reception, the DTC or DMAC must be set up first to be in a status in which transfer is enabled before making the RSPI settings. For the method for setting the DTC or DMAC, refer to section 18, DMA Controller (DMACAb), or section 20, Data Transfer Controller (DTCb).

If the conditions for generating a transmit buffer empty or receive buffer full interrupt are generated while the ICU.IRn.IR flag is 1, the interrupt is not output as a request for ICU but is retained internally (the capacity for retention is one request per source). A retained interrupt request is output when the ICU.IRn.IR flag becomes 0. A retained interrupt request is automatically discarded once it is output as an actual interrupt request. The interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) for an internally retained interrupt request can also be cleared to 0.

Table 40.12 Interrupt Sources of RSPI

Interrupt Source	Symbol	Interrupt Condition	DMAC/DTC Activation
Receive buffer full	SPRI	The receive buffer becomes full (the SPRF flag becomes 1) while the SPCR.SPRIE bit is 1.	Possible
Transmit buffer empty	SPTI	The transmit buffer becomes empty (the SPTEF flag becomes 1) while the SPCR.SPTIE bit is 1.	Possible
Errors (mode fault, underrun, overrun, and parity error)	SPEI	The SPSR.MODF, UDRF, OVRF, or PERF flag is set to 1 while the SPCR.SPEIE bit is 1.	Impossible
Idle	SPII	The SPSR.IDLNF flag is set to 0 while the SPCR2.SPIIE bit is 1.	Impossible
Communication end	SPCI	The SPSR.SPCF flag is set to 1 while the SPCR3.SPCIE bit is 1.	Impossible

40.4 Link Operation by Event Linking

The RSPI0 supports the following event output for the event link controller (ELC). The event link output signal is output regardless of the interrupt enable bit setting.

40.4.1 Receive Buffer Full Event Output

This event signal is output when received data have been transferred from the shift register to the SPDR register on completion of serial transfer.

40.4.2 Transmit Buffer Empty Event Output

This event signal is output when data for transmission have been transferred from the transmit buffer to the shift register and when the value of the SPE bit has changed from 0 to 1.

40.4.3 Mode Fault, Underrun, Overrun, or Parity Error Event Output

(1) Mode Fault

Table 40.13 lists the occurrence conditions of a mode fault event.

Table 40.13 Occurrence Conditions of Mode Fault Event

	SPCR.MODFEN Bit	SSLx0 Pin	Remarks
Master (SPCR.MSTR bit = 1)	1	Active	Under the condition (the SPCR.MSTR bit is 1 and the MODFEN bit is 1), if the SPCR.SPMS bit is 0, mode fault error, overrun error, and parity error event output cannot be used. Do not set the ELSRn register to 52h.
Slave (SPCR.MSTR bit = 0)	1	Not active	Event is output only when the pin is deactivated during transmission.

(2) Underrun

The condition for this event signal being output in response to an underrun is the start of serial transfer with the transmit buffer containing no transmit data while the value of the SPCR.MSTR bit is 0 and the value of the SPCR.SPE bit is 1, in which case the UDRF and MODF flags are set to 1.

(3) Overrun

The condition for this event signal being output in response to an overrun is completion of serial transfer while the receive buffer contains data that have not been read and the value of the SPCR.TXMD bit is 0, in which case the OVRF flag is set to 1.

(4) Parity Error

The condition for this event signal being output in response to a parity error is detection of a parity error on completion of serial transfer while the value of the TXMD bit in SPCR is 0 and the value of the SPPE bit in SPCR2 is 1.

40.4.4 Idle Event Output

(1) In Master Mode

In master mode, an event is output when the condition for setting the IDLNF flag (idle flag) to 0 is satisfied.

(2) In Slave Mode

In slave mode, an event is output when the SPCR.SPE bit is set to 0 (RSPI is initialized).

40.4.5 Communication End Event Output

During both SPI operation and clock synchronous operation in master mode, an event is output under the condition for setting the IDLNF flag (idle flag) from 1 to 0. In slave mode, an event is output under the conditions listed in Table 40.14 and Table 40.14.

Table 40.14 Generating Conditions of Communication End Event (Slave mode, full-duplex or transmit-only simplex communications)

RSPI Mode	Transmit Buffer State	Shift Register State	Others
SPI operation (SPMS = 0)	Empty	Empty	Negation of the SSLx0 input
Clock synchronous operation (SPMS = 1)	Empty	Empty	Even edge detection of the last RSPCKx for the last data

Table 40.15 Generating Conditions of Communication End Event (Slave mode, receive-only simplex communications)

RSPI Mode	Condition
SPI operation (SPMS = 0)	Negation of the SSLx0 input after the last data was received
Clock synchronous operation (SPMS = 1)	Even edge detection of the last RSPCKx for the last data

Whether the operation is in master mode or slave mode, an event is not output if 0 is written to the SPCR.SPE bit during communication or the SPCR.SPE bit is cleared by the mode fault error.

40.5 Usage Notes

40.5.1 Setting Module Stop Function

Module stop control register B (MSTPCRB) and module stop control register C (MSTPCRC) can be used to enable or disable the RSPI. Immediately after a reset, operation of the RSPI is disabled. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

40.5.2 Note on Low Power Consumption Functions

When using the module stop function and entering a low power consumption mode other than sleep mode, set the SPCR.SPE bit to 0 before completing communication.

40.5.3 Notes on Starting Transfer

If the ICU.IRn.IR flag is 1 at the time transfer is to be started, an interrupt request is internally retained after transfer starts, and this can lead to unanticipated behavior of the ICU.IRn.IR flag.
When the ICU.IRn.IR flag is 1 at the time transfer is to start, follow the procedure below to clear interrupt requests before enabling operations (by setting the SPCR.SPE bit to 1).

1. Confirm that transfer has stopped (i.e. that the SPCR.SPE bit is 0).
2. Set the relevant interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) to 0.
3. Read the relevant interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) and confirm that its value is 0.
4. Set the ICU.IRn.IR flag to 0.

40.5.4 Notes on the SPRF and SPTEF flags

When polling the SPSR.SPRF flag and/or SPSR.SPTEF flag, set the SPCR.SPRIE bit and/or SPCR.SPTIE bit to 0.

41. Serial Peripheral Interface (RSPIA)

41.1 Overview

The serial peripheral interface (RSPIA) supports high-speed, full-duplex or simplex synchronous serial communications with multiple processors and peripheral devices.

41.1.1 Features

The RSPIA has the following features.

Table 41.1 RSPIA Specifications (1/2)

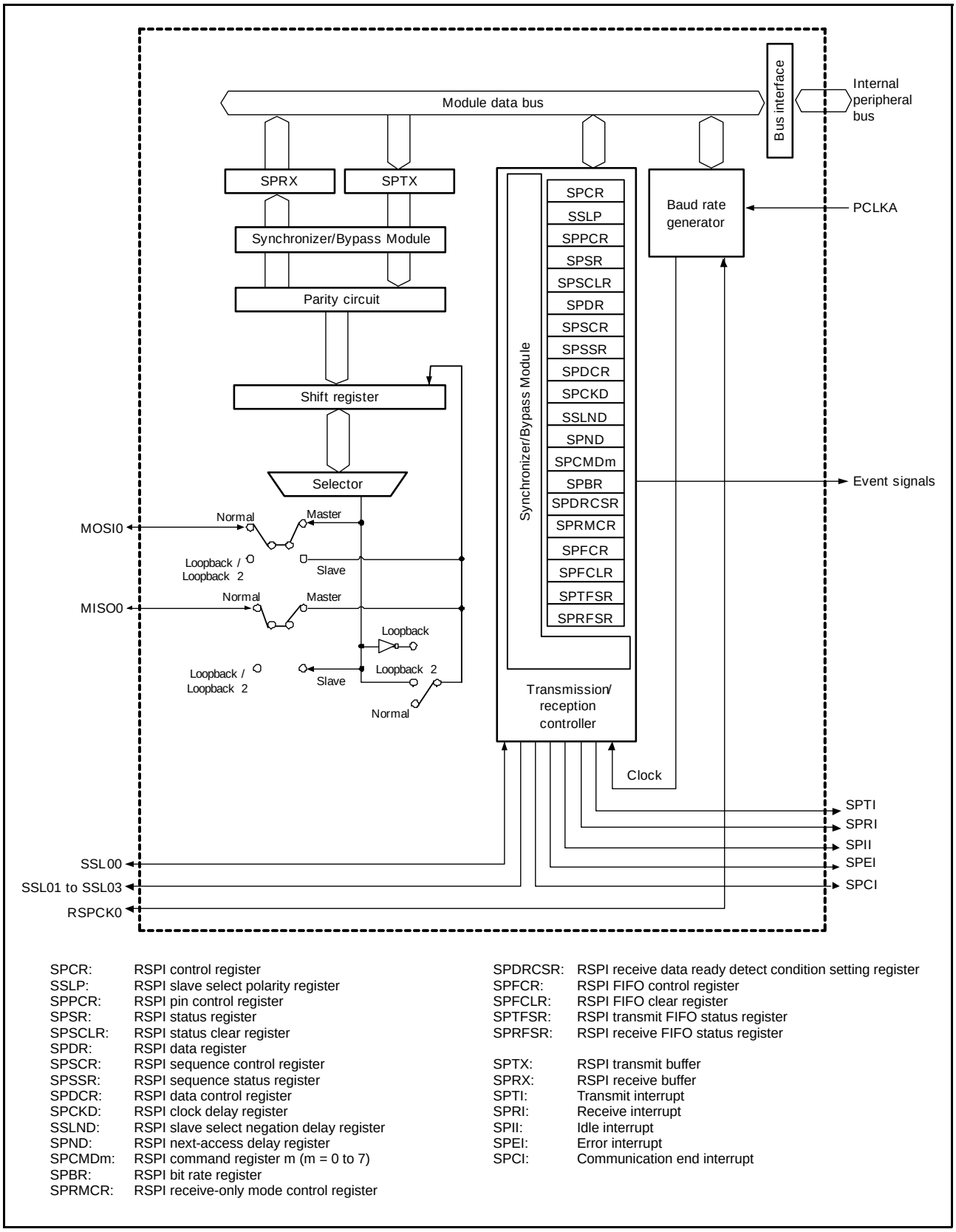
Item	Description
Transfer functions	• Use of MOSI (master out/slave in), MISO (master in/slave out), SSL (slave select), and RSPCK (RSPI clock) signals allows serial communications through SPI operation (4-wire method) or clock synchronous operation (3-wire method). • Full-duplex or simplex (transmit-only or receive-only) communications can be selected. • Serial communication is possible in master mode and slave mode. • Variable serial transfer clock polarity • Variable serial transfer clock phase
Data format	• MSB first/LSB first selectable • Transfer bit length can be changed to 4 through 32 bits. • Transmit buffer size/receive buffer size: 32 bits × 4 stages FIFO • Up to four frames can be transferred in one round of transmission/reception (each frame consisting of up to 32 bits). • Byte swapping of transmit and receive data is selectable. • Logic level of transmit and receive data can be inverted.
Bit rate	• In master mode, the on-chip baud rate generator generates RSPCK by frequency-dividing PCLKA. The division ratio ranges from divided by 2 to divided by 4096. • In slave mode, an external input clock is used as a serial clock. The maximum frequency of RSPCK is that of PCLKA divided by 2 (width at high level: 1 cycle of PCLKA; width at low level: 1 cycle of PCLKA).
Buffer configuration	• Transmit buffer and receive buffer are configured independently.
Error detection	• Mode fault error detection • Underrun error detection • Overrun error detection • Parity error detection • Receive data ready detection
SSL control function	[Motorola SPI mode] • Four SSL signals per RSPIA channel (SSL00 to SSL03) • In single-master mode, the SSL00 to SSL03 signals are output. • In multi-master mode: The SSL00 signal is input and the SSL01 to SSL03 signals are output or Hi-Z. • In slave mode: The SSL00 signal is input and the SSL01 to SSL03 signals are Hi-Z (not used). • 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

Table 41.1 RSPIA Specifications (2/2)

Item	Description
SSL control function	[TI SSP mode] • Four SSL signals per RSPIA channel (SSL00 to SSL03) • In single-master mode, the SSL00 to SSL03 signals are output. • In multi-master mode: The SSL00 signal is input and the SSL01 to SSL03 signals are output or Hi-Z. • In slave mode: The SSL00 signal is input and the SSL01 to SSL03 signals are Hi-Z (not used). • Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 0 to 8 RSPCK cycles (set in RSPCK-cycle units) • Controllable delay from RSPCK stop to OE output negation (OE 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
Communication Protocol	• Motorola SPI • TI SSP (synchronous serial protocol)
Control in master transfer	[Motorola SPI mode] • 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, transmit data length, MSB/LSB first, burst, RSPCK delay, SSL negation delay, and next-access delay • A transfer can be initiated by writing to the transmit buffer. • MOSI signal value specifiable in SSL negation • RSPCK auto-stop function (enabled or disabled by setting) • Delay between data bytes under burst transfer can be reduced. [TI SSP mode] • 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, transmit data length, MSB/LSB first, burst, RSPCK delay, SSL negation delay (OE negation delay), and next-access delay • A transfer can be initiated by writing to the transmit buffer. • MOSI signal value specifiable in SSL negation • RSPCK auto-stop function (enabled or disabled by setting) • Delay between data bytes under burst transfer can be reduced.
Interrupt sources	Five interrupt sources • Receive buffer full/receive data ready interrupt • Transmit buffer empty interrupt • Communication end interrupt • Error interrupt (mode fault error, underrun error, overrun error, parity error, or receive data ready) • Idle interrupt
Event link function	Five event types can be output to the event link controller. • Receive buffer full/receive data ready event signal • Transmit buffer empty event signal • Mode fault error, underrun error, overrun error, parity error, or receive data ready event signal • Idle event signal • Communication end event signal
Others	• Function for disabling (initializing) the RSPI • Loopback mode • SPE bit status polling function
Low power consumption function	Module stop state can be set.

41.1.2 Block Diagram

Figure 41.1 shows a block diagram of the RSPIA.



SPCR: RSPI control register
SSLP: RSPI slave select polarity register
SPPCR: RSPI pin control register
SPSR: RSPI status register
SPSCLR: RSPI status clear 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 negation delay register
SPND: RSPI next-access delay register
SPCMDm: RSPI command register m (m = 0 to 7)
SPBR: RSPI bit rate register
SPRMCR: RSPI receive-only mode control register
SPDRCSR: RSPI receive data ready detect condition setting register
SPFCR: RSPI FIFO control register
SPFCLR: RSPI FIFO clear register
SPTFSR: RSPI transmit FIFO status register
SPRFSR: RSPI receive FIFO status register

SPTX: RSPI transmit buffer
SPRX: RSPI receive buffer
SPTI: Transmit interrupt
SPRI: Receive interrupt
SPII: Idle interrupt
SPEI: Error interrupt
SPCI: Communication end interrupt

Figure 41.1 RSPIA Block Diagram

41.1.3 Pin Configuration

The RSPIA has serial pins listed in Table 41.2 for each channel. The RSPIA automatically switches the I/O direction of the SSL00 pin. SSL00 is set as an output when the RSPIA is a single master and as an input when the RSPIA is a multi-master or a slave. Pins RSPCK0, MOSI0, and MISO0 are automatically set as inputs or outputs according to the setting of master or slave, and the SPI operation (4-wire method) or clock synchronous operation (3-wire method) level input on the SSL00 pin. Refer to section 41.3.2, Controlling RSPI Pins for details.

Table 41.2 RSPIA Pin Configuration

Pin Name	Abbreviation	I/O	Function
RSPI clock	RSPCK0	I/O	Clock I/O
Master transmit data	MOSI0	I/O	Master transmit data I/O
Slave transmit data	MISO0	I/O	Slave transmit data I/O
Slave select 0	SSL00	I/O	Slave selection I/O
Slave select 1	SSL01	Output	Slave selection output
Slave select 2	SSL02	Output	Slave selection output
Slave select 3	SSL03	Output	Slave selection output

41.2 Register Descriptions

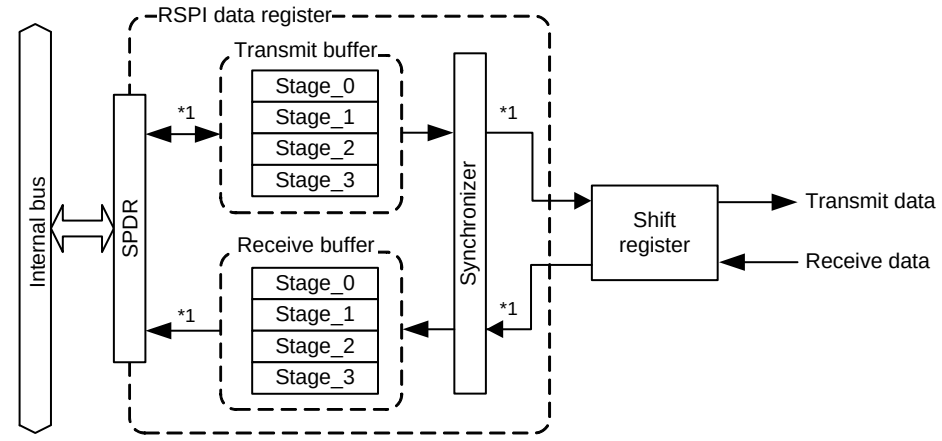
41.2.1 RSPI Data Register (SPDR)

Address(es): RSPIA0.SPDR 000E 2800h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

The SPDR register is the interface with the buffers that hold data for transmission and reception by the RSPI. The transmit buffer and receive buffer are independent. Figure 41.2 shows the configuration of the SPDR register.



Note 1. The destination buffer and stage for access is automatically switched by hardware.

Figure 41.2 Configuration of SPDR Register

The transmit and receive buffers each have 32 bits × 4 stages of FIFO. The eight stages of FIFO are all mapped to the single address of the SPDR register. Data written to the SPDR register are written to a transmit-buffer stage (SPTXn, n = 0 to 3) and then transmitted from the buffer. The receive buffer (SPRXn, n = 0 to 3) holds received data on completion of reception. The receive buffer is not updated if an overrun is generated.

(1) Bus Interface

The SPDR register is the interface with 32-bit wide transmit buffer (SPTXn, n = 0 to 3) and receive buffer (SPRXn, n = 0 to 3), each of which has four stages, for a total of 32 bytes. In other words, the 32 bytes are mapped to the 4-byte address space for the SPDR register.
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 the SPDR register 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 the SPDR register. Figure 41.3 shows the configuration of the bus interface with the transmit buffer in the case of writing to the SPDR register.

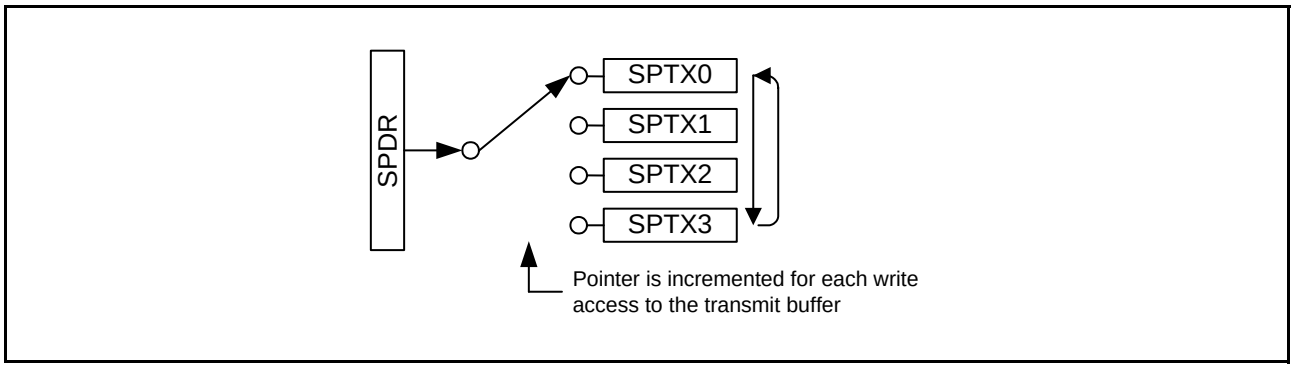


Figure 41.3 Configuration of SPDR Register (Writing)

Sequence of switching the pointer among SPTX0 to SPTX3:
SPTX0 → SPTX1 → SPTX2 → SPTX3 → SPTX0 → SPTX1 → …

When writing to the transmit buffer (SPTXn) after generation of the transmit buffer empty interrupt (after the SPSR.SPTEF flag becomes 1), write the number of frames + 1 set by the transmission FIFO threshold setting bits of RSPI FIFO control register (SPFCR.TTRG[1:0]). Writing to the transmit buffer (SPTXn, n = 0 to 3) in the state where there is no empty stages in the transmit FIFO does not update the buffer value.

(b) Reading

The SPDR register can be read to read 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 receive/transmit data select bit in the RSPI data control register (SPDCR.SPRDTD) selects whether reading is of the receive or transmit buffer.
The sequence of reading the SPDR register is controlled by independent pointers, receive buffer read pointer and transmit buffer read pointer.
Figure 41.4 shows the configuration of the bus interface with the receive and transmit buffers in the case of reading from the SPDR register.

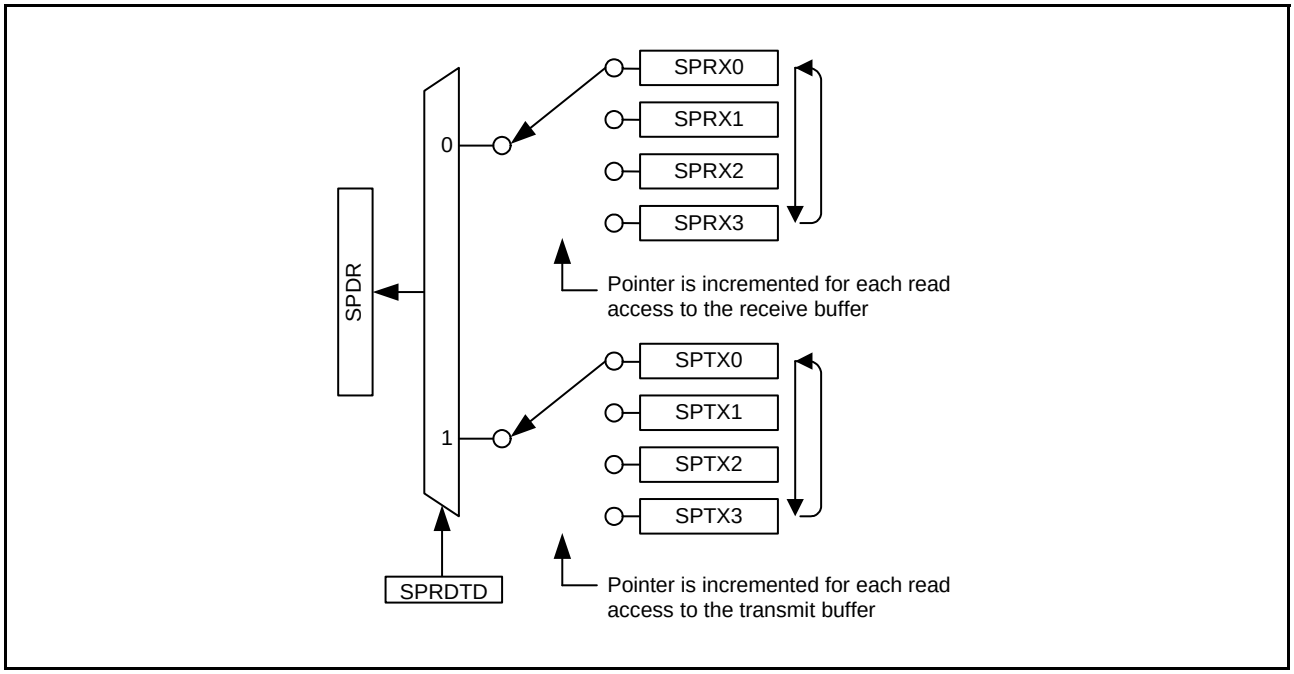


Figure 41.4 Configuration of SPDR (Reading)

Reading the receive buffer switches the receive buffer read pointer to the next buffer automatically.
The sequence of switching the receive buffer read pointer is the same as that for the transmit buffer write pointer.
The transmit buffer read pointer is updated when writing to the SPDR register, and not updated when reading from the transmit buffer. When reading from the transmit buffer, the value most recently written to the SPDR register is read.

41.2.2 RSPI Clock Delay Register (SPCKD)

Address(es): RSPIA0.SPCKD 000E 2804h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SCKDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SCKDL[2:0]	RSPCK Delay Setting	b2 b0 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

[Motorola SPI]
The SPCKD register sets a period from the beginning of SSL0n signal assertion to RSPCK oscillation (RSPCK delay) when the SPCMDm.SCKDEN bit is 1. If the value of the SPCKD register is changed while both the SPCR.MSTR and SPCR.SPE bits are 1, the operation after the change is not guaranteed.

[TI SSP]
The SPCKD register sets a period from the beginning of SSL0n signal assertion to RSPCK oscillation (RSPCK delay) when the SPCMDm.SCKDEN bit is 1, and also sets a period until the SSL0n signal is negated. If the value of the SPCKD register is changed while both the SPCR.MSTR and SPCR.SPE bits are 1, the operation after the change is not 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 RSPIA in slave mode, set the SCKDL[2:0] bits to 000b.

41.2.3 RSPI Slave Select Negation Delay Register (SSLND)

Address(es): RSPIA0.SSLND 000E 2805h

	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	[In master mode] 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 [In slave mode, TI SSP] b2 b0 0 0 0: 1 PCLKA 0 0 1: 2 PCLKA 0 1 0: 3 PCLKA 0 1 1: 4 PCLKA 1 0 0: 5 PCLKA 1 0 1: 6 PCLKA 1 1 0: 7 PCLKA 1 1 1: 8 PCLKA	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

[Motorola SPI]
The SSLND register sets a period (SSL negation delay) from the transmission of a final RSPCK edge to the negation of the SSL0n signal during a serial transfer by the RSPIA in master mode when the SPCMDm.SLNDEN bit is 1. If the value of the SSLND register is changed while both the SPCR.MSTR and SPCR.SPE bits are 1, the operation after the change is not guaranteed.

[TI SSP]
The SSLND register sets a period (SSL negation delay) from the transmission of a final RSPCK edge to the negation of the OE signal during a serial transfer by the RSPIA in master mode, or from the detection of a final RSPCK edge to the negation of the OE signal during a serial transfer by the RSPIA in slave mode. If the value of the SSLND register is changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

SLNDL[2:0] Bits (SSL Negation Delay Setting)

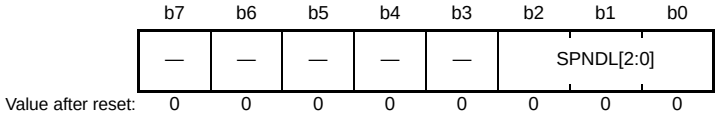
[Motorola SPI]
The SLNDL[2:0] bits set an SSL negation delay value when the SPCMDm.SLNDEN bit is 1.
When using the RSPIA in slave mode, set the SLNDL[2:0] bits to 000b.

[TI SSP]
The SLNDL[2:0] bits set an OE negation delay value when the SPCMDm.SLNDEN bit is 1.

[Clock synchronous]
When using the RSPIA in slave mode, set the SLNDL[2:0] bits to 000b.

41.2.4 RSPI Next-Access Delay Register (SPND)

Address(es): RSPIA0.SPND 000E 2806h



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 + 5 PCLKA 0 0 1: 2 RSPCK + 5 PCLKA 0 1 0: 3 RSPCK + 5 PCLKA 0 1 1: 4 RSPCK + 5 PCLKA 1 0 0: 5 RSPCK + 5 PCLKA 1 0 1: 6 RSPCK + 5 PCLKA 1 1 0: 7 RSPCK + 5 PCLKA 1 1 1: 8 RSPCK + 5 PCLKA	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SPND register sets a non-active period (next-access delay) of the SSL0n signal after termination of a serial transfer when the SPCMDm.SPNDEN bit is 1. If the value of the SPND register is changed while both the SPCR.MSTR and SPCR.SPE bits are 1, the operation after the change is not 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 RSPIA in slave mode, set the SPNDL[2:0] bits to 000b.

41.2.5 RSPI Control Register (SPCR)

Address(es): RSPIA0.SPCR 000E 2808h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	SYNDIS	MSTR	CMMD[1:0]		—	—	FRFS	SPMS	—	—	SPCIE	SPTIE	RDRIS	SPIIE	SPRIE	SPEIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	MODFEN	SCKDDIS	SCKASE	PTE	—	SPOE	SPPE	—	—	—	—	—	—	—	SPE
Value after 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	SPE	RSPI Function Enable	0: Disables the RSPI function 1: Enables the RSPI function	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	SPPE	Parity Enable	0: A parity bit is not added to transmit data. Received-data parity check is not performed. 1: A parity bit is added to transmit data. Received-data parity check is performed.	R/W
b9	SPOE	Parity Mode	0: Even parity is used for transmission and reception. 1: Odd parity is used for transmission and reception.	R/W
b10	—	Reserved	This bit is read as 0. The write value should be 0.	R
b11	PTE	Parity Self-Diagnosis	0: Disables the self-diagnosis function of the parity circuit 1: Enables the self-diagnosis function of the parity circuit	R/W
b12	SCKASE	RSPCK Auto-Stop Function Enable	0: Disables the RSPCK auto-stop function 1: Enables the RSPCK auto-stop function	R/W
b13	SCKDDIS	RSPCK Delay Between Data Bytes Disable	0: Delay between frames is inserted in burst transfer 1: Delay between frames is not inserted in burst transfer.	R/W
b14	MODFEN	Mode Fault Error Detection Enable	0: Disables the detection of mode fault error 1: Enables the detection of mode fault error	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R
b16	SPEIE	Error Interrupt Enable	0: Disables the error interrupt 1: Enables the error interrupt	R/W
b17	SPRIE	Receive Buffer Full Interrupt Enable	0: Disables the receive buffer full interrupt 1: Enables the receive buffer full interrupt	R/W
b18	SPIIE	Idle Interrupt Enable	0: Disables the idle interrupt 1: Enables the idle interrupt	R/W
b19	RDRIS	Receive Data Ready Interrupt Select	0: Receive data full interrupt 1: Error interrupt	R/W
b20	SPTIE	Transmit Buffer Empty Interrupt Enable	0: Disables the transmit buffer empty interrupt 1: Enables the transmit buffer empty interrupt	R/W
b21	SPCIE	Communication End Interrupt Enable	0: Disables the communication end interrupt 1: Enables the communication end interrupt	R/W
b23, b22	—	Reserved	These bits are read as 0. The write value should be 0.	R
b24	SPMS	RSPI Mode Select	0: SPI operation (4-wire method) 1: Clock synchronous operation (3-wire method)	R/W
b25	FRFS	Frame Format Select	0: Motorola SPI 1: TI SSP When the SPMS bit is 1 (clock synchronous operation (3-wire method)), the setting of this bit is disabled.	R/W
b27, b26	—	Reserved	These bits are read as 0. The write value should be 0.	R

Bit	Symbol	Bit Name	Description	R/W
b29, b28	CMMD[1:0]	Communication Mode Select	b29 b28 0 0: Transmit-receive mode (full-duplex) 0 1: Transmit-only mode (simplex, receiver disabled) 1 0: Receive-only mode (simplex, transmitter disabled) 1 1: Setting prohibited	R/W
b30	MSTR	RSPI Master/Slave Mode Select	0: Slave mode 1: Master mode	R/W
b31	SYNDIS	Synchronizer Disable	Set this bit to 1.	R/W

The SPCR register is used to specify the operating modes. If the value of the SPPE, SPOE, PTE, SCKASE, SCKDDIS, MODFEN, SPMS, FRFS, CMMD[1:0], MSTR, or SYNDIS bit is changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

SPE Bit (RSPI Function Enable)

This bit enables or disables the RSPI function. Setting this bit to 1 enables the RSPI function. When the SPSR.MODF flag is 1, this bit is set to 0. This bit cannot be set to 1 until the MODF flag is cleared. For details, refer to section 41.3.10, Error Detection.

Setting this bit to 0 disables the RSPI function, and initializes a part of the module function. For details, refer to section 41.3.12, Initializing RSPIA.

SPPE Bit (Parity Enable)

This bit is used to enable or disable the parity function.

SPOE Bit (Parity Mode)

This bit is used to specify even parity or odd parity.

In even parity mode, the parity bit is determined so that the sum of 1 (parity bit + transmit characters or receive characters) becomes an even number. In the same way, in odd parity mode, a parity bit is determined so that the sum of 1 (parity bit + transmit characters or receive characters) becomes an odd number. The SPOE bit is effective only when the SPCR.SPPE bit is set to 1.

PTE Bit (Parity Self-Diagnosis)

This bit is used to enable or disable self-diagnosis of the parity circuit to confirm that the parity function is normal.

SCKASE Bit (RSPCK Auto-Stop Function Enable)

This bit is used to enable or disable the RSPCK auto-stop function. When this function is enabled, the RSPCK stops immediately before an overrun error occurs during data reception in master mode. For details, refer to section 41.3.10.1, Overrun Error.

SCKDDIS Bit (RSPCK Delay Between Data Bytes Disable)

This bit controls whether insert the delay time between the burst transfer frames.

Valid in the master mode (SPCR.MSTR = 1) for frames with the SPCMDm.SSLKP bit set to 1.

This bit should be set to 0 in slave mode (SPCR.MSTR = 0). Table 41.3 shows the usage of SSL delay control between transfer frames. For details, refer to (4) Burst Transfer in section 41.3.13.1, Master Mode Operation.

Table 41.3 Usage of SSL Delay Control between Transfer Frames (Master mode)

Conditions		SPCMDm.SSLKP bit	SPCR.SCKDDIS bit	SSL delay control register *1 (RSPCK delay, SSL negation delay, next access delay)
Non-burst transfers		0	0	It is possible to control each delay of RSPCK delay, SSL negation delay and next access delay
Burst transfer with delay between frames	From the 1st frame to the last previous frame	1	0	
	The last frame	0	0	
Burst transfer with no delay between frames	From the 1st frame to the last previous frame	1	1	• RSPCK delay of the 1st frame • SSL negation delay and next access delay of the last frame
	The last frame	0	1	

Note 1. RSPCK delay is controlled by the SPCKD.SCKDL[2:0] bits and SPCMDm.SCKDEN bit.
SSL negation delay is controlled by the SSLND.SLNDL[2:0] bits and SPCMDm.SLNDEN bit.
Next access delay is controlled by the SPND.SPNDL[2:0] bits and SPCMDm.SPNDEN bit.

Setting	Operation example (Motorola SPI, SCKDDIS = 1)
SPCMD0.SSLKP = 1	Burst transfer/no interframe delay between 0 and 1 (SSL0n keep active)
SPCMD1.SSLKP = 1	Burst transfer/no interframe delay between 1 and 2 (SSL0n keep active)
SPCMD2.SSLKP = 1	Burst transfer/no interframe delay between 2 and 3 (SSL0n keep active)
SPCMD3.SSLKP = 1	Burst transfer/no interframe delay between 3 and 4 (SSL0n keep active)
SPCMD4.SSLKP = 0	Does not perform burst transfer, and once in-activate SSL0n.*1
SPCMD5.SSLKP = 1	Burst transfer/no interframe delay between 5 and 6 (SSL0n keep active)
SPCMD6.SSLKP = 1	Burst transfer/no interframe delay between 6 and 7 (SSL0n keep active)
SPCMD7.SSLKP = 0	Does not perform burst transfer, and once in-activate SSL0n.*1

Note 1. SCKDDIS setting is invalid because it does not perform burst transfer.

MODFEN Bit (Mode Fault Error Detection Enable)

This bit is used to enable or disable detection of a mode fault error. (Refer to section 41.3.10, Error Detection.) The RSPIA determines SSL00 pin input or output direction according to the combination of the MODFEN and MSTR bits (refer to section 41.3.2, Controlling RSPI Pins).

SPEIE Bit (Error Interrupt Enable)

This bit is used to enable or disable an error interrupt request.
When the SPSR.MODF, OVRF, or PERF flag is set to 1 while this bit is 1, an error interrupt request is generated (refer to section 41.3.10, Error Detection).

SPRIE Bit (Receive Buffer Full Interrupt Enable)

This bit is used to enable or disable a receive buffer full interrupt request.

SPIIE Bit (Idle Interrupt Enable)

This bit is used to enable or disable an idle interrupt requests.
When the SPSR.IDLNF flag changes from 1 to 0 while this bit is 1, an idle interrupt request is generated.

RDRIS Bit (Receive Data Ready Interrupt Select)

This bit is used to select the SPRI interrupt request or SPEI interrupt request to be generated when the receive data ready is detected (SPSR.RRDYF = 1).

SPTIE Bit (Transmit Buffer Empty Interrupt Enable)

This bit is used to enable or disable a transmit buffer empty interrupt request.

SPCIE Bit (Communication End Interrupt Enable)

This bit controls generation of a communication end interrupt request.

SPMS Bit (RSPI Mode Select)

This bit is used to select SPI operation (4-wire method) or clock synchronous operation (3-wire method).
For clock synchronous operation, the SSL0n pin is not used but three pins RSPCK0, MOSI0, and MISO0 are used for communication. When SPMS = 1 (clock synchronous operation (3-wire)), the setting of the FRFS bit is invalid.
To perform clock synchronous operation in master mode (SPCR.MSTR = 1), the SPCMDm.CPHA bit can be set to 0 or 1. To perform clock synchronous operation in slave mode (SPCR.MSTR = 0), set the CPHA bit to 1. If this bit is set to 0 for clock synchronous operation in slave mode (SPCR.MSTR = 0), subsequent operation is not guaranteed.
Table 41.4 lists the communication status according to the settings of the MSTR bit, CMMD[1:0] bits, FRFS bit, and SPMS bit of the RSPI control register (SPCR) as follows.

Table 41.4 RSPI Communication Status

SPCR. MSTR	SPCR. CMMD[1:0]	SPCR. FRFS	SPCR. SPMS	Communication Status	Communi-cation Status No.
1	00b	0	0	Transmit-Receive Master mode/Motorola SPI/SPI operation (4-wire)	1-1
1	00b	1	0	Transmit-Receive Master mode/TI SSP/SPI operation (4-wire)	1-2
1	00b	—	1	Transmit-Receive Master mode/Clock synchronous operation (3-wire)	1-3
1	01b	0	0	Transmit-only Master mode/Motorola SPI/SPI operation (4-wire)	1-4
1	01b	1	0	Transmit-only Master mode/TI SSP/SPI operation (4-wire)	1-5
1	01b	—	1	Transmit-only Master mode/Clock synchronous operation (3-wire)	1-6
1	10b	0	0	Receive-only Master mode/Motorola SPI/SPI operation (4-wire)	1-7
1	10b	1	0	Receive-only Master mode/TI SSP/SPI operation (4-wire)	1-8
1	10b	—	1	Receive-only Master mode/Clock synchronous operation (3-wire)	1-9
0	00b	0	0	Transmit-Receive Slave mode/Motorola SPI/SPI operation (4-wire) (default)	0-1
0	00b	1	0	Transmit-Receive Slave mode/TI SSP/SPI operation (4-wire)	0-2
0	00b	—	1	Transmit-Receive Slave mode/Clock synchronous operation (3-wire)	0-3
0	01b	0	0	Transmit-only Slave mode/Motorola SPI/SPI operation (4-wire)	0-4
0	01b	1	0	Transmit-only Slave mode/TI SSP/SPI operation (4-wire)	0-5
0	01b	—	1	Transmit-only Slave mode/Clock synchronous operation (3-wire)	0-6
0	10b	0	0	Receive-only Slave mode/Motorola SPI/SPI operation (4-wire)	0-7
0	10b	1	0	Receive-only Slave mode/TI SSP/SPI operation (4-wire)	0-8
0	10b	—	1	Receive-only Slave mode/Clock synchronous operation (3-wire)	0-9

—: don't care

FRFS Bit (Frame Format Select)

This bit selects the communication protocol. The format of the RSPI terminal (RSPCK0, SSL0n (n = 0 to 3) can be set according to the set communication protocol.
When the SPMS bit is 1 (clock synchronous operation (3-wire method)), this bit is invalid because SSL0n pin is not used.

CMMD[1:0] Bits (Communication Mode Select)

This bit is used to select the transmit-receive, transmit-only, and receive-only serial communication.
When CMMD[1:0] is set to 01b for communication, transmit-only is performed without reception.
When CMMD[1:0] is set to 10b for communication, receive-only is performed without transmission.

When CMMD[1:0] is set to 01b for communication, a receive buffer full interrupt request cannot be used.
When CMMD[1:0] is set to 10b for communication, a transmit buffer empty interrupt request cannot be used.
Refer to section 41.3.6, Communication Mode.

MSTR Bit (RSPI Master/Slave Mode Select)

This bit is used to select master mode or slave mode of the RSPIA. The RSPIA determines input/output directions of pins RSPCK0, MOSI0, MISO0, and SSL01 to SSL03 according to the MSTR bit setting.

SYNDIS Bit (Synchronizer Disable)

Set this bit to 1 for this MCU.

41.2.6 RSPI Receive-Only Mode Control Register (SPRMCR)

Address(es): RSPIA0.SPRMCR 000E 280Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	START	TERM	—	RFC[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	RFC[4:0]	Receive Frame Count Select	The number of received frames can be adjusted in receive-only master mode. b4 b0 0 0 0 0 0: This function is not used. 0 0 0 0 1: Automatically stop communication after processing 1 received frame : 1 1 1 1 1: Automatically stop communication after processing 31 received frames Settings are not effective except in receive-only master mode.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R
b6	TERM	Reception Terminate	1: Receive End (Writable only in receive-only master mode) The read value is always 0.	W
b7	START	Reception Start	1: Receive Start (Writable only in receive-only master mode) The read value is always 0.	W

The SPRMCR register is used to control the start and completion of communication for receive-only master mode. If the RFC[4:0] bits are changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

RFC[4:0] Bits (Receive Frame Count Select)

The number of received frames can be adjusted when operating in receive-only master mode. Valid only when the master mode (SPCR.MSTR = 1) and the communication mode select bits (SPCR.CMMD[1:0]) are 10b. After reception is started by the START bit, the communication is automatically stopped after processing frames according to the value set in these bits.

TERM Bit (Reception Terminate)

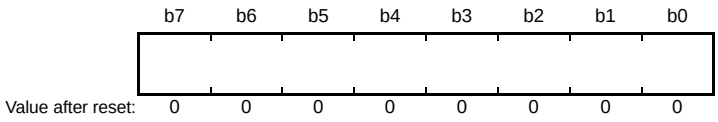
This bit is used to end reception in receive-only master mode. Valid only when the master mode (SPCR.MSTR = 1) and the communication mode select bits (SPCR.CMMD[1:0]) are 10b.

START Bit (Reception Start)

This bit is used to start reception in receive-only master mode. Valid only when the master mode (SPCR.MSTR = 1) and the communication mode select bits (SPCR.CMMD[1:0]) are 10b.
Writing 1 to this bit during reception is not accepted. Write again after reception is completed.

41.2.7 RSPI Receive Data Ready Detect Condition Setting Register (SPDRCSR)

Address(es): RSPIA0.SPDRCSR 000E 280Dh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	—	0h: Disables receive data ready detection function 1h: Performs receive data ready detection after 1 PCLKA : FFh: Performs receive data ready detection after 255 PCLKA	R/W

The SPDRCSR register is used to set the RSPI receive data ready detection function. If the setting value is changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

The receive data ready detection function can be disabled or, if used, the period until detection can be set from 1 to 255 PCLKA. The value set in the SPDRCSR register is used to 1 set the RRDYF flag. For details, see the description of the RRDYF flag in section 41.2.16, RSPI Status Register (SPSR).

41.2.8 RSPI Pin Control Register (SPPCR)

Address(es): RSPIA0.SPPCR 000E 280Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	MOIFE	MOIFV	—	—	SPLP2	SPLP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPLP	RSPI Loopback	0: Normal mode 1: Loopback mode (data is inverted for transmission)	R/W
b1	SPLP2	RSPI Loopback 2	0: Normal mode 1: Loopback mode (data is not inverted for transmission)	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b4	MOIFV	MOSI Idle Fixed Value	0: The fixed value during MOSI idling corresponds to low 1: The fixed value during MOSI idling corresponds to high	R/W
b5	MOIFE	MOSI Idle Value Fixing Enable	0: MOSI output value equals final data from previous transfer 1: MOSI output value equals the value set in the MOIFV bit	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SPPCR register is used to set pin mode of the RSPIA. If this register is changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

SPLP Bit (RSPI Loopback)

When the SPLP bit is set to 1, the RSPIA shuts off the path between the MISO0 pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSI0 pin and the shift register if the SPCR.MSTR bit is 0, and connects (inverts) the input path and output path for the shift register (loopback mode).

SPLP2 Bit (RSPI Loopback 2)

When the SPLP2 bit is set to 1, the RSPIA shuts off the path between the MISO0 pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSI0 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).
If this bit is set to 1 together with the SPLP bit, setting this bit takes precedence.

MOIFV Bit (MOSI Idle Fixed Value)

If the MOIFE bit is 1 in master mode, the MOIFV bit determines the MOSI0 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 MOSI0 output value when the RSPIA 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 RSPIA outputs the last data from the previous serial transfer during the SSL negation period to the MOSI0 pin. When the MOIFE bit is 1, the RSPIA outputs the fixed value set in the MOIFV bit to the MOSI0 pin.

41.2.9 RSPI Slave Select Polarity Register (SSLP)

Address(es): RSPIA0.SSLP 000E 2810h

	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	SSL00 Signal Polarity Setting	[Motorola SPI] 0: The SSL00 signal is active low. 1: The SSL00 signal is active high. [TI SSP] 0: The SSL00 signal is active high. 1: The SSL00 signal is active low.	R/W
b1	SSL1P	SSL01 Signal Polarity Setting	[Motorola SPI] 0: The SSL01 signal is active low. 1: The SSL01 signal is active high. [TI SSP] 0: The SSL01 signal is active high. 1: The SSL01 signal is active low.	R/W
b2	SSL2P	SSL02 Signal Polarity Setting	[Motorola SPI] 0: The SSL02 signal is active low. 1: The SSL02 signal is active high. [TI SSP] 0: The SSL02 signal is active high. 1: The SSL02 signal is active low.	R/W
b3	SSL3P	SSL03 Signal Polarity Setting	[Motorola SPI] 0: The SSL03 signal is active low. 1: The SSL03 signal is active high. [TI SSP] 0: The SSL03 signal is active high. 1: The SSL03 signal is active low.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SSLP register is used to set the polarity of SSL0n signals (n = 0 to 3) of the RSPIA. If this register is changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

SSLnP Bits (SSL0n Signal Polarity Setting) (n = 0 to 3)

These bits are used to specify the polarity of SSL0n signals. The values of the SSLnP bit indicate the active polarity of SSL0n signals.

Note: SSL00 is different from SSL01 to SSL03. When slave or multi-master, it functions as an input.
For details, refer to section 41.3.3.2, Single Master/Single Slave (with This MCU Acting as Slave), and section 41.3.3.5, Multi-Master/Multi-Slave (with This MCU Acting as Master).

41.2.10 RSPI Bit Rate Register (SPBR)

Address(es): RSPIA0.SPBR 000E 2811h

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

The SPBR register is used to set the bit rate in master mode. If the value of the SPBR register is changed while both the SPCR.MSTR and SPCR.SPE bits are 1, the operation after the change is not guaranteed.
The bit rate is determined by combinations of the SPBR register setting and the SPCMDm.BRDV[1:0] bit setting.
When the RSPIA is used in slave mode, the bit rate depends on the frequency of the input clock (bit rate satisfying the electrical characteristics should be used) regardless of the settings of the SPBR register and the SPCMDm.BRDV[1:0] bits.
The equation for calculating the bit rate is given below. In the equation, n denotes the SPBR register setting (0, 1, 2, …, 255), and N denotes the BRDV[1:0] bit setting (0, 1, 2, 3).

$$\text{Bit rate} = \frac{f(PCLKA)}{2 \times (n + 1) \times 2^{N}}$$

Table 41.5 lists examples of the relationship among the SPBR register settings, the BRDV[1:0] bit settings, and bit rates.

Table 41.5 Relationship among SPBR Register Settings, BRDV[1:0] Bit Settings, and Bit Rates

SPBR (n)	BRDV[1:0] Bits (N)	Division Ratio	Bit Rate				
			PCLKA = 32 MHz	PCLKA = 36 MHz	PCLKA = 40 MHz	PCLKA = 50 MHz	PCLKA = 120 MHz
0	0	2	16.0 Mbps	18.0 Mbps	20.0 Mbps	25.0 Mbps	60.0 Mbps
1	0	4	8.00 Mbps	9.00 Mbps	10.0 Mbps	12.5 Mbps	30.0 Mbps
2	0	6	5.33 Mbps	6.00 Mbps	6.67 Mbps	8.33 Mbps	20.0 Mbps
3	0	8	4.00 Mbps	4.50 Mbps	5.00 Mbps	6.25 Mbps	15.0 Mbps
4	0	10	3.20 Mbps	3.60 Mbps	4.00 Mbps	5.00 Mbps	12.0 Mbps
5	0	12	2.67 Mbps	3.00 Mbps	3.33 Mbps	4.16 Mbps	10.0 Mbps
5	1	24	1.33 Mbps	1.50 Mbps	1.67 Mbps	2.08 Mbps	5.00 Mbps
5	2	48	677 kbps	750 kbps	833 kbps	1.04 Mbps	2.50 Mbps
5	3	96	333 kbps	375 kbps	417 kbps	521 kbps	1.25 Mbps
255	3	4096	7.81 kbps	8.80 kbps	9.78 kbps	12.2 kbps	29.3 kbps

41.2.11 RSPI Sequence Control Register (SPSCR)

Address(es): RSPIA0.SPSCR 000E 2813h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SPSLN[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPSLN[2:0]	RSPI Sequence Length Setting	b2 b0 Sequence Length — Referenced SPCMD0 to SPCMD7 (No.) 0 0 0: 1 — 0→0→… 0 0 1: 2 — 0→1→0→… 0 1 0: 3 — 0→1→2→0→… 0 1 1: 4 — 0→1→2→3→0→… 1 0 0: 5 — 0→1→2→3→4→0→… 1 0 1: 6 — 0→1→2→3→4→5→0→… 1 1 0: 7 — 0→1→2→3→4→5→6→0→… 1 1 1: 8 — 0→1→2→3→4→5→6→7→0→… The order in which the SPCMD0 to SPCMD7 registers are to be referenced is changed according to the sequence length that is set in these bits. The relationship among the setting of these bits, sequence length, and SPCMD0 to SPCMD7 registers referenced by the RSPIA is shown above. However, the RSPIA in slave mode references SPCMD0.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SPSCR register is used to set the sequence length when the RSPIA 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 Setting)

The SPSLN[2:0] bits specify a sequence length when the RSPIA in master mode performs sequential operations. The RSPIA 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. For details, refer to (3) Sequence Control in section 41.3.13.1, Master Mode Operation.
In slave mode, SPCMD0 is referred.

41.2.12 RSPI Command Register m (SPCMDm) (m = 0 to 7)

Address(es): RSPIA0.SPCMD0 000E 2814h, RSPIA0.SPCMD1 000E 2818h, RSPIA0.SPCMD2 000E 281Ch, RSPIA0.SPCMD3 000E 2820h, RSPIA0.SPCMD4 000E 2824h, RSPIA0.SPCMD5 000E 2828h, RSPIA0.SPCMD6 000E 282Ch, RSPIA0.SPCMD7 000E 2830h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	SSLA[2:0]			—	—	—	SPB[4:0]				
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
	SCKDEN	SLNDEN	SPNDEN	LSBF	—	—	—	—	SSLKP	—	—	—	BRDV[1:0]		CPOL	CPHA
Value after 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	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	—	Reserved	These bits are read as 0. The write value should be 0.	R
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	—	Reserved	These bits are read as 0. The write value should be 0.	R
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 + 5 PCLKA 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: In master mode, an SSL negation delay of 1 RSPCK In slave and TI SSP mode, an SSL negation delay of 1 PCLKA 1: An SSL negation delay is equal to the setting of the RSPI slave select negation delay register (SSLND)	R/W
b15	SCKDEN	RSPCK Delay Setting Enable	0: In Motorola SPI mode, an RSPCK delay of 1 RSPCK In TI SSP mode, an RSPCK delay of 0 RSPCK 1: An RSPCK delay is equal to the setting of the RSPI clock delay register (SPCKD)	R/W
b20 to b16	SPB[4:0]	RSPI Data Length Setting	b20 b16 00000 to 00010: Setting prohibited 0 0 0 1 1: 4 bits 0 0 1 0 0: 5 bits 0 0 1 0 1: 6 bits : : 1 1 1 1 0: 31 bits 1 1 1 1 1: 32 bits	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R
b26 to b24	SSLA[2:0]	SSL Signal Assertion Setting	b26 b24 0 0 0: SSL00 0 0 1: SSL01 0 1 0: SSL02 0 1 1: SSL03 100 to 111: Setting prohibited	R/W
b31 to b27	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SPCMDm register is used to set a transfer format for the RSPIA 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 RSPIA in slave mode. The RSPIA 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.
The 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 RSPIA in master mode can be checked by means of the SPSSR.SPCP[2:0] bits. If this register is changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

CPHA Bit (RSPCK Phase Setting)

The CPHA bit is used to set an RSPCK phase of the RSPIA in master mode or slave mode. Data communications between RSPIA modules require the same RSPCK phase setting between the modules.
When the SPCR.SPMS is 0 and the SPCR.FRFS bit is 1 in TI SSP mode, setting the CPHA bit to 0 is not effective.

CPOL Bit (RSPCK Polarity Setting)

The CPOL bit is used to set an RSPCK polarity of the RSPIA in master mode or slave mode. Data communications between RSPIA 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 the SPBR register (refer to section 41.2.10, RSPI Bit Rate Register (SPBR)). The settings in the SPBR register 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 the SPCMDm register, different BRDV[1:0] bit settings can be specified. This enables execution of serial transfers at a different bit rate for each command.

SSLKP Bit (SSL Signal Level Keeping)

When the RSPIA in master mode performs a serial transfer, the SSLKP bit specifies whether the SSL0n (n = 0 to 3) signal level for the current command is to be kept or negated between the SSL0n negation timing associated with the current command and the SSL0n assertion timing associated with the next command.
Setting the SSLKP bit to 1 enables a burst transfer. For details, refer to section 41.3.13.1, Master Mode Operation (4) Burst Transfer.
When using the RSPIA in slave mode, the SSLKP bit should be set to 0.

LSBF Bit (RSPI LSB First)

The LSBF bit is used to set the data format of the RSPIA in master mode or slave mode to MSB first or LSB first.

SPNDEN Bit (RSPI Next-Access Delay Enable)

The SPNDEN bit is used to set the period from the time the RSPIA in master mode terminates a serial transfer and sets the SSL0n signal inactive until the RSPIA enables the SSL0n signal assertion for the next access (next-access delay). If the SPNDEN bit is 0, the RSPIA sets the next-access delay to 1 RSPCK + 5 PCLKA. If the SPNDEN bit is 1, the RSPIA inserts a next-access delay in compliance with the SPND setting.
When using the RSPIA in slave mode, the SPNDEN bit should be set to 0.

SLNDEN Bit (SSL Negation Delay Setting Enable)

[Motorola SPI]
The SLNDEN bit is used to set the period from the time the RSPIA in master mode stops RSPCK oscillation until the RSPIA sets the SSL signal inactive (SSL negation delay). If the SLNDEN bit is 0, the RSPIA sets the SSL negation delay to 1 RSPCK. If the SLNDEN bit is 1, the RSPIA negates the SSL signal at an SSL negation delay in compliance

with the SSLND setting.
When using the RSPIA in slave mode, set the SLNDL[2:0] bits to 000b.

[TI SSP]
The SLNDEN bit is used to set the period from the time the RSPIA in master mode stops RSPCK oscillation until the RSPIA sets the OE signal inactive, or from the detection of a final RSPCK edge to the negation of the OE signal by the RSPIA in slave mode. If the SLNDEN bit is 0, the SSL negation delay is 1 RSPCK in master mode and 1 PCLKA in slave mode. If the SLNDEN bit is 1, the RSPIA negates the SSL signal at an SSL negation delay in compliance with the SSLND setting.

[Clock synchronous]
When using the RSPIA in slave mode, set this bit to 0.

SCKDEN Bit (RSPCK Delay Setting Enable)

[Motorola SPI]
The SCKDEN bit is used to set the period from the point when the RSPIA in master mode activates the SSL signal until the RSPCK starts oscillation (RSPCK delay). If the SCKDEN bit is 0, the RSPIA sets the RSPCK delay to 1 RSPCK. If the SCKDEN bit is 1, the RSPIA starts the oscillation of RSPCK at an RSPCK delay in compliance with the SPCKD setting.

[TI SSP]
The SCKDEN bit is used to set the period from the point when the RSPIA in master mode activates the SSL signal until the RSPCK starts oscillation (RSPCK delay) and to the negation of the SSL0n signal. If the SCKDEN bit is 0, the RSPIA does not set the RSPCK delay. If the SCKDEN bit is 1, the RSPIA starts the oscillation of RSPCK at an RSPCK delay in compliance with the SPCKD setting.
When using the RSPIA in slave mode, the SCKDEN bit should be set to 0.

SPB[4:0] Bits (RSPI Data Length Setting)

The SPB[4:0] bits are used to set a transmit data length for the RSPIA in master mode or slave mode.

SSLA[2:0] Bits (SSL Signal Assertion Setting)

The SSLA[2:0] bits control the SSL0n signal assertion when the RSPIA performs serial transfers in master mode. Setting the SSLA[2:0] bits controls the assertion for the SSL0n signal. When an SSL0n (n = 0 to 3) signal is asserted, its polarity is determined by the set value in the corresponding SSLP. When the SSLA[2:0] bits are set to 000b in multi-master mode, serial transfers are performed with all the SSL signals in the negated state (as the SSL00 pin acts as input).
When using the RSPIA in slave mode, set the SSLA[2:0] bits to 000b.

41.2.13 RSPI Data Control Register (SPDCR)

Address(es): RSPIA0.SPDCR 000E 2840h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	SPFC[1:0]		—	—	—	DINV	SPRDTD	—	—	BYSW
Value after 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	BYSW	Byte Swap	0: Byte swapping of SPDR data disabled 1: Byte swapping of SPDR data enabled	R/W
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R
b3	SPRDTD	RSPI Receive/Transmit Data Select	0: SPDR values are read from the receive buffer 1: SPDR values are read from the transmit buffer	R/W
b4	DINV	Transfer Data Invert	0: Not invert serial data 1: Invert serial data	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R
b9, b8	SPFC[1:0]	Number of Frames Setting	b9 b8 0 0: 1 frame 0 1: 2 frames 1 0: 3 frames 1 1: 4 frames	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SPDCR register is used to control the data format of the RSPIA. If this register is changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

BYSW Bit (Byte Swap)

The BYSW bit is used to swap a transmit/receive data in byte units. A data alignment after byte swap is different by a data length (setting of SPCMDm.SPB[4:0]).

When using byte swap, set the data length (setting of SPCMDm.SPB[4:0]) to 32 bits or 16 bits. Other case of data length (i.e. 4 to 15, 17 to 31-bit length), byte swap is not guaranteed. For the arrangement of data before and after swapping data lengths of 32 bits and 16 bits, refer to section 41.3.4.4, Byte Swap Transmission and section 41.3.4.5, Byte Swap Reception.

If this bit is changed while the SPCR.SPPE bit is 1, the operation is not guaranteed.

SPRDTD Bit (RSPI Receive/Transmit Data Select)

The SPRDTD bit is used to whether the SPDR register reads values from the receive buffer or from the transmit buffer. If reading is from the transmit buffer, the value written to the SPDR register immediately beforehand is read.

DINV Bit (Transfer Data Invert)

The DINV bit is used to invert transmit data and receive data.

When the DINV bit is set to 1, transmit buffer (SPTX) data is inverted to invert transmit data and receive data, and then the inverted data is stored in the receive buffer (SPRX). The parity bit is the value corresponding to the inverted transmission/reception data.

SPFC[1:0] Bits (Number of Frames Setting)

The SPFC[1:0] bits are used for the condition to set the SPSR.SPCF flag in slave receive-only mode.

For details on the SPCF flag setting conditions, refer to [Setting condition] of the SPCF flag in section 41.2.16.

Note that these bits are not effective except in the slave receive-only mode.

41.2.14 RSPI FIFO Control Register (SPFCR)

Address(es): RSPIA0.SPFCR 000E 2844h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TTRG[1:0]		—	—	—	—	—	—	RTRG[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	RTRG[1:0]	Receive FIFO Threshold Setting	b1 b0 0 0: When the receive FIFO fill level is 0. 0 1: When the receive FIFO fill level is 1. 1 0: When the receive FIFO fill level is 2. 1 1: When the receive FIFO fill level is 3.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R
b9, b8	TTRG[1:0]	Transmit FIFO Threshold Setting	b9 b8 0 0: When the transmit FIFO free level is 0. 0 1: When the transmit FIFO free level is 1. 1 0: When the transmit FIFO free level is 2. 1 1: When the transmit FIFO free level is 3.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SPFCR register is used to control the FIFO threshold. If this register is changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

RTRG[1:0] Bits (Receive FIFO Threshold Setting)

The RTRG[1:0] bits are used to select receive FIFO threshold. When the number of data stored in the receive FIFO is greater than the threshold value set by RTRG[1:0], the receive buffer full flag is set.

TTRG[1:0] Bits (Transmit FIFO Threshold Setting)

The TTRG[1:0] bits are used to select transmit FIFO threshold. When the number of empty stages in the transmit FIFO is greater than the threshold value set in TTRG[1:0], the transmit buffer empty flag is set.

41.2.15 RSPI Sequence Status Register (SPSSR)

Address(es): RSPIA0.SPSSR 000E 2851h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SPECM[2:0]			—	SPCP[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPCP[2:0]	RSPI Command Pointer	b2 b0 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
b3	—	Reserved	This bit is read as 0.	R
b6 to b4	SPECM[2:0]	RSPI Error Command	b6 b4 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
b7	—	Reserved	This bit is read as 0.	R

The SPSSR register indicates the sequence control status when the RSPIA operates in master mode.
Any writing to the SPSSR register is ignored.

SPCP[2:0] Bits (RSPI Command Pointer)

The SPCP[2:0] bits indicate the SPCMDm register that is currently pointed to by the pointer during sequence control by the RSPIA.
For the RSPIA's sequence control, refer to section 41.3.13.1, Master Mode Operation.

SPECM[2:0] Bits (RSPI Error Command)

The SPECM[2:0] bits indicate the SPCMDm register that is specified by the SPCP[2:0] bits when an error is detected during sequence control by the RSPIA. The RSPIA updates the SPECM[2:0] bits only when an error is detected. If the SPSR.OVRF, MODF, and PERF flags are 0 and there is no error, the values of the SPECM[2:0] bits have no meaning.
For the RSPIA's error detection function, refer to section 41.3.10, Error Detection. For the RSPIA's sequence control, refer to section 41.3.13.1, Master Mode Operation.

41.2.16 RSPI Status Register (SPSR)

Address(es): RSPIA0.SPSR 000E 2852h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SPRF	SPCF	SPTEF	UDRF	PERF	MODF	IDLNF	OVRF	RRDYF	—	—	—	—	—	—	—
Value after reset:	0	0	1	0	0	0	0	0	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.	R
b7	RRDYF	Receive Data Ready Flag	0: Receive data ready is not detected 1: Receive data ready is detected	R
b8	OVRF	Overrun Error Flag	0: No overrun error occurs 1: An overrun error occurs	R
b9	IDLNF	Idle Flag	0: RSPIA is in the idle state 1: RSPIA is in the transfer state	R
b10	MODF	Mode Fault Error Flag	0: Neither a mode fault error nor an underrun error occurs 1: A mode fault error or an underrun error occurs	R
b11	PERF	Parity Error Flag	0: No parity error occurs 1: A parity error occurs	R
b12	UDRF	Underrun Error Flag	This flag is used with the MODF flag to check the state in terms of mode fault errors and underrun errors. MODF UDRF 0 x: Neither a mode fault error nor an underrun error occurs 1 0: A mode fault error occurs 1 1: An underrun error occurs	R
b13	SPTEF	Transmit Buffer Empty Flag	0: The number of empty stages in the transmit FIFO ≤ the value set by the SPFCR.TTRG[1:0] bits 1: The number of empty stages in the transmit FIFO > the value set by the SPFCR.TTRG[1:0] bits	R
b14	SPCF	Communication End Flag	0: RSPIA is not communicating or communicating 1: RSPIA communication is completed	R
b15	SPRF	Receive Buffer Full Flag	0: The number of data stored in the receive FIFO ≤ number of frames set by the SPFCR.RTRG[1:0] bits 1: The number of data stored in the receive FIFO > number of frames set by the SPFCR.RTRG[1:0] bits	R

x: don't care

The SPSR register is used to store flags that indicate RSPIA's operating status.

RRDYF Flag (Receive Data Ready Flag)

The RRDYF flag indicates a certain period of time has elapsed while the number of data stored in the receive FIFO during communication (SPCR.SPE = 1) is equal to or less than the receive FIFO threshold value.
This flag is set to 0 when the receive operation is not performed (SPCR.CMMD[1:0] = 01b).

[Setting condition]

When all the following conditions are satisfied.

- SPDRCSR register ≠ 00h.
- After the receive FIFO has been written, when the number of data stored in the receive FIFO is equal to or less than the receive FIFO threshold value and the value set by the SPDRCSR register has elapsed

[Clearing condition]

When 1 is written to the SPSCLR.RRDYFC bit.

OVRF Flag (Overrun Error Flag)

The OVRF flag indicates the occurrence of an overrun error. In master mode (when the SPCR.MSTR bit is 1) and when the RSPCK auto-stop function is enabled (the SPCR.SCKASE bit is 1), an overrun error does not occur; accordingly this flag does not become 1. For details, refer to section 41.3.10.1, Overrun Error.

[Setting condition]

When serial transfer is completed with no free space in the receive FIFO under any of the following conditions.

- The SPCR.CMMD[1:0] bits are 00b (transmit-receive master mode or transmit-receive slave mode).
- The SPCR.CMMD[1:0] bits are 10b (receive-only master mode or receive-only slave mode).

[Clearing condition]

When 1 is written to the SPSCLR.OVRFC bit

IDLNF Flag (Idle Flag)

The IDLNF flag indicates the transfer status of the RSPIA.

[Setting condition]

(1) Transmit-receive master mode or transmit-only master mode

- When none of the conditions are satisfied in transmit-receive master mode or transmit-only master mode under the [Clearing condition] below.

(2) Transmit-only master mode

- When 1 is written to the SPRMCR.START bit.

(3) Slave mode

- The SPCR.SPE bit is 1 (enables the RSPI function)

[Clearing condition]

(1) Transmit-receive master mode or transmit-only master mode

Communication status: No. 1-1 to 1-6 (for details, refer to Table 41.4 RSPI Communication Status)

- When the SPCR.SPE bit is set to 0 (disables the RSPI function)
- When all of the following conditions are satisfied.
  (a) The the next transmit data is not set in the transmit buffer (SPTX)
  (b) The SPSSR.SPCP[2:0] bits are 000b (beginning of sequence control)
  (c) The operations up to the next-access delay have finished (the master main state machine has entered the idle state)

(2) Transmit-only master mode

Communication status: No. 1-7 to 1-9 (for details, refer to Table 41.4 RSPI Communication Status)

- When the SPCR.SPE bit is set to 0 (disables the RSPI function)
- When any of the following conditions is satisfied.
  (a) When the SPRMCR.RFC[4:0] bits are 00000b, after 1 is written to the SPRMCR.TERM bit, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)
  (b) When the SPRMCR.RFC[4:0] bits are other than 00000b, after 1 is written to the SPRMCR.TERM bit, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)
  (c) When the SPRMCR.RFC[4:0] bits are other than 00000b, after the operation is completed for the number of received frames set by the SPRMCR.RFC[4:0] bits, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)

(3) Slave mode

Communication status: No. 0-1 to 0-9 (for details, refer to Table 41.4 RSPI Communication Status)

- When the SPCR.SPE bit is set to 0 (disables the RSPI function)

MODF Flag (Mode Fault Error Flag)

Indicates the occurrence of a mode fault error or an underrun error. The UDRF flag indicates whether the error is a mode fault error or an underrun error.

[Setting condition]

(1) Multi-master mode

- When the input level of the SSL0n 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 RSPIA detects a mode fault error

(2) Slave mode (SPI operation, Motorola SPI)

When any of the following conditions are satisfied.

- When the SSL00 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 RSPIA detects a mode fault error
- When the serial transfer starts while the SPCR.SPE bit is 1 (enables the RSPI function) and the transmit data are not ready for output, the RSPIA detects an underrun error

(3) Slave mode (SPI operation, TI SSP)

When any of the following conditions are satisfied.

- When the SSL00 pin is asserted 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 RSPIA detects a mode fault error
- When the serial transfer starts while the SPCR.SPE bit is 1 (enables the RSPI function) and the transmit data are not ready for output, the RSPIA detects an underrun error

The active level of the SSL0n signal is determined by the SSLP.SSLnP bit (SSL signal polarity setting bit).

[Clearing condition]

When 1 is written to the SPSCLR.MODFC bit

PERF Flag (Parity Error Flag)

Indicates the occurrence of a parity error.

[Setting condition]

When the serial transfer ends and a parity error is detected while the SPCR.SPPE bit is 1 under any of the following conditions.

- The SPCR.CMMD[1:0] bits are 00b (transmit-receive master mode or transmit-receive slave mode)
- The SPCR.CMMD[1:0] bits are 10b (receive-only master mode or receive-only slave mode)

[Clearing condition]

When 1 is written to the SPSCLR.PERFC bit

UDRF Flag (Underrun Error Flag)

When a mode fault error or an underrun error has occurred, the UDRF flag indicates whether the error is a mode fault error or an underrun error.

[Setting condition]

When the serial transfer starts while the SPCR.MSTR bit is 0, the SPCR.CMMD[1:0] bits are 00b or 01b (transmit-receive slave mode or transmit-only slave mode), the SPCR.SPE bit is 1 (enables the RSPI function) and the transmit data are not ready for output, the RSPIA detects an underrun error

[Clearing condition]

When 1 is written to the SPSCLR.UDRFC bit

SPTEF Flag (Transmit Buffer Empty Flag)

Indicates whether the transmit buffer (SPTXn, n = 0 to 3) in the RSPI data register (SPDR) has valid data.

[Setting condition]

When any of the following conditions are satisfied.

- When the SPCR.SPE bit is 0 (disables the RSPI function)
- When the number of empty transmission FIFO stages is greater than the threshold value set in SPFCR.TTRG[1:0]
- When 1 is written to the SPFCLR.FCLR bit

[Clearing condition]

When any of the following conditions are satisfied.

- At the final access when transmit data is written to SPDR (SPTXn) in one processing routine using DTC/DMAC
- When 1 is written to the SPSCLR.SPTEFC bit

The SPDR register can be set only when the SPSR.SPTEF flag is 1. The data in the transmit buffer (SPTXn) is not updated when the SPDR register is set while the SPSR.SPTEF flag is 0.

SPCF Flag (Communication End Flag)

Indicates communication end status of RSPI. It turns 1 at communication end and turns 0 at starting next communication.

[Setting condition]

(1) Transmit-receive master mode or transmit-only master mode

Communication status: No. 1-1 to 1-6 (for details, refer to Table 41.4 RSPI Communication Status)

When all of the following conditions are satisfied.

- The next transmit data is not set in the transmission buffer (SPTXn)
- The SPSSR.SPCP[2:0] bits are 000b (beginning of sequence control)
- The operations up to the next-access delay have finished (the master main state machine has entered the idle state)

(2) Transmit-only master mode

Communication status: No. 1-7 to 1-9 (for details, refer to Table 41.4 RSPI Communication Status)

When any of the following conditions are satisfied.

- When the SPRMCR.RFC[4:0] bits are 00000b, after 1 is written to the SPRMCR.TERM bit, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)
- When the SPRMCR.RFC[4:0] bits are other than 00000b, after 1 is written to the SPRMCR.TERM bit, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)

- When the SPRMCR.RFC[4:0] bits are other than 00000b, after the operation is completed for the number of received frames set by the SPRMCR.RFC[4:0] bits, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)

(3) Transmit-receive slave mode or transmit-only slave mode (SPI operation, Motorola SPI)
Communication status: No. 0-1, 0-4 (for details, refer to Table 41.4 RSPI Communication Status)
When all of the following conditions are satisfied.

- The next transmit data is not set in the transmission buffer (SPTXn)
- The transmission shift register is empty (RSPIA does not perform serial transfer)
- The SSL00 input is negated

(4) Transmit-receive slave mode or transmit-only slave mode (SPI operation, TI SSP)
Communication status: No. 0-2, 0-5 (for details, refer to Table 41.4 RSPI Communication Status)
When all of the following conditions are satisfied.

- The next transmit data is not set in the transmission buffer (SPTXn)
- The transmission shift register is empty (RSPIA does not perform serial transfer)
- When the SSL negation delay is completed

(5) Transmit-receive slave mode or transmit-only slave mode (clock synchronous operation)
Communication status: No. 0-3, 0-6 (for details, refer to Table 41.4 RSPI Communication Status)
When all of the following conditions are satisfied.

- The next transmit data is not set in the transmission buffer (SPTXn)
- The transmission shift register is empty (RSPIA does not perform serial transfer)
- The last even edge of RSPCK of the frame was detected (SPCMDm.CPHA = 1)

(6) Receive-only slave mode (SPI operation, Motorola SPI)
Communication status: No. 0-7 (for details, refer to Table 41.4 RSPI Communication Status)

- SSL00 input was negated after getting frames for SPDCR.SPFC[1:0] set value in the receive buffer (SPRX)

(7) Receive-only slave mode (SPI operation, TI SSP)
Communication status: No. 0-8 (for details, refer to Table 41.4 RSPI Communication Status)

- SSL negation delay is completed after getting frames for SPDCR.SPFC[1:0] set value in the receive buffer (SPRX).

(8) Receive-only slave mode (clock synchronous operation)
Communication status: No. 0-9 (for details, refer to Table 41.4 RSPI Communication Status)

- The last even edge of RSPCK of the last frame received for SPFC[1:0] set value was detected (SPCMDm.CPHA = 1)

[Clearing condition]
(1) Transmit-receive master mode or transmit-only master mode
Communication status: No. 1-1 to 1-6 (for details, refer to Table 41.4 RSPI Communication Status)
When any of the following conditions are satisfied.

- The next transmit data is set in the transmission buffer (SPTXn)
- When 1 is written to the SPSCLR.SPCFC bit

(2) Transmit-only master mode

Communication status: No. 1-7 to 1-9 (for details, refer to Table 41.4 RSPI Communication Status)
When any of the following conditions are satisfied.

- When 1 is written to the SPRMCR.START bit while the SPCR.SPE bit is 1
- When 1 is written to the SPSCLR.SPCFC bit

(3) Transmit-receive slave mode or transmit-only slave mode

Communication status: No. 0-1 to 0-6 (for details, refer to Table 41.4 RSPI Communication Status)
When any of the following conditions are satisfied.

- The next transmit data is set in the transmission buffer (SPTXn)
- When 1 is written to the SPSCLR.SPCFC bit

(4) Receive-only slave mode (SPI operation)

Communication status: No. 0-7, 0-8 (for details, refer to Table 41.4 RSPI Communication Status)
When any of the following conditions are satisfied.

- The SSL00 input assertion of next data is detected
- When 1 is written to the SPSCLR.SPCFC bit

(5) Receive-only slave mode (clock synchronous operation)

Communication status: No. 0-9 (for details, refer to Table 41.4 RSPI Communication Status)
When any of the following conditions are satisfied.

- The first edge of RSPCK of the next data was detected
- When 1 is written to the SPSCLR.SPCFC bit

SPRF Flag (Receive Buffer Full Flag)

Indicates whether the receive buffer (SPRX) in the RSPI data register has valid data.

[Setting condition]

When the number of data stored in the receive FIFO is greater than the number of frames set in the SPFCR.RTRG[1:0] bits in transmit-receive mode or receive-only mode.
Note that the SPRF flag does not become 1 when the OVRF flag is 1 (Refer to section 41.3.10, Error Detection, for details).

[Clearing condition]

When any of the following conditions are satisfied.

- At the final access when receive data is read from SPDR register (SPRXn, n = 0 to 3) in one processing routine using DTC/DMAC
- When 1 is written to the SPSCLR.SPRFC bit
- When 1 is written to the SPFCLR.FCLR bit

41.2.17 RSPI Transmit FIFO Status Register (SPTFSR)

Address(es): RSPIA0.SPTFSR 000E 2858h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	FREE[2:0]		
Value after reset:	0	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	FREE[2:0]	Transmit FIFO Free Level	b2 b0 0 0 0: Number of empty stages 0 0 0 1: Number of empty stages 1 0 1 0: Number of empty stages 2 0 1 1: Number of empty stages 3 1 0 0: Number of empty stages 4	R
b7 to b3	—	Reserved	These bits are read as 0.	R

FREE[2:0] Bits (Transmit FIFO Free Level)

Indicate the number of empty transmit FIFO stages. By setting the SPCR.SPE bit to 0, the FREE[2:0] bits will be 100b.

41.2.18 RSPI Receive FIFO Status Register (SPRFSR)

Address(es): RSPIA0.SPRFSR 000E 285Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	FILL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	FILL[2:0]	Receive FIFO Fill Level	b2 b0 0 0 0: Number of stored stages 0 0 0 1: Number of stored stages 1 0 1 0: Number of stored stages 2 0 1 1: Number of stored stages 3 1 0 0: Number of stored stages 4	R
b7 to b3	—	Reserved	These bits are read as 0.	R

FILL[2:0] Bits (Receive FIFO Fill Level)

Indicate the number of stored receive FIFO stages. By setting the SPCR.SPE bit to 0, the FILL[2:0] bits will be 000b.

41.2.19 RSPI Status Clear Register (SPSCLR)

Address(es): RSPIA0.SPSCLR 000E 286Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SPRFC	SPCFC	SPTEFC	UDRFC	PERFC	MODFC	—	OVRFC	RRDYFC	—	—	—	—	—	—	—
Value after 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	—	Reserved	These bits are read as 0. The write value should be 0.	R
b7	RRDYFC	RSPI Receive Data Ready Flag Clear	By writing 1, the RSPI receive data ready flag can be cleared. The read value is always 0.	W
b8	OVRFC	Overrun Error Flag Clear	By writing 1, the overrun error flag can be cleared. The read value is always 0.	W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R
b10	MODFC	Mode Fault Error Flag Clear	By writing 1, the mode fault error flag can be cleared. The read value is always 0.	W*1
b11	PERFC	Parity Error Flag Clear	By writing 1, the parity error flag can be cleared. The read value is always 0.	W
b12	UDRFC	Underrun Error Flag Clear	By writing 1, the underrun error flag can be cleared. The read value is always 0.	W*1, *2
b13	SPTEFC	Transmit Buffer Empty Flag Clear	By writing 1, the transmit buffer empty flag can be cleared. The read value is always 0.	W
b14	SPCFC	Communication End Flag Clear	By writing 1, the communication end flag can be cleared. The read value is always 0.	W
b15	SPRFC	Receive Buffer Full Flag Clear	By writing 1, the receive buffer full flag can be cleared. The read value is always 0.	W

Note 1. Before setting the MODFC and UDRFC, make sure that the SPSR.MODF and UDRF flags are set to 1.
Note 2. When the UDRF flag is cleared, the MODF flag is cleared at the same time (MODFC = 1).

The SPSCLR register is used to clear the status flags in the SPSR register that indicates the operating status of RSPIA.

41.2.20 RSPI FIFO Clear Register (SPFCLR)

Address(es): RSPIA0.SPFCLR 000E 286Ch

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

Bit	Symbol	Bit Name	Description	R/W
b0	FCLR	FIFO Clear	By writing 1, the pointer in the FIFO and the stored data are initialized. The read value is always 0.	W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

The SPFCLR register is used to clear the FIFO. If the value of the SPFCLR register is changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

FCLR Bit (FIFO Clear)

Writing 1 to this bit initializes the pointer and stored data in the transmit/receive FIFO.

41.3 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.

41.3.1 Overview of RSPI Operations

The RSPIA is capable of serial transfers in the following five modes.

(1) Slave mode (SPI operation)
(2) Single-master mode (SPI operation)
(3) Multi-master mode (SPI operation)
(4) Slave mode (clock synchronous operation)
(5) Master mode (clock synchronous operation).

A particular mode of the RSPIA can be selected by using the MSTR, MODFEN, SPMS, and FRFS bits in SPCR. Table 41.6 lists the relationship between RSPI modes and SPCR settings, and a description of each mode.

Table 41.6 Relationship between RSPI Modes and SPCR Settings and Description of Each Mode

	SPI Operation			Clock Synchronous Operation	
Mode	**Slave**	**Single-Master**	**Multi-Master**	**Slave**	**Master**
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
FRFS bit setting	Enabled	Enabled	Enabled	Disabled	Disabled
RSPCK0 signal	Input	Output	Output/Hi-Z*1	Input	Output
MOSI0 signal	Input	Output	Output/Hi-Z*1	Input	Output
MISO0 signal	Output/Hi-Z*2	Input	Input	Output	Input
SSL00 signal	Input	Output	Input	Hi-Z (not used)	Hi-Z (not used)
SSL01 to SSL03 signals	Hi-Z (not used)	Output	Output/Hi-Z*1	Hi-Z (not used)	Hi-Z (not used)
Output pin mode	CMOS/open-drain				
SSL polarity change function	Supported	Supported	Supported	—	—
Transfer rate	Up to PCLKA/2				
Clock source	RSPCK input	On-chip baud rate generator	On-chip baud rate generator	RSPCK input	On-chip baud rate generator
Clock polarity	Two				
Clock phase	Two*3	Two*3	Two*3	One (CPHA = 1)	Two
First transfer bit	MSB/LSB				
Transmit data length	4 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*4	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 when a transmit buffer empty interrupt request is generated or the SPTEF flag is 1	Transmit buffer is written to when a transmit buffer empty interrupt request is generated or the SPTEF flag is 1	RSPCK oscillation	Transmit buffer is written to when a transmit buffer empty interrupt request is generated or the SPTEF flag is 1
Sequence control	Not supported	Supported	Supported	Not supported	Supported
Transmit buffer empty detection	Supported*5	Supported	Supported	Supported*5	Supported
Receive buffer full detection	Supported*6				
Overrun error detection	Supported*6	Supported*6, *7	Supported*6, *7	Supported*6	Supported*6, *7
Parity error detection	Supported*6, *8				
Mode fault error detection	Supported (MODFEN = 1)	Not supported	Supported	Not supported	Not supported
Underrun error detection	Supported*5	Not supported	Not supported	Supported*5	Not supported

Note 1. Motorola SPI: When SSL00 is asserted by another master device, the pin becomes Hi-Z.
TI SSP: When the SPCR.SPE bit is 1, the pin becomes Hi-Z from the time SSL00 is asserted by another master device until the end of communication.

Note 2. Motorola SPI: When SSL00 is negated or the SPCR.SPE bit is 0, the pin becomes Hi-Z.
TI SSP: The pin becomes Hi-Z when the RSPIA is not communicating (from the end of communication to the next assertion of SSL00) or when the SPCR.SPE bit is 0.

Note 3. In TI SSP mode, setting the SPCMDm.CPHA bit to 0 is not effective.

Note 4. Available only in TI SSP mode.

Note 5. When RSPIA is receive-only slave mode (refer to Table 41.4 RSPI Communication Status), transmit buffer empty error detection and underrun error detection are not performed.

Note 6. When RSPIA is transmit-only master mode or transmit-only slave mode (refer to Table 41.4 RSPI Communication Status), receiver buffer full detection, overrun error detection, and parity error detection are not performed.

Note 7. When the SPCR.SCKASE bit is 1, overrun error detection does not proceed.

Note 8. When the SPCR.SPPE bit is 0, parity error detection is not performed.

41.3.2 Controlling RSPI Pins

The RSPIA 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 41.7.

Table 41.7 MOSI Signal Value Determination during SSL Negation Period

MOIFE Bit	MOIFV Bit	MOSI Signal Value during SSL Negation Period
0	0, 1	Final data from previous transfer
1	0	Low
1	1	High

41.3.3 RSPI System Configuration Examples

This configuration example describes that the low level of SSL0n signals is the active level.
When connecting and using in a multi-slave or multi-master mode, the transfer format of the connected device should be unified to either Motorola SPI or TI SSP.

41.3.3.1 Single Master/Single Slave (with This MCU Acting as Master)

Figure 41.5 shows a single-master/single-slave RSPI system configuration example when this MCU is used as a master. In the single-master/single-slave configuration, the SSL00 to SSL03 output of this MCU (master) are not used. The SSL input of the SPI slave is fixed to the low level, and the SPI slave is maintained in a select state. 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 SSL0n output of this MCU should be connected to the SSL input of the slave device.
This MCU (master) drives the RSPCK0 and MOSI0. The SPI slave drives the MISO.

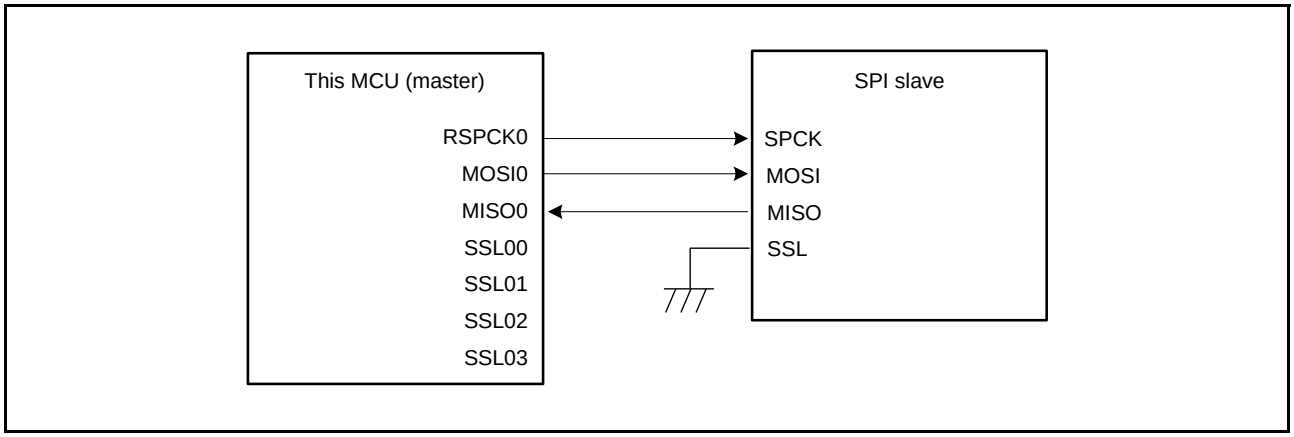


Figure 41.5 Single-Master/Single-Slave Configuration Example (This MCU = Master)

41.3.3.2 Single Master/Single Slave (with This MCU Acting as Slave)

Figure 41.6 shows a single-master/single-slave RSPI system configuration example when this MCU is used as a slave. When this MCU is to operate as a slave, the SSL00 pin is used as SSL input. The SPI master drives the SPCK and MOSI. This MCU (slave) drives the MISO0. When SSL00 is at the non-active level, the pin state is Hi-Z.
In the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, set the SPCR.FRFS bit to 0, and set the SPCR.SPMS bit to 0, the SSL00 input of this MCU (slave) is fixed to the low level, this MCU (slave) is maintained in a select state, and in this manner it is possible to execute serial transfer (Figure 41.7). Note that the communication end event and the communication end interrupt do not output when SSL00 input was fixed as Figure 41.7.

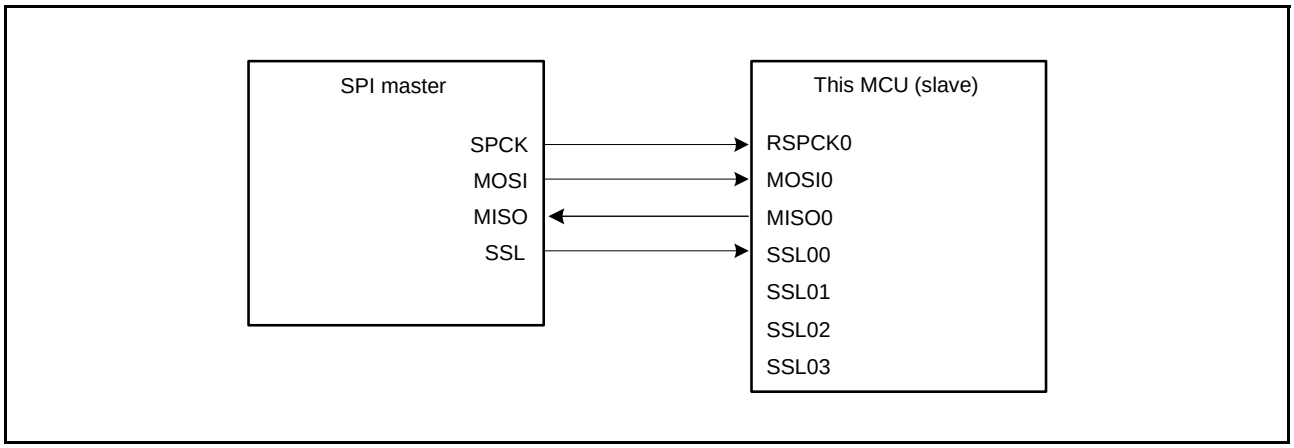


Figure 41.6 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 0)

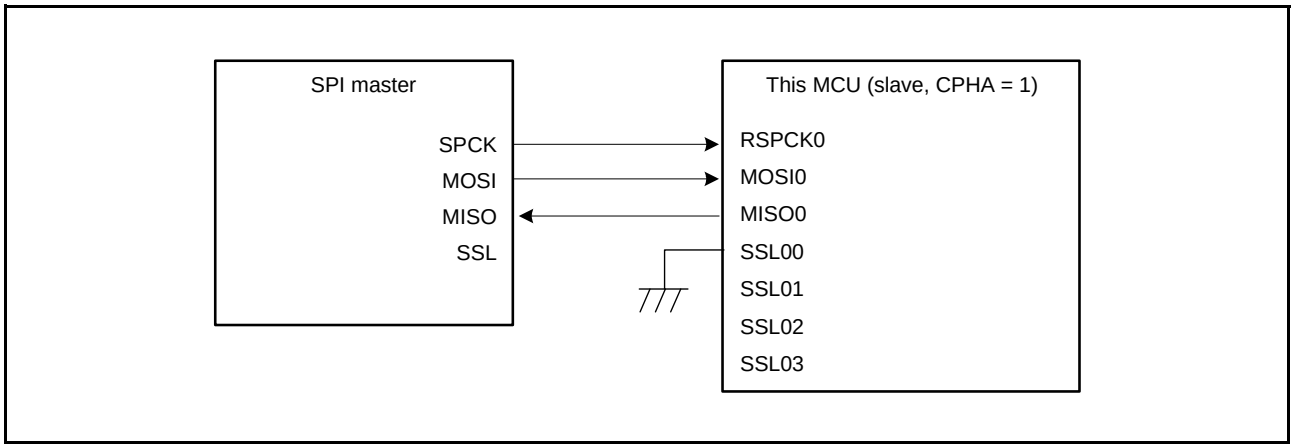


Figure 41.7 Single-Master/Single-Slave Configuration Example (This MCU = Slave, CPHA = 1)

41.3.3.3 Single Master/Multi-Slave (with This MCU Acting as Master)

Figure 41.8 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 41.8, the RSPI system is comprised of this MCU (master) and four slaves (SPI slave 0 to SPI slave 3).

The RSPCK0 and MOSI0 outputs of this MCU (master) are connected to the SPCK and MOSI inputs of SPI slave 0 to SPI slave 3. The MISO outputs of SPI slave 0 to SPI slave 3 are all connected to the MISO0 input of this MCU (master). SSL00 to SSL03 outputs of this MCU (master) are connected to the SSL inputs of SPI slave 0 to SPI slave 3, respectively.

This MCU (master) drives RSPCK0, MOSI0, and SSL00 to SSL03. Of the SPI slave 0 to SPI slave 3, the slave that receives low-level input into the SSL input drives MISO.

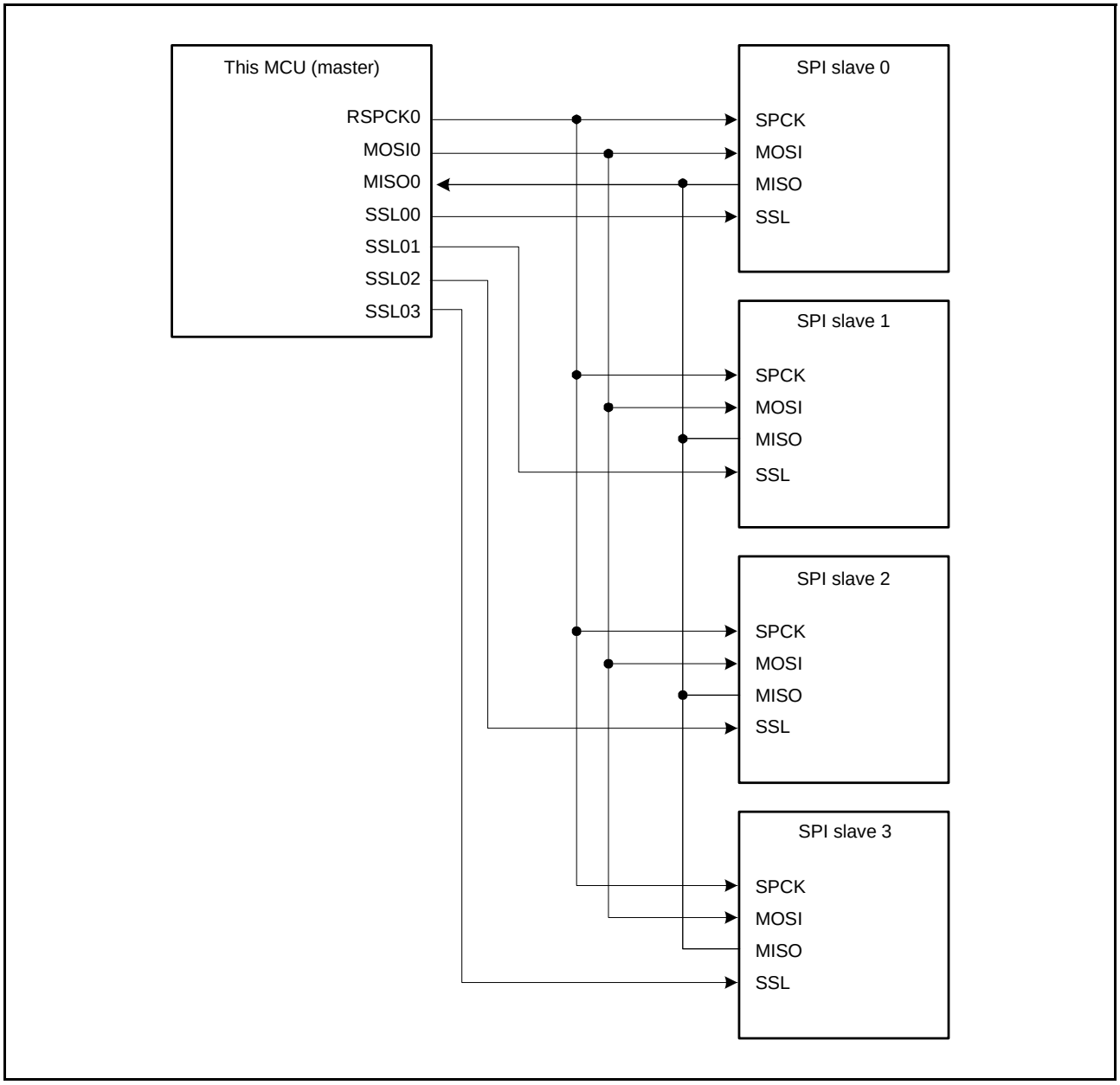


Figure 41.8 Single-Master/Multi-Slave Configuration Example (This MCU = Master)

41.3.3.4 Single Master/Multi-Slave (with This MCU Acting as Slave)

Figure 41.9 shows a single-master/multi-slave RSPI system configuration example when this MCU is used as a slave. In the example of Figure 41.9, the RSPI system is comprised of an SPI master and two MCUs (slave X and slave Y). The SPCK and MOSI outputs of the SPI master are connected to the RSPCK0 and MOSI0 inputs of the MCUs (slave X and slave Y). The MISO0 outputs of the MCUs (slave X and slave Y) are all connected to the MISO input of the SPI master. SSLX and SSLY outputs of the SPI master are connected to the SSL00 inputs of the MCUs (slave X and slave Y), respectively.

The SPI master always drives SPCK, MOSI, SSLX, and SSLY. Of the MCUs (slave X and slave Y), the slave that receives low-level input into the SSL00 input drives MISO0.

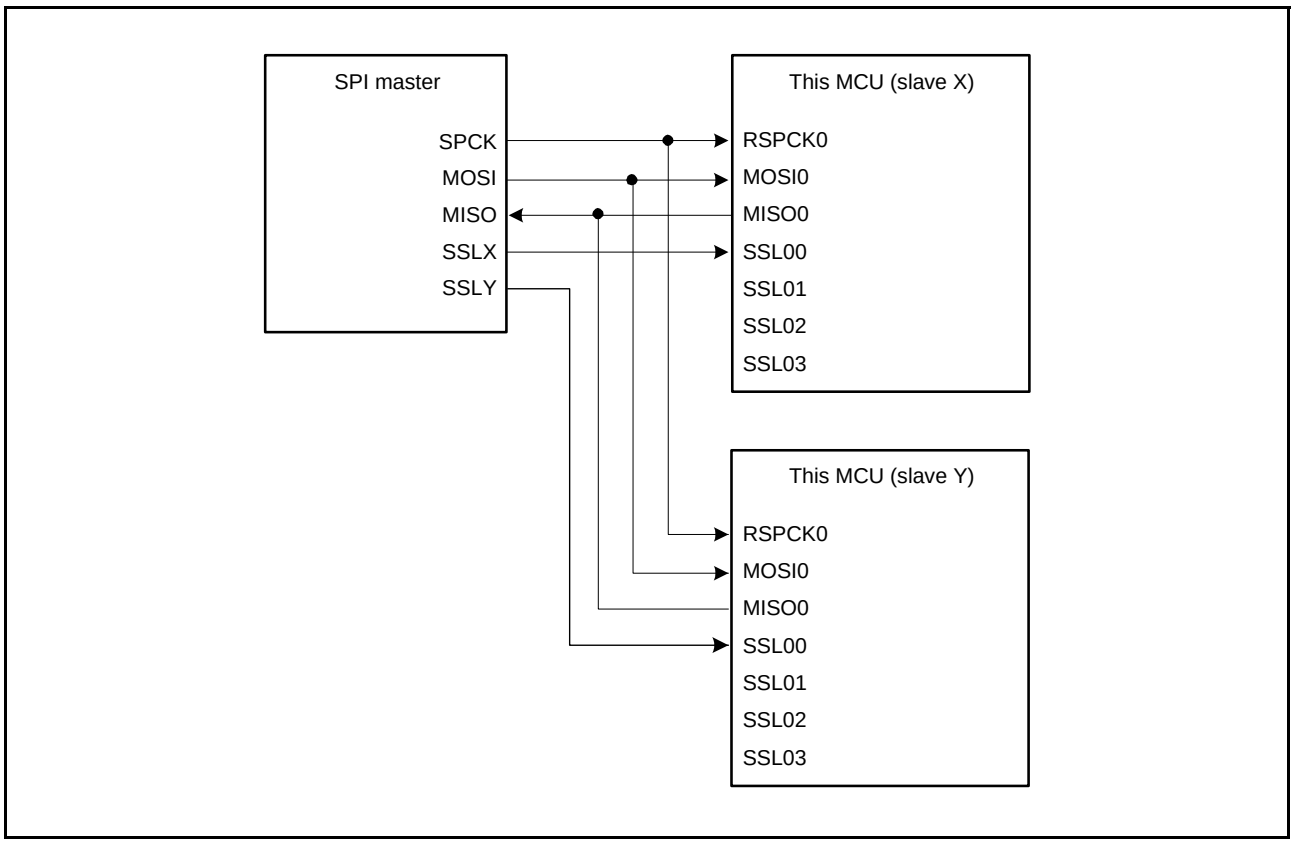


Figure 41.9 Single-Master/Multi-Slave Configuration Example (This MCU = Slave)

41.3.3.5 Multi-Master/Multi-Slave (with This MCU Acting as Master)

Figure 41.10 shows a multi-master/multi-slave RSPI system configuration example when this MCU is used as a master. In the example of Figure 41.10, the RSPI system is comprised of two MCUs (master X and master Y) and two SPI slaves (SPI slave 1 and SPI slave 2).

The RSPCK0 and MOSI0 outputs of the MCUs (master X and master Y) are connected to the SPCK and MOSI inputs of SPI slaves 1 and 2. The MISO outputs of SPI slaves 1 and 2 are connected to the MISO0 inputs of the MCUs (master X and master Y). Any generic port Y output from this MCU (master X) is connected to the SSL00 input of this MCU (master Y). Any generic port X output of this MCU (master Y) is connected to the SSL00 input of this MCU (master X). The SSL01 and SSL02 outputs of the MCUs (master X and master Y) are connected to the SSL inputs of the SPI slaves 1 and 2. In this configuration example, since the system can be comprised solely of SSL00 input, and SSL01 and SSL02 outputs for slave connections, the SSL03 output of this MCU is not required.

This MCU drives RSPCK0, MOSI0, SSL01, and SSL02 when the SSL00 input level is high. When the SSL00 input level is low, this MCU detects a mode fault error, sets RSPCK0, MOSI0, SSL01, and SSL02 to Hi-Z, and releases the RSPI bus right to the other master. Of the SPI slaves 1 and 2, the slave that receives low-level input into the SSL input drives MISO.

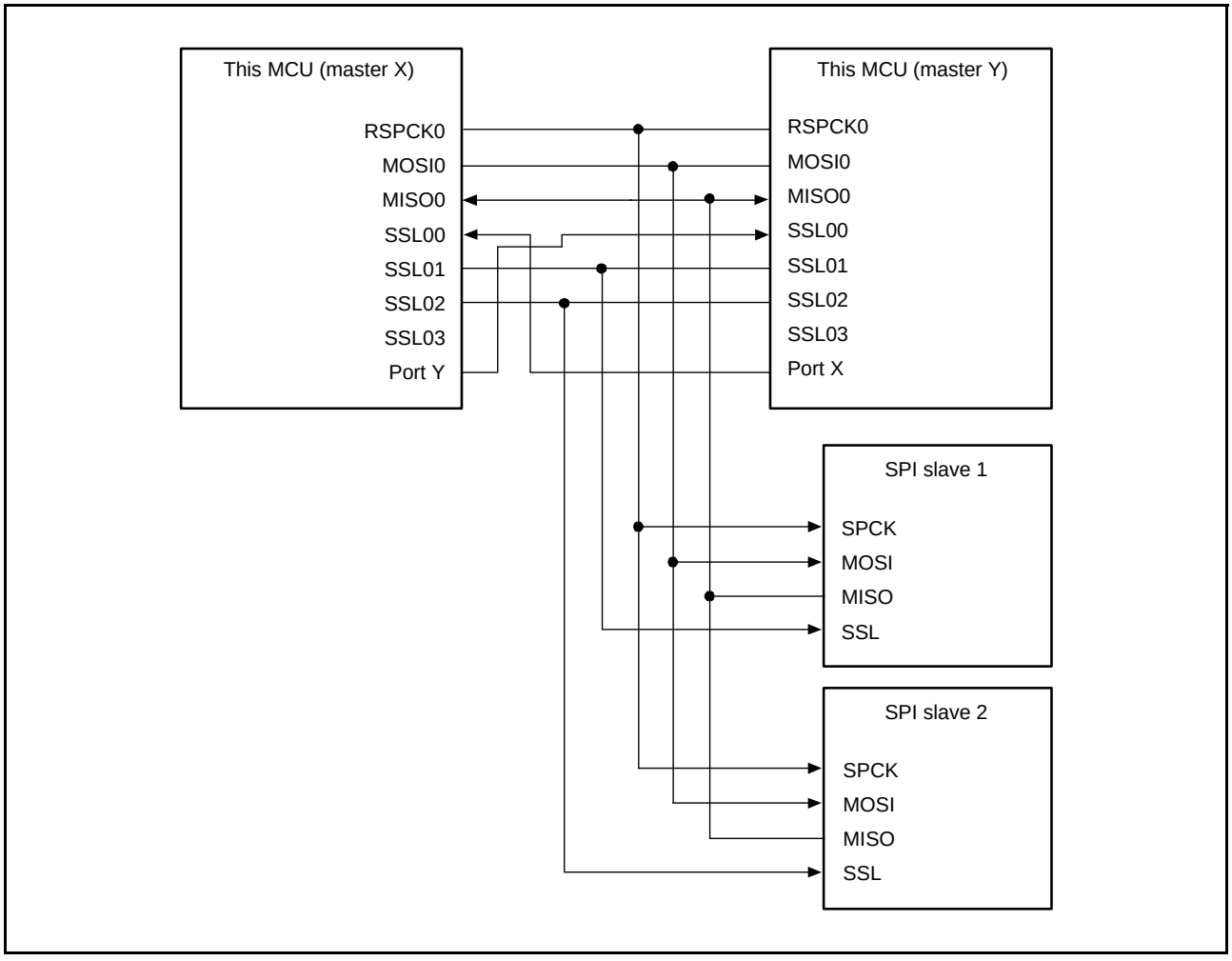


Figure 41.10 Multi-Master/Multi-Slave Configuration Example (This MCU = Master)

When setting TI SSP, enter the following levels for port X and port Y.

- Start of communication: the value of SSLP.SSL0P of the other master
- End of communication: the inverted value of SSLP.SSL0P of the other master

41.3.3.6 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Master)

Figure 41.11 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this MCU is used as a master. In the master (clock synchronous operation)/slave (clock synchronous operation) configuration, SSL00 to SSL03 of this MCU (master) are not used.
This MCU (master) drives the RSPCK0 and MOSI0. The SPI slave drives the MISO.

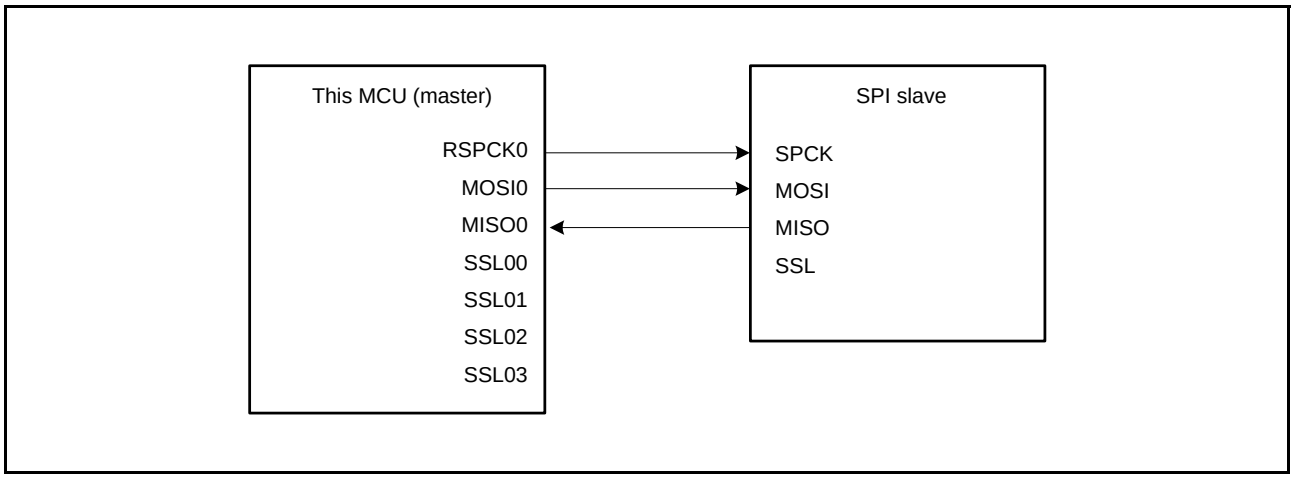


Figure 41.11 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Master)

41.3.3.7 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This MCU Acting as Slave)

Figure 41.12 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPI system configuration example when this MCU is used as a slave. When this MCU is to operate as a slave (clock synchronous operation), this MCU (slave) drives the MISO0 and the SPI master drives the SPCK and MOSI. In addition, SSL00 to SSL03 of this MCU (slave) are not used.
Only in the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, this MCU (slave) can execute serial transfer.

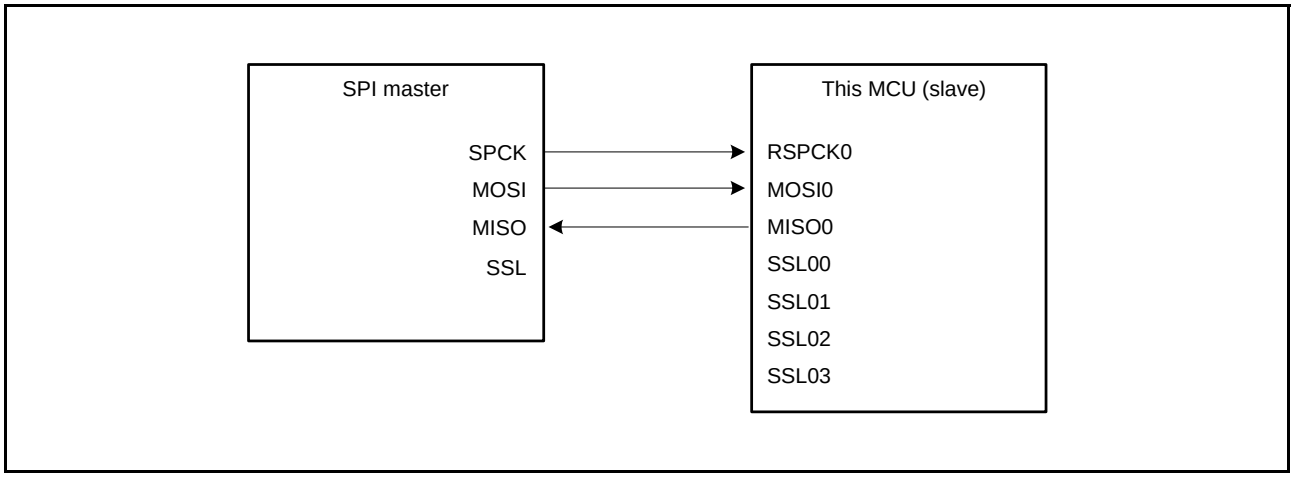


Figure 41.12 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This MCU = Slave, CPHA = 1)

41.3.4 Data Format

The RSPIA's data format depends on the settings in RSPI command register m (SPCMDm) (m = 0 to 7) and the parity enable bit in RSPI control register (SPCR.SPPE). Regardless of whether the MSB or LSB is first, the RSPIA treats the range from the LSB bit in the RSPI data register (SPDR) to the selected data length as transmit data.

41.3.4.1 Data Format of One Frame

The format of one frame of data before or after transfer is shown below.

(a) With Parity Disabled

When parity is disabled, transmission or reception of data proceeds with the length in bits selected in the RSPI data length setting bits in RSPI command register m (SPCMDm.SPB[4:0]).

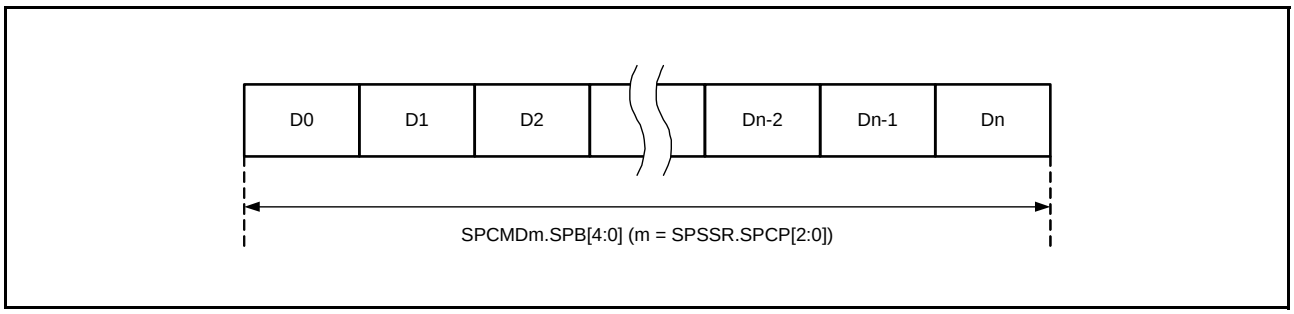


Figure 41.13 Outline of the Data Format (with Parity Disabled)

(b) With Parity Enabled

When parity is enabled, transmission or reception of data proceeds with the length in bits selected in the RSPI data length setting bits in RSPI command register m (SPCMDm.SPB[4:0]). In this case, however, the last bit is a parity bit.

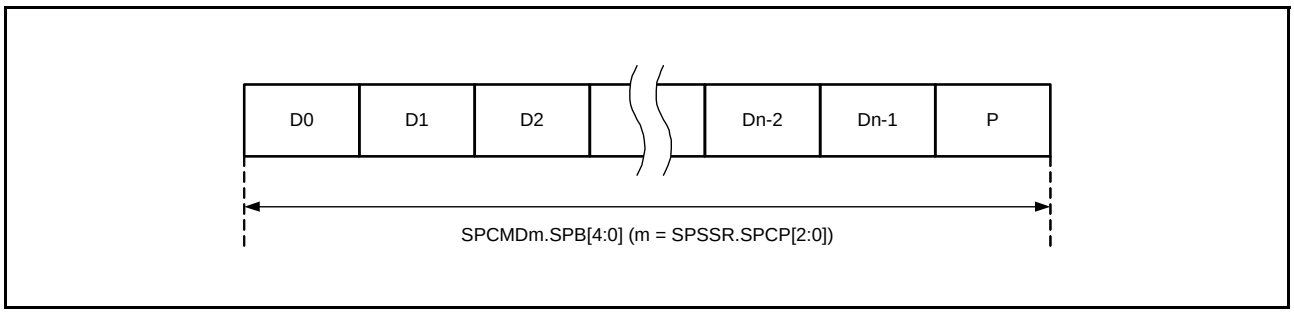


Figure 41.14 Outline of the Data Format (with Parity Enabled)

41.3.4.2 When Parity is Disabled (SPCR.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 41.15 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.

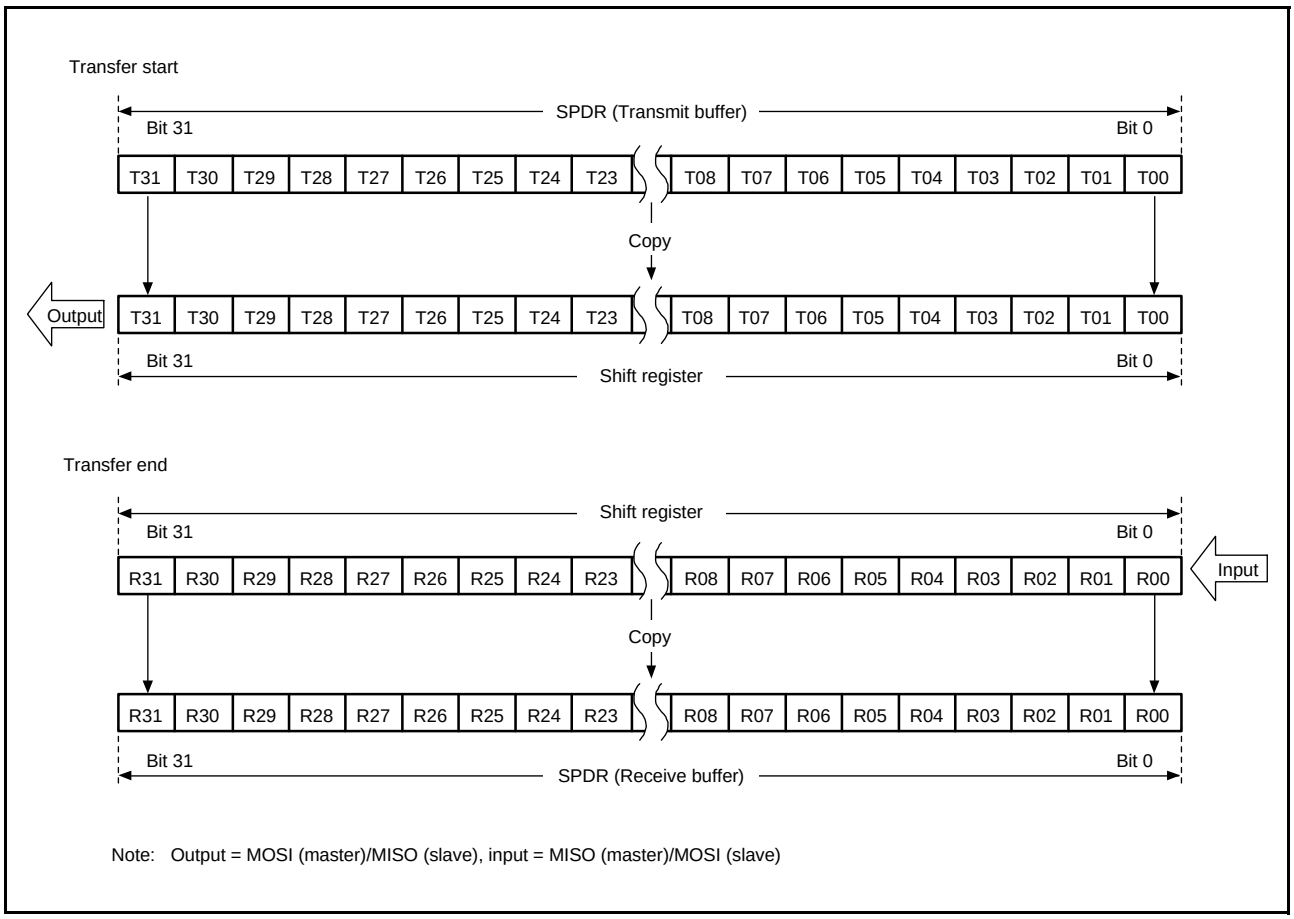


Figure 41.15 MSB First Transfer (32-Bit Data, Parity Disabled)

(2) MSB First Transfer (24-Bit Data)

Figure 41.16 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.

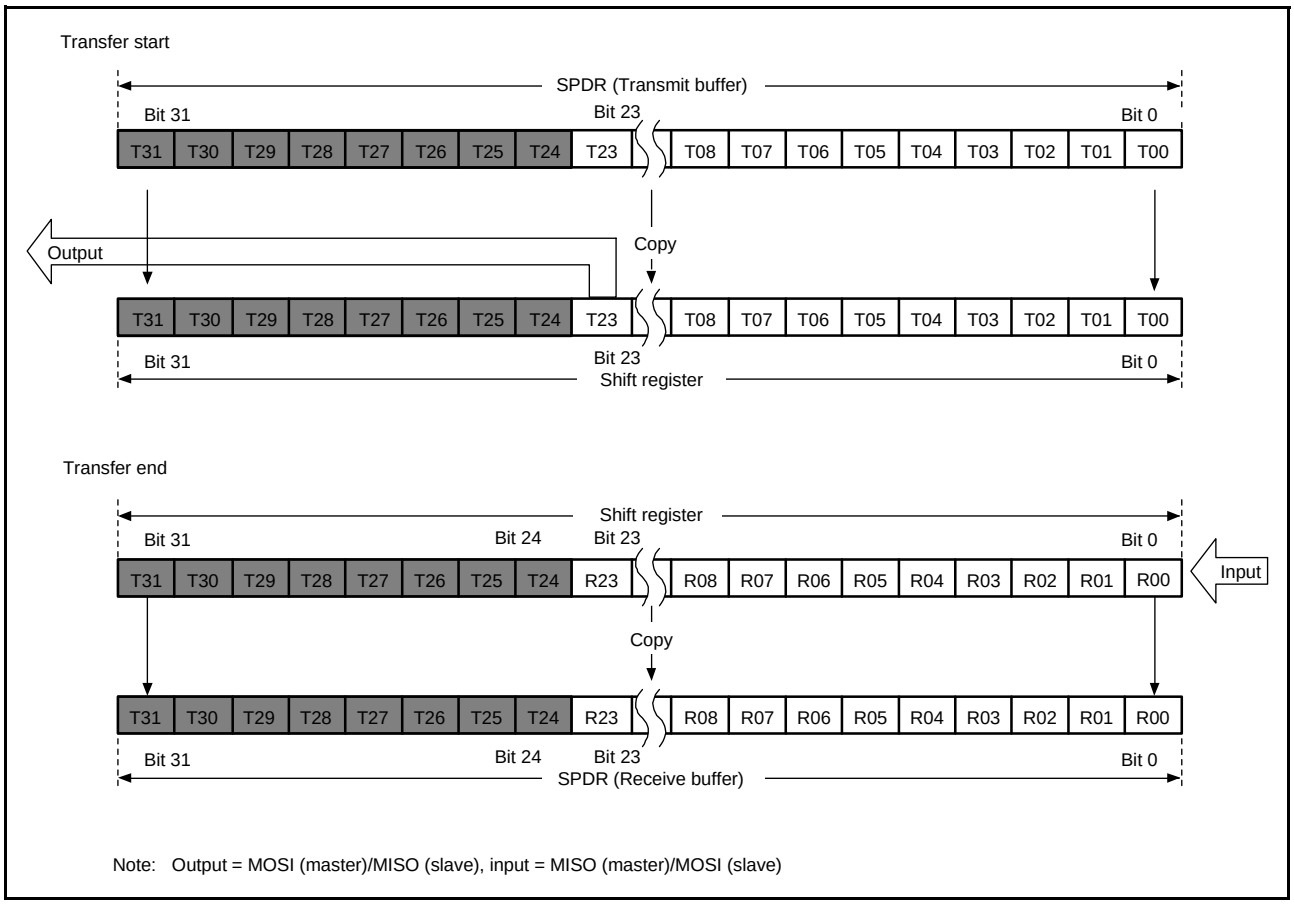


Figure 41.16 MSB First Transfer (24-Bit Data, Parity Disabled)

(3) LSB First Transfer (32-Bit Data)

Figure 41.17 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 values in the shift register are reordered bit by bit to obtain the order R31 to R00 for copying to the receive buffer.

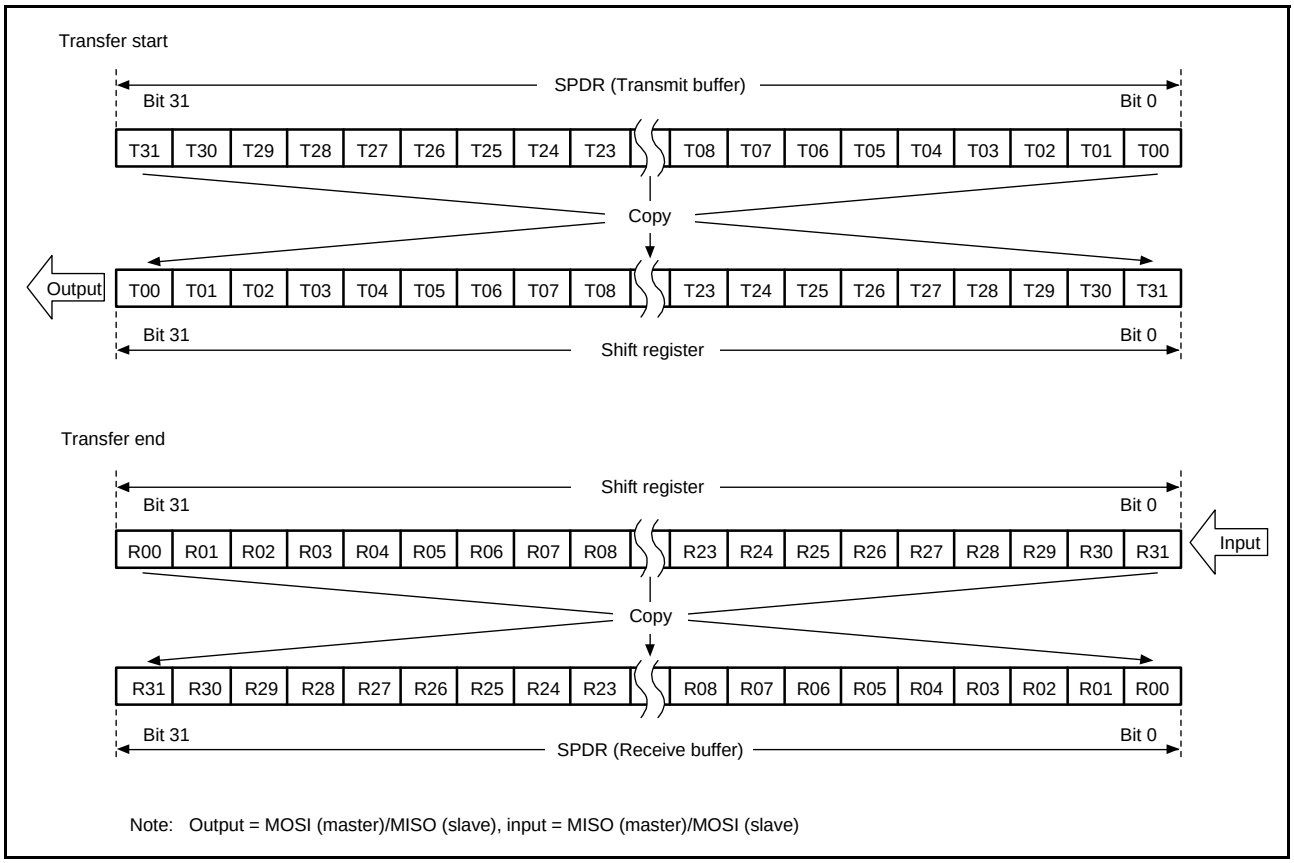


Figure 41.17 LSB First Transfer (32-Bit Data, Parity Disabled)

(4) LSB First Transfer (24-Bit Data)

Figure 41.18 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 values in the shift register are reordered bit by bit to obtain the order R23 to R00 for copying to the receive buffer.

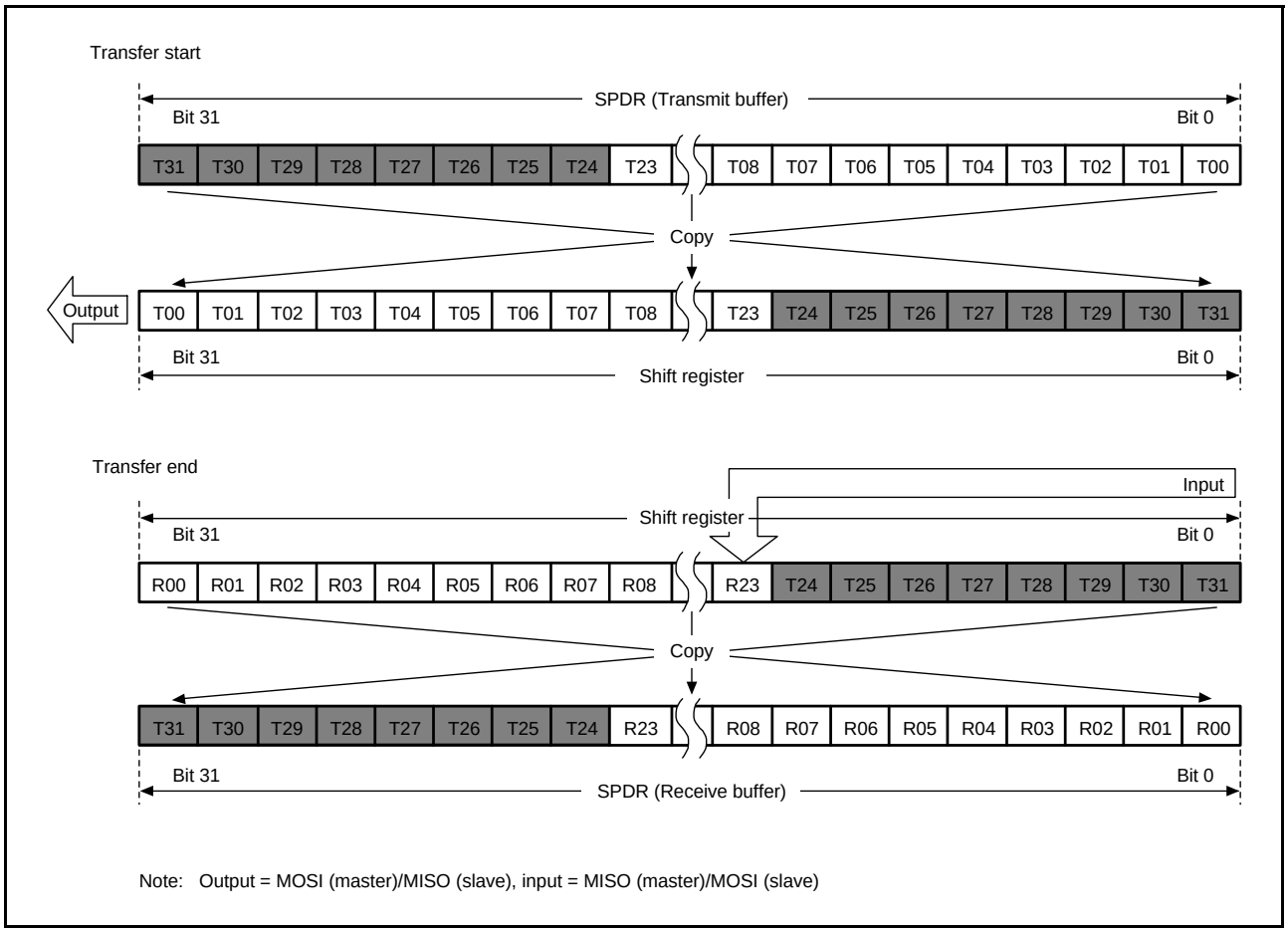


Figure 41.18 LSB First Transfer (24-Bit Data, Parity Disabled)

41.3.4.3 When Parity is Enabled (SPCR.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 41.19 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 receive buffer, the data from R31 to P are checked to judge the parity error.

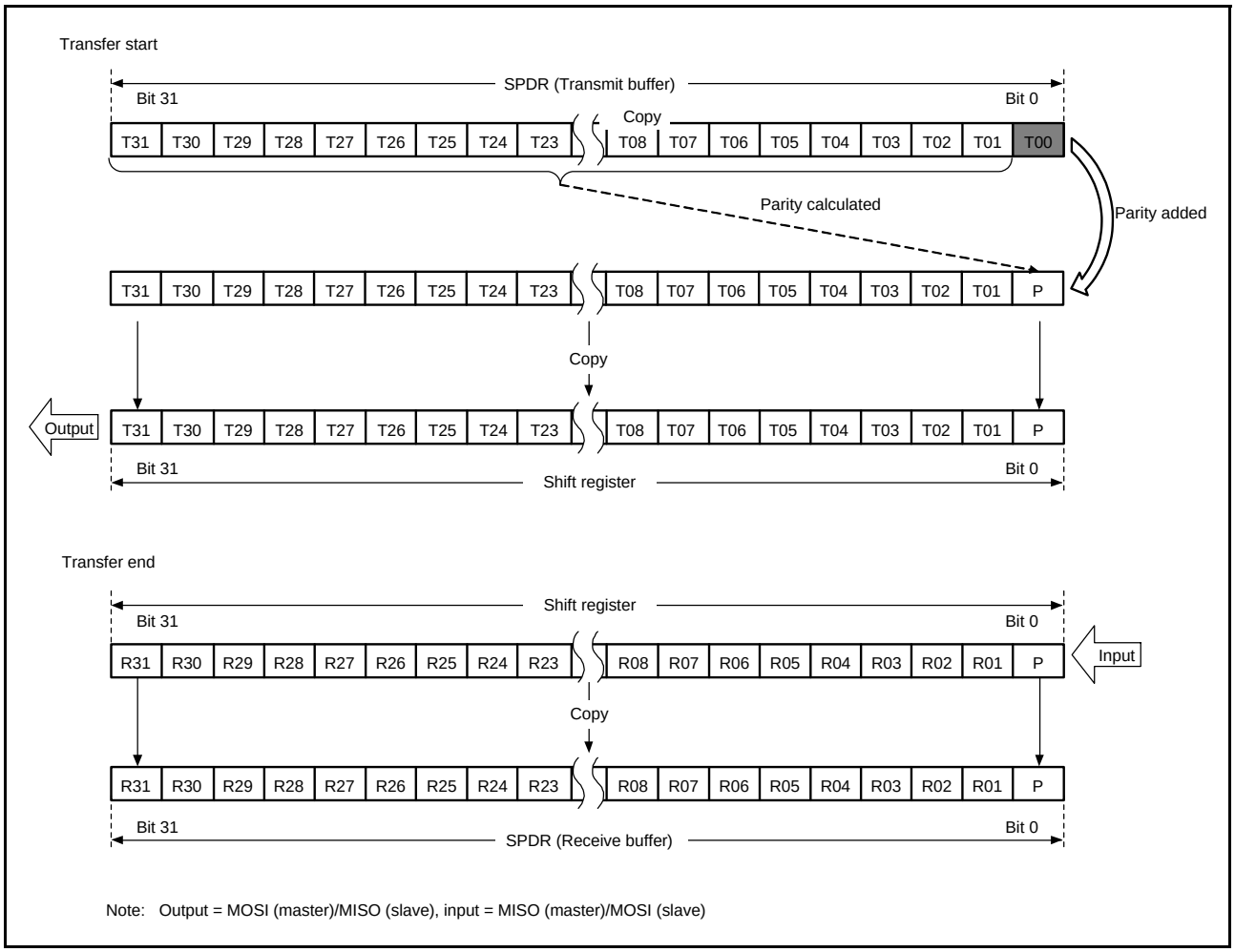


Figure 41.19 MSB First Transfer (32-Bit Data, Parity Enabled)

(2) MSB First Transfer (24-Bit Data)

Figure 41.20 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 T23 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data are transmitted in the order T23, T22, …, T01, and P.
In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R23 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the receive buffer, the data from R23 to P are checked to judge the parity error.

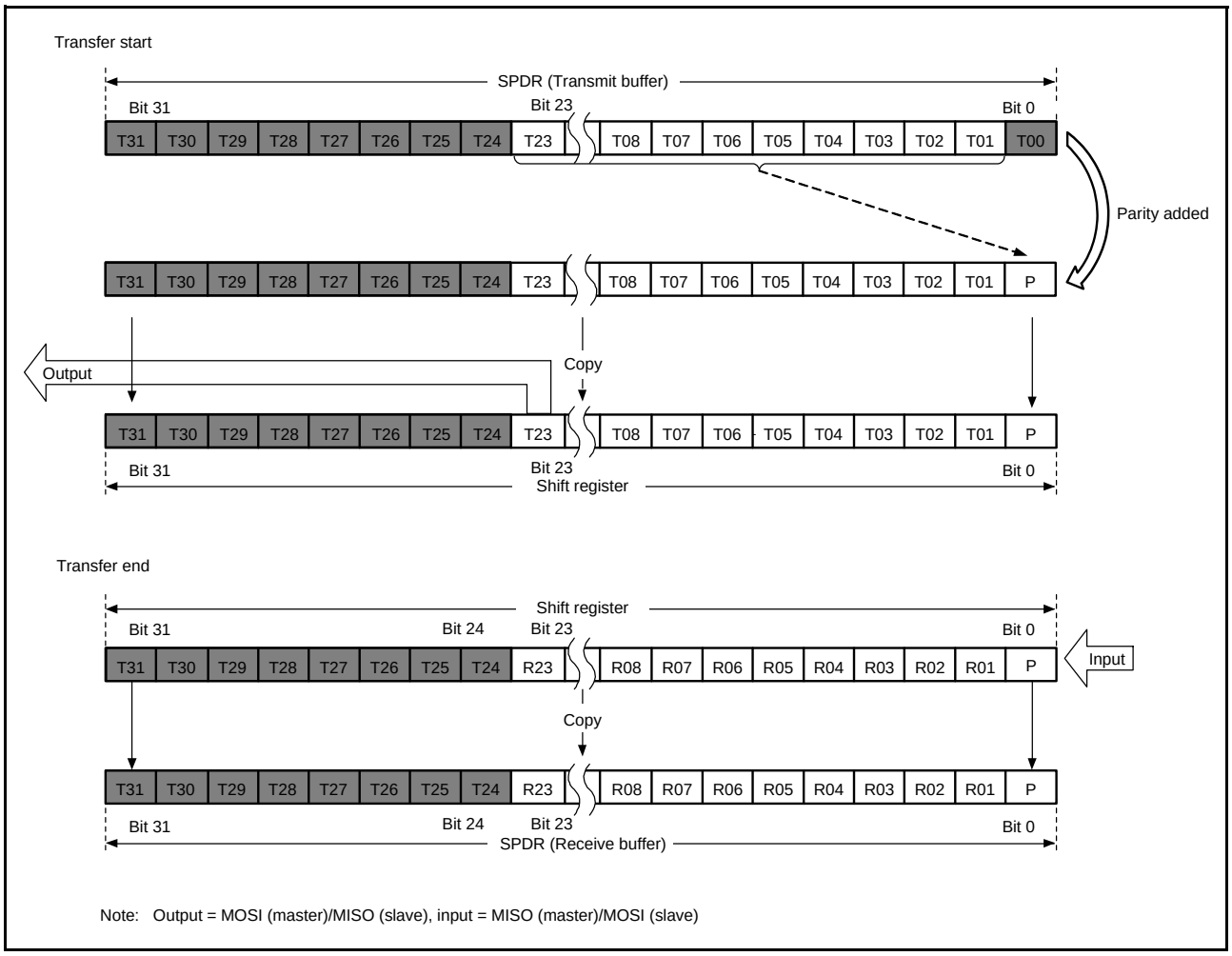


Figure 41.20 MSB First Transfer (24-Bit Data, Parity Enabled)

(3) LSB First Transfer (32-Bit Data)

Figure 41.21 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 T00. This replaces the final bit, T31, and the whole is copied to the shift register. Data are transmitted in the order T00, T01, …, T30, and P.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R00 to P have been collected after input of the required number of cycles of RSPCK, the values in the shift register are reordered bit by bit to obtain the order P to R00 for copying to the receive buffer. On copying of data to the receive buffer, the data from R00 to P are checked to judge the parity error.

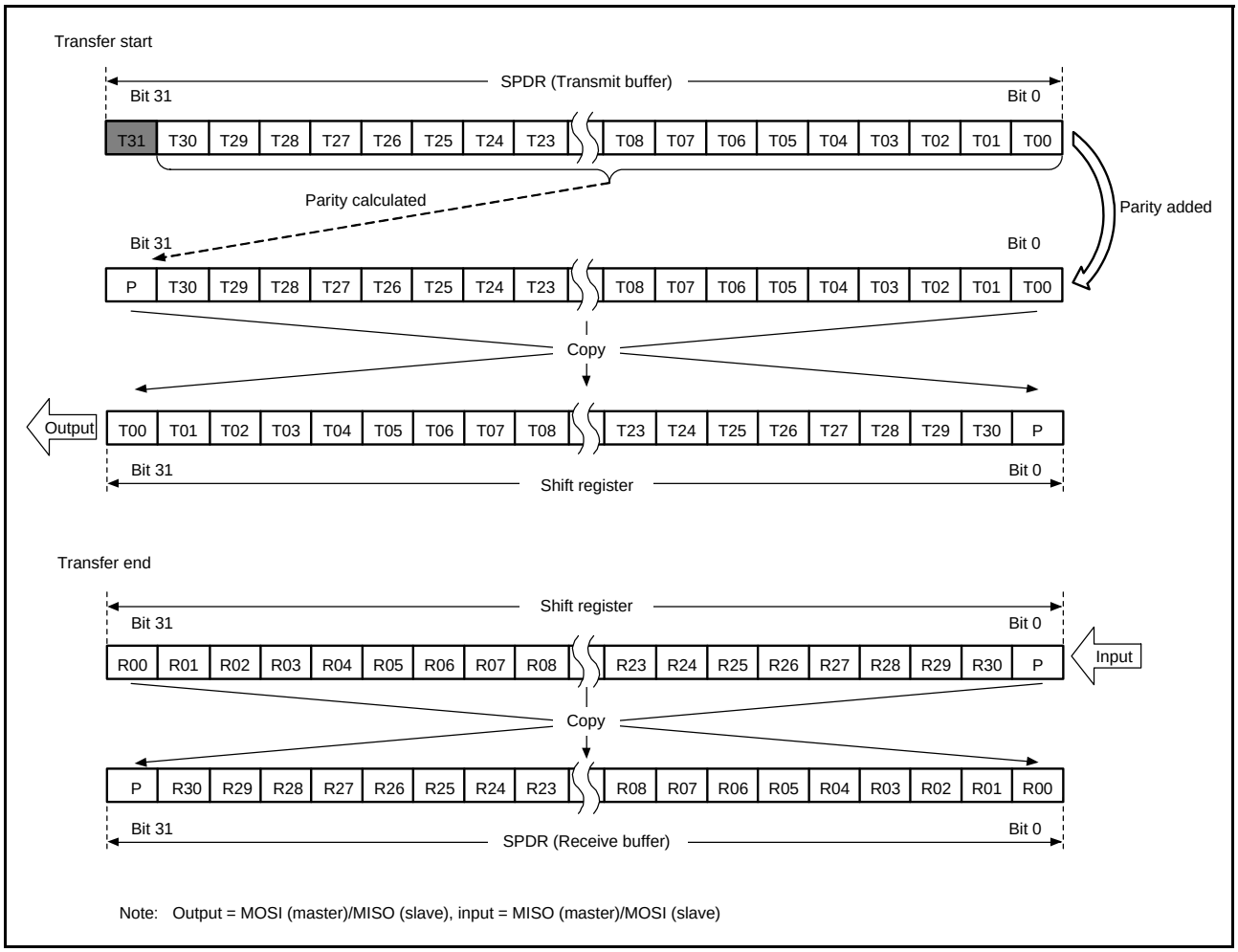


Figure 41.21 LSB First Transfer (32-Bit Data, Parity Enabled)

(4) LSB First Transfer (24-Bit Data)

Figure 41.22 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 R00 to P have been collected after input of the required number of cycles of RSPCK, the values in the shift register are reordered bit by bit to obtain the order P (bit 23) to R00 (bit 0) for copying to the receive buffer. On copying of data to the receive buffer, the data from R00 to P are checked to judge the parity error.

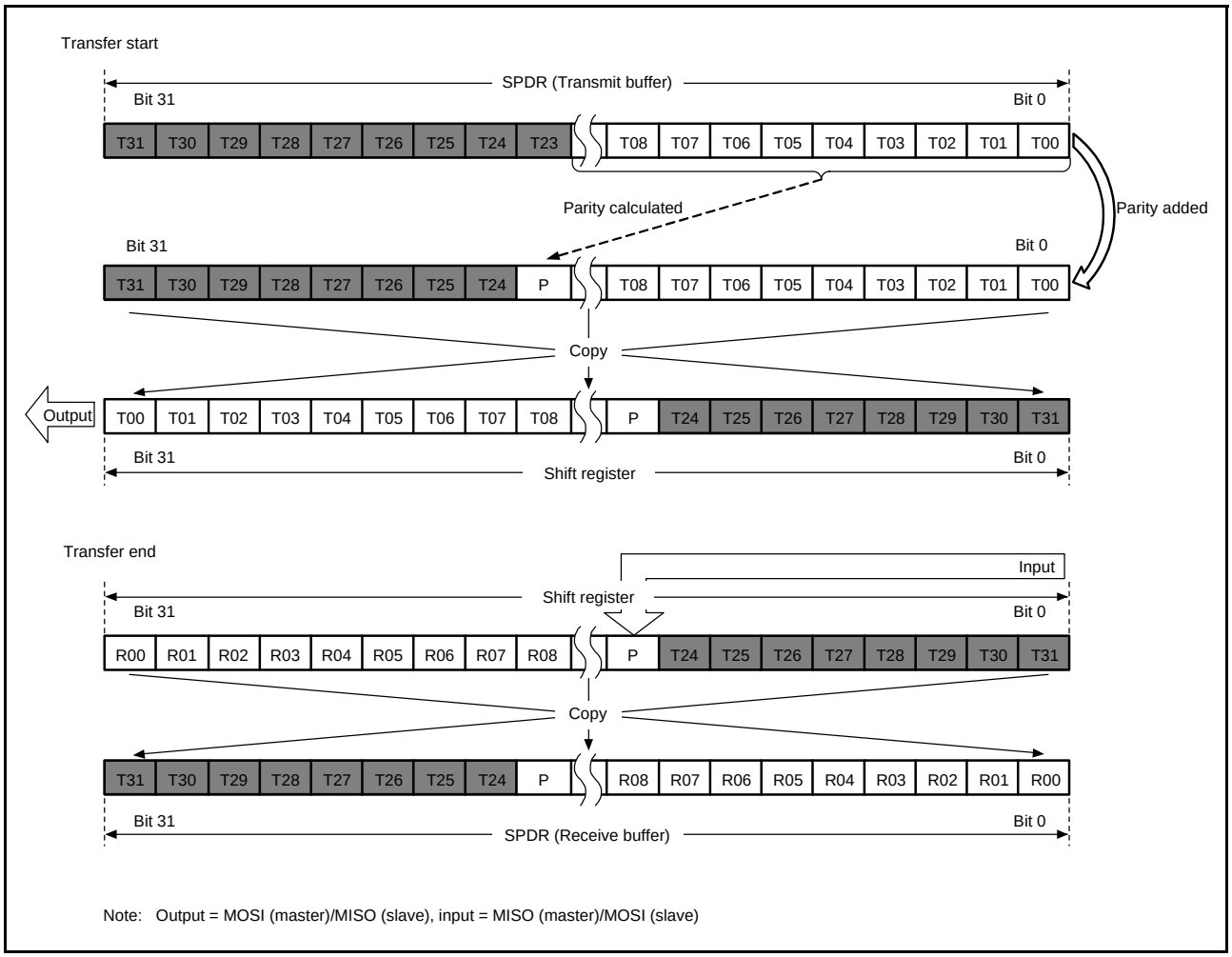


Figure 41.22 LSB First Transfer (24-Bit Data, Parity Enabled)

41.3.4.4 Byte Swap Transmission

When the SPDCR.BYSW bit is 1 (byte swapping of SPDR data enabled), data bytes written in the transmit buffer (SPDR) will be swapped (in 8 bit units) when the data is transferred to the shift register. Figure 41.23 shows data transfer between the SPDR register and the shift register when data length is 32 bits in combination of MSB/LSB first with byte swapping enabled/disabled.

(1) MSB first transfer (when the byte swap is disabled)
Data (Byte 3 [T31 to T24] to Byte 0 [T07 to T00]) in the transmit buffer are copied to the shift register. Bit values in the shift register are shifted and transmitted in the order of T31, T30, …, and T00 as transmit data.

(2) MSB first transfer (when the byte swap is enabled)
Byte values of the transmit buffer (Byte 3 [T31 to T24] to Byte 0 [T07 to T00]) are reversed in byte units and are copied to the shift register in the order of Byte 0 [T07 to T00] to Byte 3 [T31 to T24].
Bit values in the shift register are shifted and transmitted in the order of T07, T06, …, T00, T15, T14, …, T08, T23, T22, …, T16, T31, T30, …, and T24 as transmit data.

(3) LSB first transfer (when the byte swap is disabled)
Bit values of the transmit buffer (Byte 3 [T31 to T24] to Byte 0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte 0 [T00 to T07] to Byte 3 [T24 to T31].
Bit values in the shift register are shifted and transmitted in the order of T00, T01, …, and T31 as transmit data.

(4) LSB first transfer (when the byte swap is enabled)
Bit values of each byte of the transmit buffer (Byte 3 [T31 to T24] to Byte 0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte 3 [T24 to T31] to Byte 0 [T00 to T07].
Bit values in the shift register are shifted and transmitted in the order of T24, T25, …, T31, T16, T17, …, T23, T08, T09, …, T15, T00, T01, …, and T07 as transmit data.

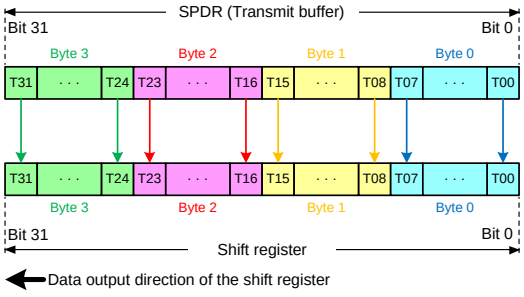
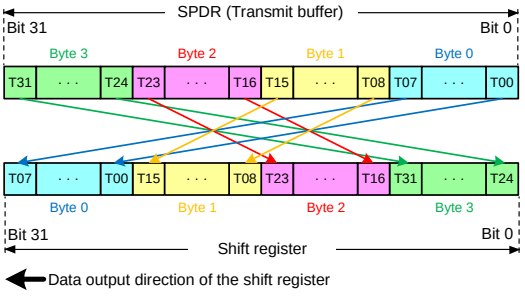
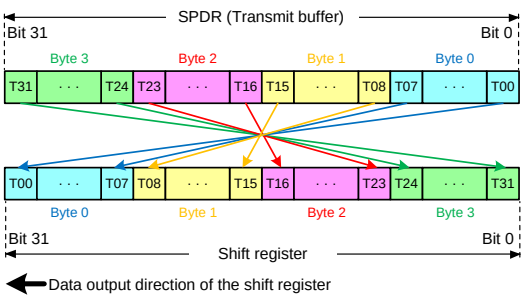
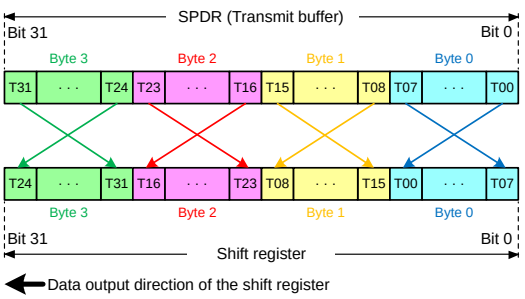
32-bit Byte Swap Specifications	Byte Swap is Disabled (BYSW = 0)	Byte Swap is Enabled (BYSW = 1)
MSB-first (LSBF = 0)		
LSB-first (LSBF = 1)		

Figure 41.23 Transmit Data Handling in MSB/LSB First with Byte Swapping Enabled/Disabled (Data Length = 32 bits)

Figure 41.24 shows data transfer between the SPDR register and the shift register when data length is 16 bits in combination of MSB/LSB first with byte swapping enabled/disabled.

(1) MSB first transfer (when the byte swap is disabled)
Data (Byte 1 [T15 to T08] to Byte 0 [T07 to T00]) in the transmit buffer are copied to the shift register in the order of Byte 1 [T15 to T08] to Byte 0 [T07 to T00]. Bit values in the shift register are shifted and transmitted in the order of T15, T14, …, and T00 as transmit data.

(2) MSB first transfer (when the byte swap is enabled)
Byte values of the transmit buffer (Byte 1 [T15 to T08] to Byte 0 [T07 to T00]) are reversed in byte units and are copied to the shift register in the order of Byte 0 [T07 to T00] to Byte 1 [T15 to T08]. Bit values in the shift register are shifted and transmitted in the order of T07, T06, …, T00, T15, T14, …, T08 as transmit data.

(3) LSB first transfer (when the byte swap is disabled)
Bit values of the transmit buffer (Byte 1 [T15 to T08] to Byte 0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte 0 [T00 to T07] to Byte 1 [T15 to T08]. Bit values in the shift register are shifted and transmitted in the order of T00, T01, …, and T15 as transmit data.

(4) LSB first transfer (when the byte swap is enabled)
Bit values of each byte of the transmit buffer (Byte 1 [T15 to T08] to Byte 0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte 1 [T08 to T15] to Byte 0 [T00 to T07].
Bit values in the shift register are shifted and transmitted in the order of T08, T09, …, T15, T00, T01, …, and T07 as transmit data.

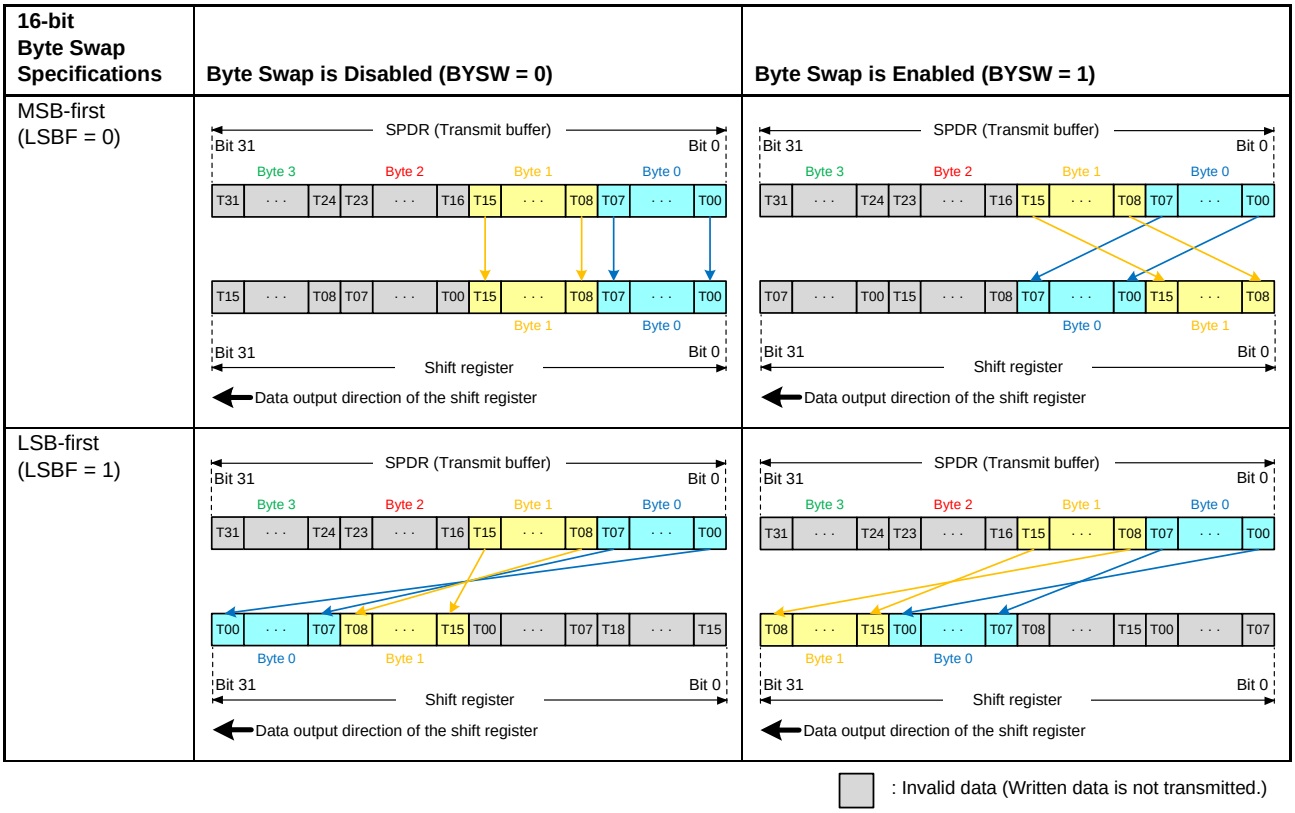


Figure 41.24 Transmit Data Handling in MSB/LSB First with Byte Swapping Enabled/Disabled (Data Length = 16 bits)

Note 1. When using the byte swap, set 16 bits or 32 bits to the data length (SPCMDm.SPB[4:0] setting). If setting the other length (4 to 15 or 17 to 31 bits), the operation is not guaranteed.

Note 2. When the byte swap is enabled, disable the parity function (SPCR.SPPE = 0). If the parity function is enabled (SPCR.SPPE = 1), the operation after the change is not guaranteed.
Note 3. Set the SPDCR.BYSW bit, while the SPCR.SPE bit is 0. If the value of the SPDCR.BYSW bit is changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

41.3.4.5 Byte Swap Reception

When the SPDCR.BYSW bit is 1 (byte swapping of SPDR data enabled), data bytes in the shift register will be swapped (in 8 bit units) when the data is transferred to the receive buffer (SPDR). Figure 41.25 shows data transfer between the shift register and the SPDR register when data length is 32 bits in combination of MSB/LSB first with byte swapping enabled/disabled.

(1) MSB first transfer (when the byte swap is disabled)
The first received data (R31) is stored in bit 0 of the shift register, and received data is shifted in the order of R31, R30, …, and R00. When necessary RSPCK cycles are input and data is stored from Byte 3 [R31 to R24] to Byte 0 [R07 to R00], the shift register value is copied to the receive buffer.

(2) MSB first transfer (when the byte swap is enabled)
The first received data (R07) is stored in bit 0 of the shift register, and received data is shifted in the order of R07, R06, …, R00, R15, R14, …, R08, R23, R22, …, R16, R31, R30, …, and R24. When necessary RSPCK cycles are input and data is stored from Byte 0 [R07 to R00] to Byte 3 [R31 to R24], byte values in the shift register are reversed in byte units and are copied to the receive buffer in the order of Byte 3 [R31 to R24] to Byte 0 [R07 to R00].

(3) LSB first transfer (when the byte swap is disabled)
The first received data (R00) is stored in bit 0 of the shift register, and received data is shifted in the order of R00, R01, …, and R31. When necessary RSPCK cycles are input and data is stored from Byte 0 [R00 to R07] to Byte 3 [R24 to R31], bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte 3 [R31 to R24] to Byte 0 [R07 to R00].

(4) LSB first transfer (when the byte swap is enabled)
The first received data (R24) is stored in bit 0 of the shift register, and received data is shifted in the order of R24, R25, …, R31, R16, R17, …, R23, R08, R09, …, R15, R00, R01, …, and R07. When necessary RSPCK cycles are input and data is stored from Byte 3 [R24 to R31] to Byte 0 [R00 to R07], bit values of each byte in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte 3 [R31 to R24] to Byte 0 [R07 to R00].

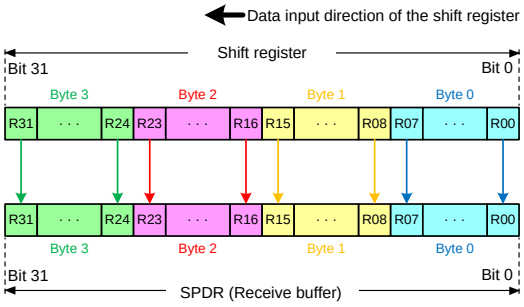
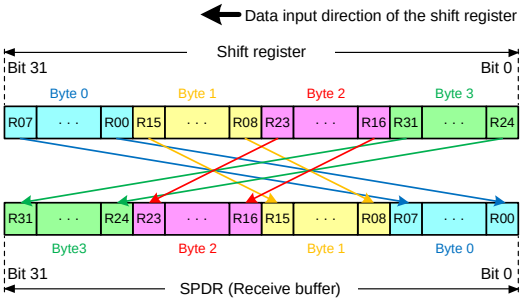
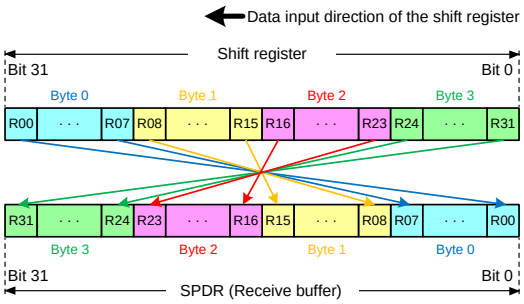
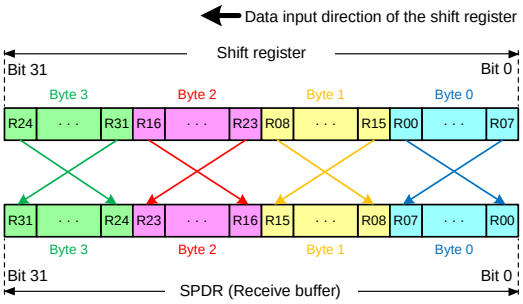
32-bit Byte Swap Specifications	Byte Swap is Disabled (BYSW = 0)	Byte Swap is Enabled (BYSW = 1)
MSB-first (LSBF = 0)		
LSB-first (LSBF = 1)		

Figure 41.25 Receive Data Handling in MSB/LSB First with Byte Swapping Enabled/Disabled (Data Length = 32 bits)

Figure 41.26 shows data transfer between the shift register and the SPDR register when data length is 16 bits in combination of MSB/LSB first with byte swapping enabled/disabled.

(1) MSB first transfer (when the byte swap is disabled)
The first received data (R15) is stored in bit 0 of the shift register, and received data is shifted in the order of R15, R14, …, and R00. When necessary RSPCK cycles are input and data is stored from Byte 3 [R31 to R24] to Byte 0 [R07 to R00], the shift register value is copied to the receive buffer.

(2) MSB first transfer (when the byte swap is enabled)
The first received data (R07) is stored in bit 0 of the shift register, and received data is shifted in the order of R07, R06, …, R00, R15, R14, …, and R08. When necessary RSPCK cycles are input and data is stored from Byte 0 [R07 to R00] to Byte 1 [R15 to R08], byte values in the shift register are reversed in byte units and are copied to the receive buffer in the order of Byte 3 [R31 to R24] to Byte 0 [R07 to R00].

(3) LSB first transfer (when the byte swap is disabled)
The first received data (R00) is stored in bit 15 of the shift register, and received data is shifted in the order of R00, R01, …, R07, R08, R09, …, and R15. When necessary RSPCK cycles are input and data is stored from Byte 0 [R00 to R07] to Byte 1 [R08 to R15], bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte 3 [R31 to R24] to Byte 0 [R07 to R00].

(4) LSB first transfer (when the byte swap is enabled)
The first received data (R08) is stored in bit 15 of the shift register, and received data is shifted in the order of R08, R09, …, R15, R00, R01, …, and R15. When necessary RSPCK cycles are input and data is stored from Byte 1 [R08 to R15] to Byte 0 [R00 to R07], bit values of each byte in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte 3 [R31 to R24] to Byte 0 [R07 to R00].

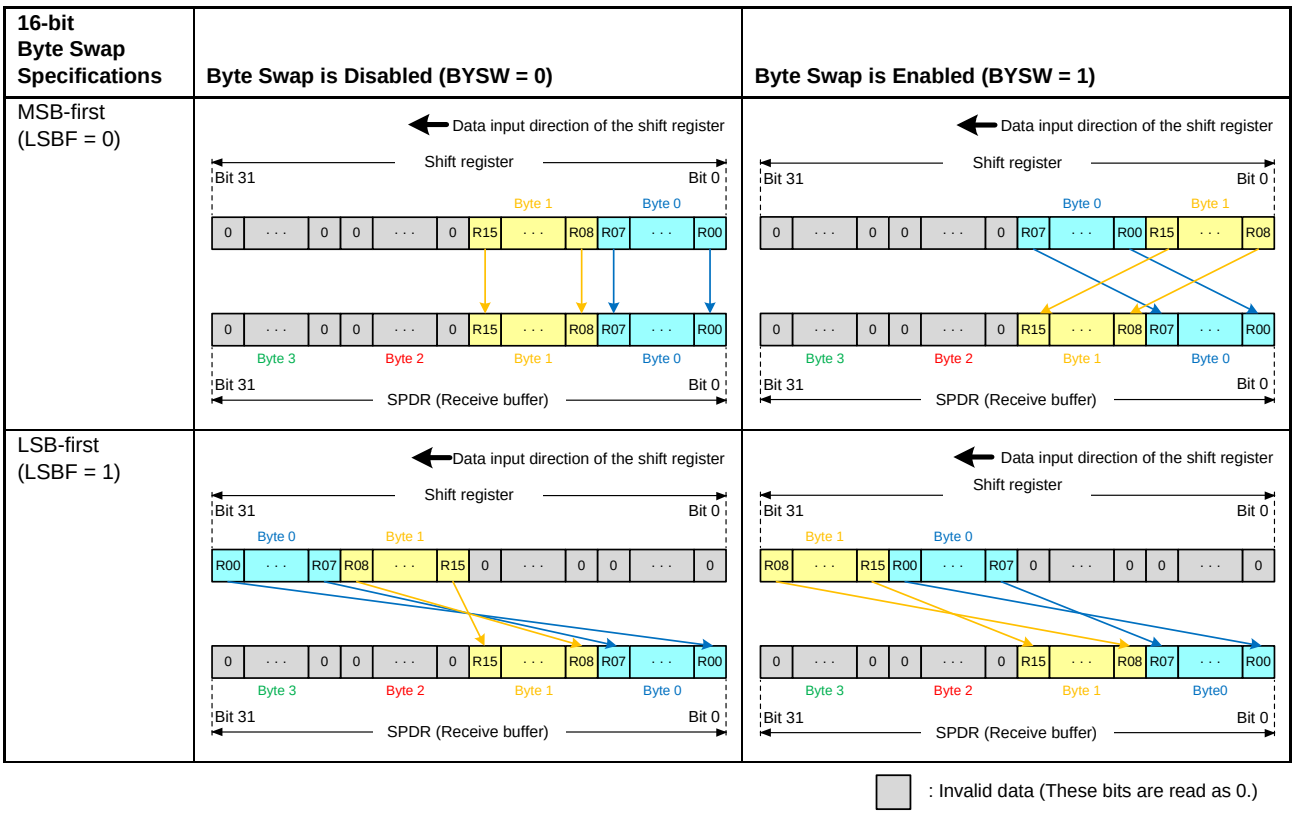


Figure 41.26 Receive Data Handling in MSB/LSB First with Byte Swapping Enabled/Disabled (Data Length = 16 bits)

Note 1. When using the byte swap, set 16 bits or 32 bits to the data length (SPCMDm.SPB[4:0] setting). If setting the other length (4 to 15 or 17 to 31 bits), the operation is not guaranteed.
Note 2. When the byte swap is enabled, disable the parity function (SPCR.SPPE = 0). If the parity function is enabled (SPCR.SPPE = 1), the operation after the change is not guaranteed.
Note 3. Set the SPDCR.BYSW bit, while the SPCR.SPE bit is 0. If the value of the SPDCR.BYSW bit is changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

41.3.5 Transfer Format (Frame Format)

41.3.5.1 CPHA = 0

Figure 41.27 shows a sample transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 0. Note that clock synchronous operation (the SPCR.SPMS bit is 1) should not performed when the RSPIA operates in slave mode (SPCR.MSTR = 0) and the CPHA bit is 0. In Figure 41.27, RSPCK0 (CPOL = 0) indicates the RSPCK0 signal waveform when the SPCMDm.CPOL bit is 0; RSPCK0 (CPOL = 1) indicates the RSPCK0 signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPIA fetches serial transfer data into the shift register. The I/O directions of the signals depend on the RSPIA settings. For details, refer to section 41.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 0, the driving of valid data to the MOSI0 and MISO0 signals commences at an SSL0n signal assertion timing. The first RSPCK0 signal change timing that occurs after the SSL0n signal assertion becomes the first transfer data fetch timing. After this timing, data is sampled at every 1 RSPCK cycle. The change timing for MOSI0 and MISO0 signals is 1/2 RSPCK cycles after the transfer data fetch timing. The CPOL bit setting does not affect the RSPCK0 signal operation timing; it only affects the signal polarity.

t1 denotes a period from an SSL0n signal assertion to RSPCK0 oscillation (RSPCK delay). t2 denotes a period from the termination of RSPCK0 oscillation to an SSL0n signal negation (SSL negation delay). t3 denotes a period in which SSL0n 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 RSPIA of this MCU is in master mode, refer to section 41.3.13.1, Master Mode Operation.

[Motorola SPI]

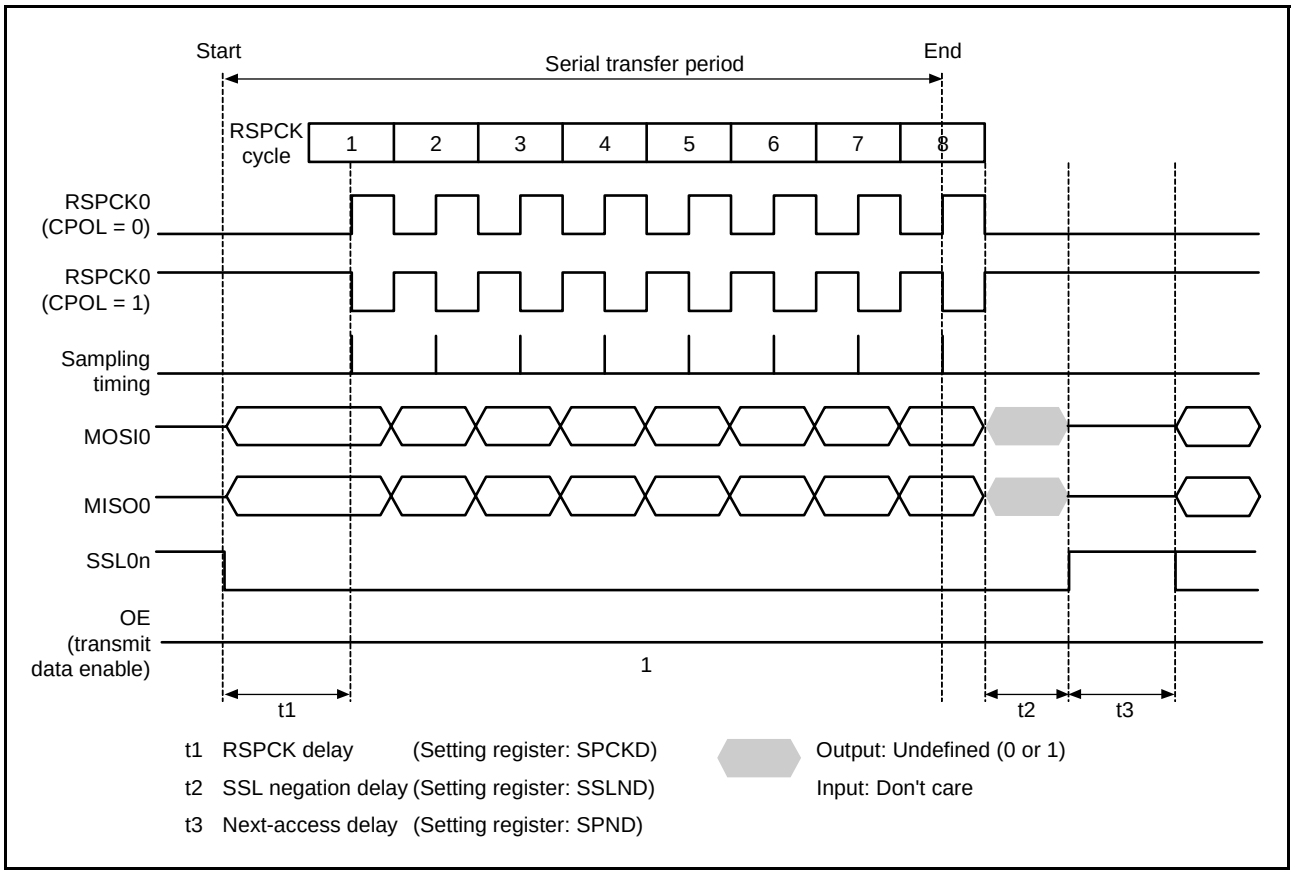


Figure 41.27 RSPI Transfer Format (CPHA = 0, FRFS = 0)

[TI SSP]

In TI SSP mode, setting the CPHA bit to 0 is not effective.

41.3.5.2 CPHA = 1

Figure 41.28 and Figure 41.29 show sample transfer formats for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 1. However, when the SPCR.SPMS bit is 1, the SSL0n signals are not used, and only the three signals RSPCK0, MOSI0, and MISO0 handle communications. In Figure 41.28, RSPCK0 (CPOL = 0) indicates the RSPCK0 signal waveform when the SPCMDm.CPOL bit is 0; RSPCK0 (CPOL = 1) indicates the RSPCK0 signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPIA fetches serial transfer data into the shift register. The I/O directions of the signals depend on the RSPI mode (master or slave). For details, refer to section 41.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 1, the driving of invalid data to the MISO0 signal commences at an SSL0n signal assertion timing. The output of valid data to the MOSI0 and MISO0 signals commences at the first RSPCK0 signal change timing that occurs after the SSL0n signal assertion. After this timing, data is updated at every 1 RSPCK cycle. The transfer data fetch timing is 1/2 RSPCK cycles after the data update timing. The SPCMDm.CPOL bit setting does not affect the RSPCK0 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 RSPIA of this MCU is in master mode, refer to section 41.3.13.1, Master Mode Operation.

[Motorola SPI]

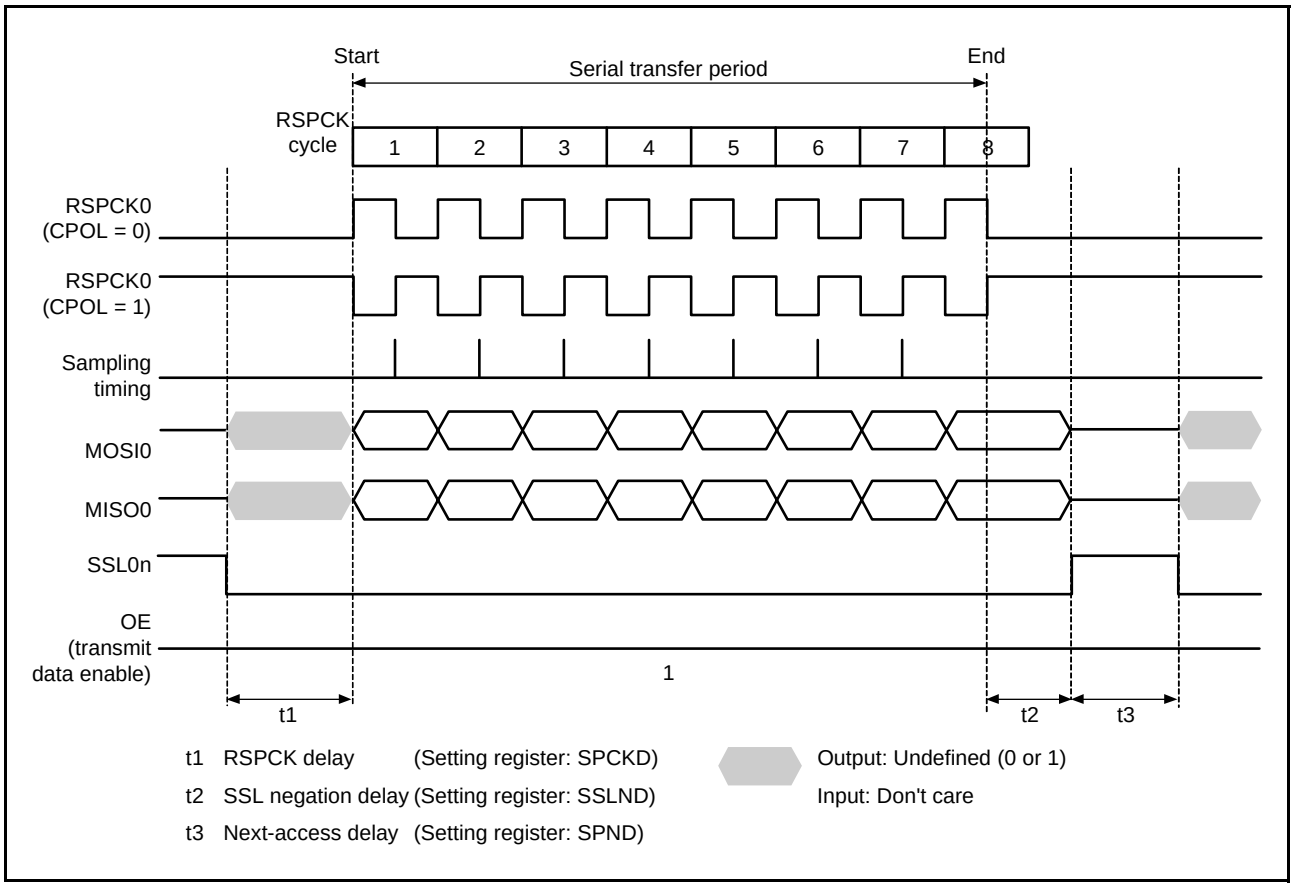


Figure 41.28 RSPI Transfer Format (CPHA = 1, FRFS = 0)

[TI SSP]

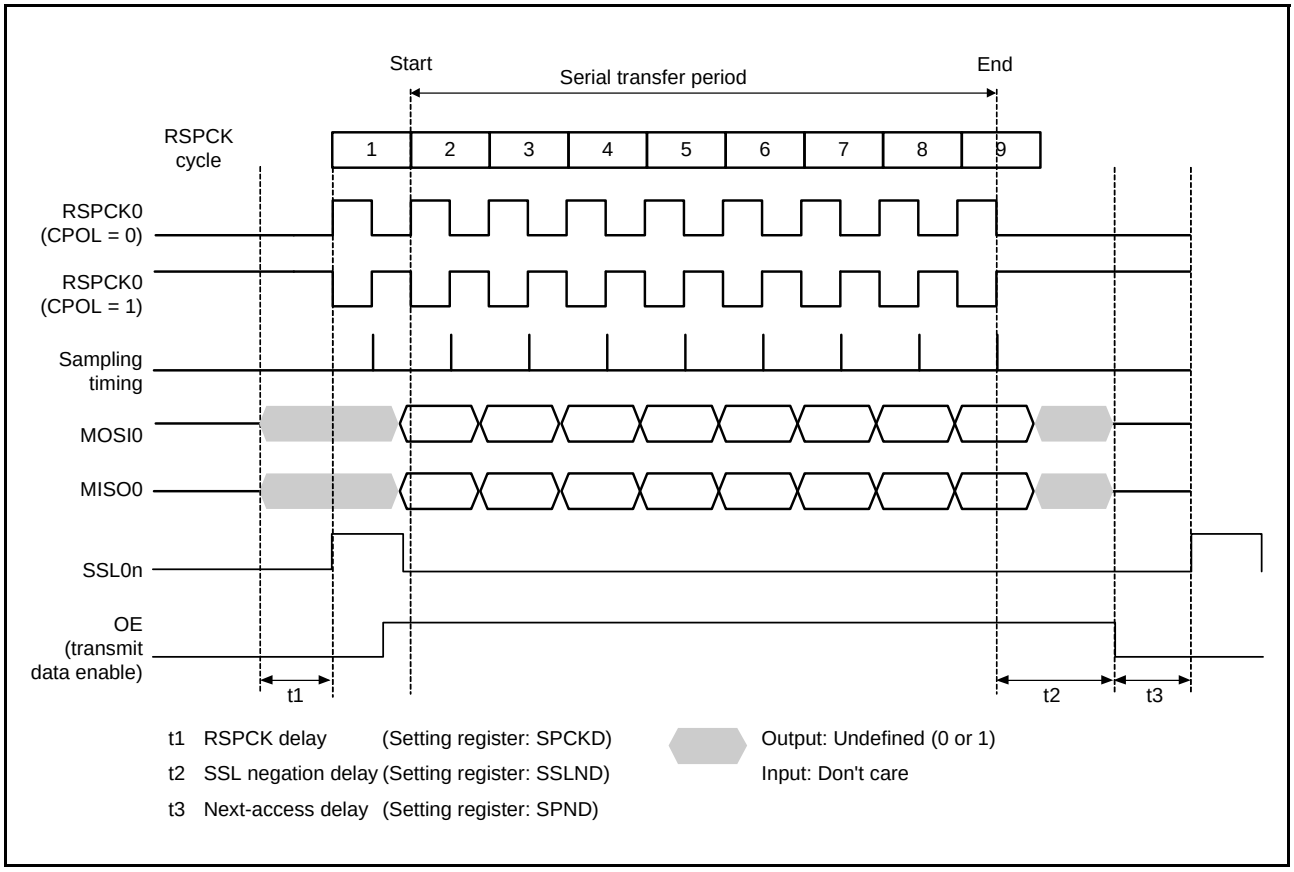


Figure 41.29 RSPI Transfer Format (CPHA = 1, FRFS = 1)

41.3.6 Communication Mode

Transmit-receive mode, transmit-only mode, and receive-only mode can be selected by the communication mode select bit (SPCR.CMMD[1:0]).

The SPDR access shown in Figure 41.30, Figure 41.31 and Figure 41.32 indicate the condition of access to the SPDR register, where W denotes a write cycle.

41.3.6.1 Transmit-Receive Mode (SPCR.CMMD[1:0] = 00b)

Figure 41.30 shows an example of operation when the communication mode select bits (SPCR.CMMD[1:0]) are set to 00b. In the example in Figure 41.30, the RSPIA performs an 8-bit serial transfer in which SPFCR.TTRG[1:0] = 00b, SPFCR.RTRG[1:0] = FIFO stage – 1, SPCMDm.CPHA = 1, and SPCMDm.CPOL = 0. The numbers given under the RSPCK0 waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

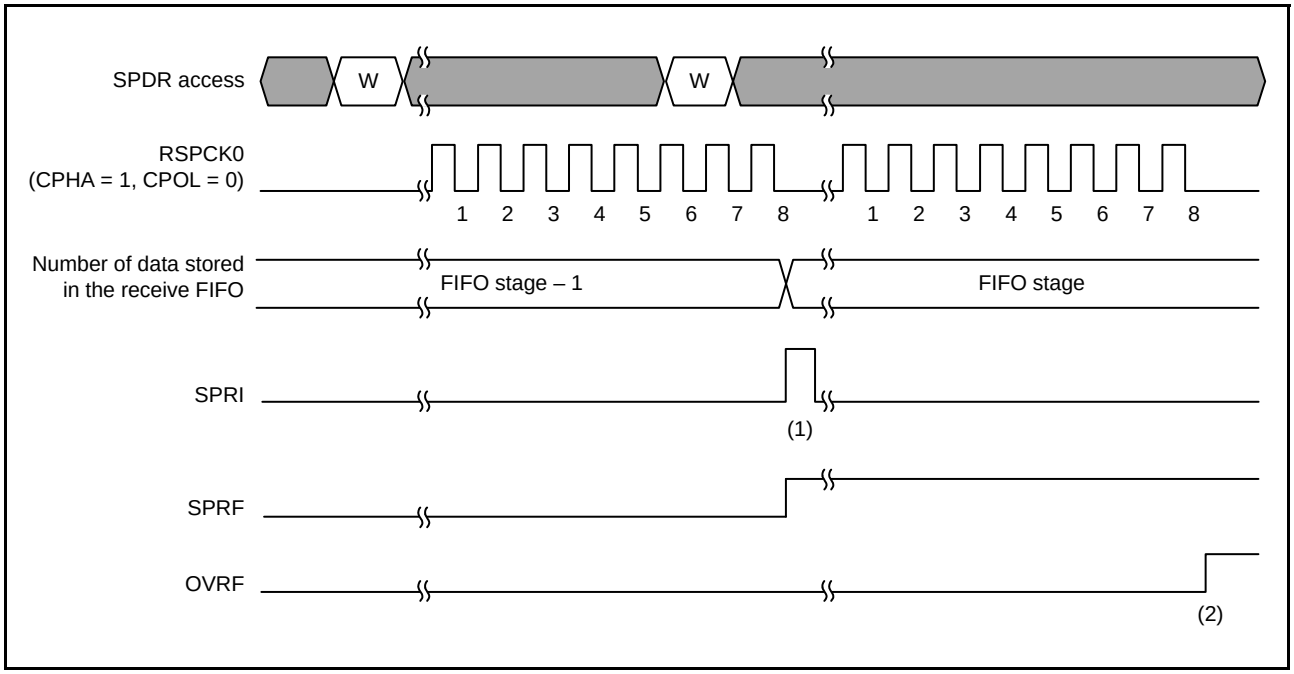


Figure 41.30 Operation Example of SPCR.CMMD[1:0] = 00b

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 while the number of data stored in the SPDR receive buffer matches the number of frames set in SPFCR.RTRG[1:0] bits, the RSPIA generates a receive buffer full interrupt request (SPRI) (sets the SPSR.SPRF flag to 1) and copies the received data in the shift register to the receive buffer.

(2) When a serial transfer ends with data for the number of FIFO stages is stored in the SPDR receive buffer, the RSPIA sets the SPSR.OVRF flag to 1 and discards the received data in the shift register.

When transmit-receive mode (CMMD[1:0] = 00b) is selected, reception occurs simultaneously with transmit operations. As such, the SPRF and OVRF flags in the SPSR register become 1 at the timing described in (1) and (2), respectively, according to the state of the receive buffer.

41.3.6.2 Transmit-Only Mode (SPCR.CMMD[1:0] = 01b)

Figure 41.31 shows an example of operation when the communication mode select bits (SPCR.CMMD[1:0]) are set to 01b. In the example in Figure 41.31, the RSPIA performs an 8-bit serial transfer in which SPFCR.TTRG[1:0] = 00b, SPFCR.RTRG[1:0] = 00b, SPCMDm.CPHA = 1, and SPCMDm.CPOL = 0. The numbers given under the RSPCK0 waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

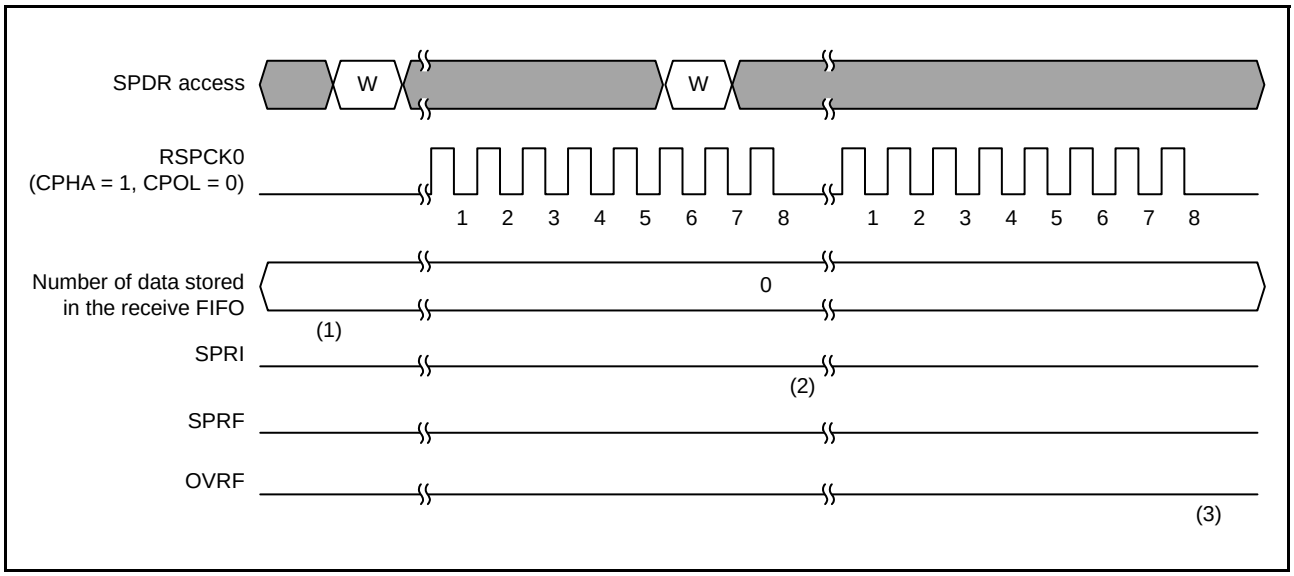


Figure 41.31 Operation Example of SPCR.CMMD[1:0] = 01b

The operation of the flags at timings shown in steps (1) to (3) in the figure is described below.

(1) Make sure there is no data left in the receive buffer and the SPSR.SPRF, OVRF flags are 0 before entering the mode of transmit operations only (SPCR.CMMD[1:0] = 01b).

(2) When a serial transfer ends without received data in the receive FIFO of SPDR, if transmit-only mode is selected (SPCR.CMMD[1:0] = 01b), the SPRF flag remains 0 and the RSPIA does not copy the data from the shift register to the receive buffer.

(3) Since the receive buffer of SPDR does not hold data that was received in the previous serial transfer, even when a serial transfer ends, the SPSR.OVRF flag retains the value of 0, and the data in the shift register is not copied to the receive buffer.

When transmit-only mode (SPCR.CMMD[1:0] = 01b) is selected, the RSPIA transmits data but does not receive data. Therefore, the SPSR.SPRF, OVRF flags remain 0 at the timings of (1) to (3).

41.3.6.3 Receive-Only Mode (SPCR.CMMD[1:0] = 10b)

Figure 41.32 shows an example of operation when the communication mode select bits (SPCR.CMMD[1:0]) are set to 10b. In the example in Figure 41.32, the RSPIA performs an 8-bit serial transfer in which SPFCR.TTRG[1:0] = 00b, SPFCR.RTRG[1:0] = FIFO stage – 1, SPCMDm.CPHA = 1, and SPCMDm.CPOL = 0. The numbers given under the RSPCK0 waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

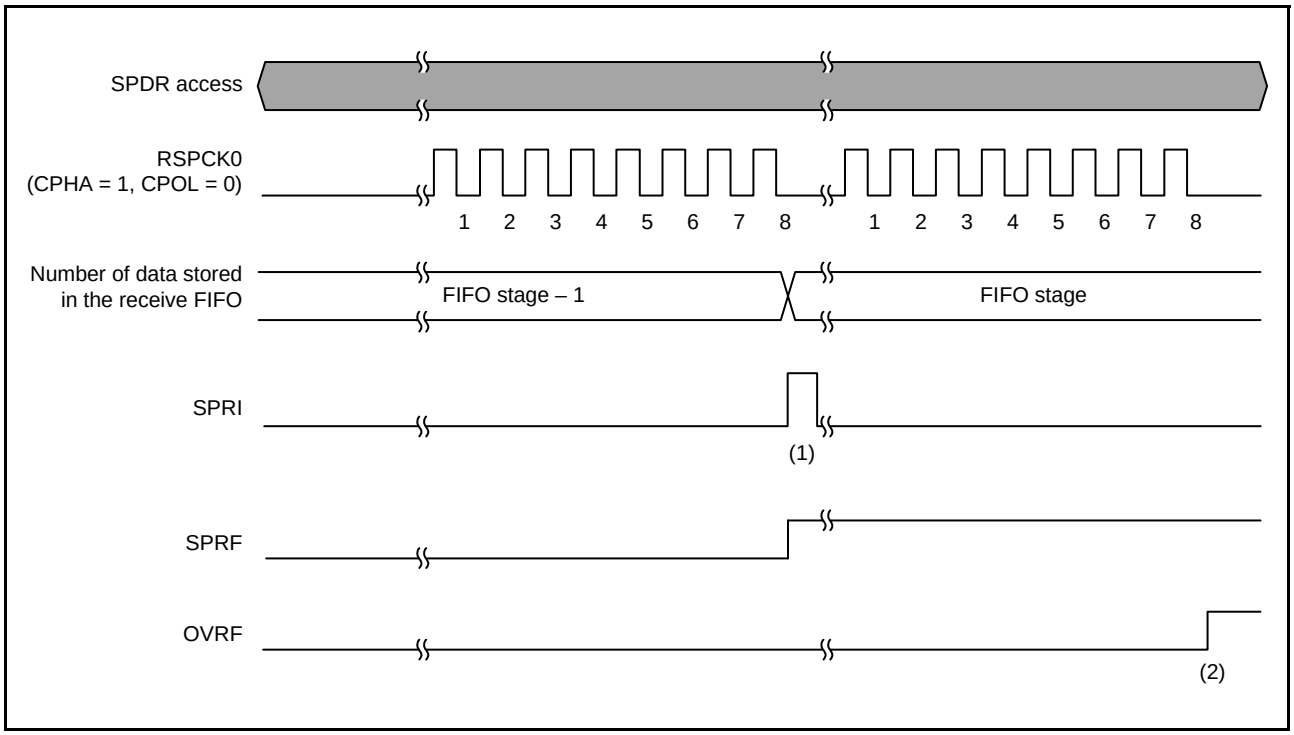


Figure 41.32 Operation Example of SPCR.CMMD[1:0] = 10b

The operation of the flags at timings shown in steps (1) to (2) in the figure is described below.

(1) When a serial transfer ends while the number of data stored in the SPDR receive buffer matches the number of frames set in SPFCR.RTRG[1:0] bits, the RSPIA generates a receive buffer full interrupt request (SPRI) (sets the SPSR.SPRF flag to 1) and copies the received data in the shift register to the receive buffer.

(2) When a serial transfer ends with data for the number of FIFO stages is stored in the SPDR receive buffer, the RSPIA sets the SPSR.OVRF flag to 1 and discards the received data in the shift register.

41.3.7 Transmit Buffer Empty/Receive Buffer Full Interrupts

Figure 41.33 shows an example of operation of the transmit buffer empty interrupt (SPTI) and the receive buffer full interrupt (SPRI). The SPDR register access shown in Figure 41.33 indicate the conditions of access to the SPDR register, where W denotes a write cycle, and R a read cycle. In the examples in Figure 41.33, the RSPIA performs an 8-bit serial transfer in which the SPCR.CMMD[1:0] bits are 00b, SPFCR.TTRG[1:0] bits are 00b, the SPFCR.RTRG[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCK0 waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

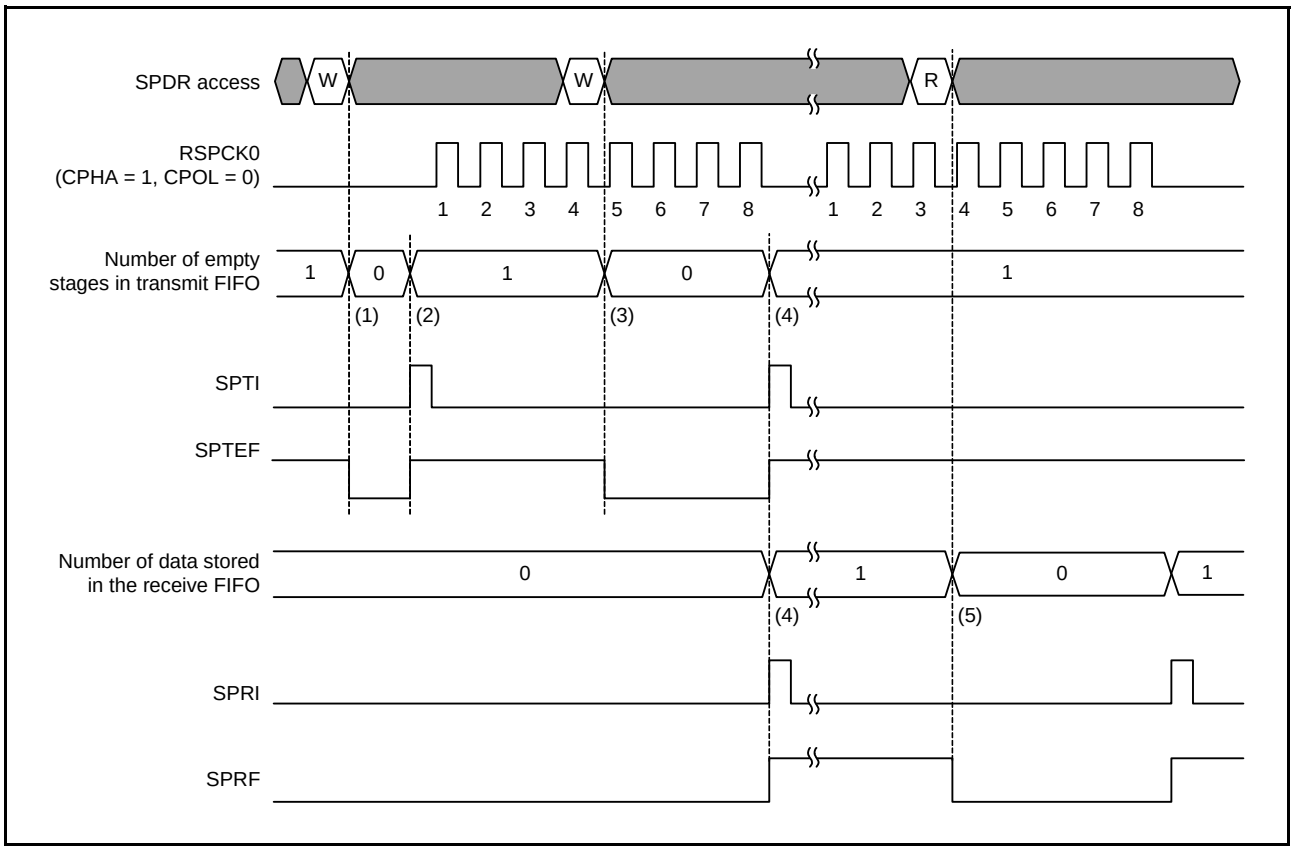


Figure 41.33 Operation Example of SPTI and SPRI Interrupts

The operation of the interrupts at timings shown in steps (1) to (5) in the figure is described below.

(1) If transmit data is written to SPDR when the next transmit data is not set in the transmit FIFO, the RSPIA writes data to the transmit buffer. When transmit data is written to SPDR in one processing routine using DTC/DMAC, the SPSR.SPTEF flag becomes 0 at the last access.

(2) If the shift register is empty, the RSPIA copies the data from the transmit buffer to the shift register. If the number of empty stage of the transmit FIFO is greater than the threshold value, the RSPIA generates a transmit buffer empty interrupt request (SPTI) and sets the SPSR.SPTEF flag to 1. How a serial transfer is started depends on the mode of the RSPIA. For details, refer to section 41.3.13, SPI Operation, and section 41.3.14, Clock Synchronous Operation.

(3) When transmit data is written to SPDR in the transmit buffer empty interrupt routine or in the transmit buffer empty detecting process by polling the SPTEF flag, the data is transferred to the transmit buffer. When transmit data is written to SPDR in one processing routine using DTC/DMAC, the SPSR.SPTEF flag becomes 0 at the last access. Because the data being transmitted is stored in the shift register, the RSPIA does not copy the data from the transmit buffer to the shift register.

(4) If the serial transfer ends (data sampling clock edge in the final bit is detected) when the data stored in the receive FIFO of SPDR < the number of FIFO stages, the RSPIA copies the receive data from the shift register to the receive buffer. At this time, if the data stored in the receive FIFO is greater than the threshold value, the RSPIA generates a

receive buffer full interrupt request (SPRI), and sets the SPSR.SPRF flag to 1. Since the shift register becomes empty upon completion of the serial transfer, when the next transmit data had been set in the transmit FIFO before the serial transfer ended, the RSPIA sets the SPSR.SPTEF flag to 1 and copies the data from the transmit buffer to the shift register. Even when received data is not copied from the shift register to the receive buffer in an overrun error status, upon completion of the serial transfer, the RSPIA determines that the shift register is empty, thus data transfer from the transmit buffer to the shift register is enabled.

(5) When SPDR is read in the receive buffer full interrupt (SPRI) routine or in the receive buffer full detecting process by polling the SPRF flag, the receive data can be read. If the received data is read from SPDR in one processing routine using DTC/DMAC, the SPRF flag becomes 0 at the last access.

If transmit data is written to SPDR while no empty stages in the transmit FIFO, the RSPIA does not update the data in the transmit buffer. Transmit data should be written to SPDR in the transmit buffer empty interrupt request routine or in the transmit buffer empty detecting process by polling the SPTEF flag.

When setting the SPCR.SPE bit to 0 (RSPI function disabled), if the SPCR.SPTIE is 1, a transmit buffer empty interrupt request will occur. However, transmit buffer empty interrupt is inhibited by disabling transmit buffer empty interrupt (the SPTIE bit is 0) at the same time as SPE bit setting.

When serial transfer ends while data is stored in the receive FIFO for the number of FIFO stages, the RSPIA does not copy data from the shift register to the receive buffer, and detects an overrun error (refer to section 41.3.10, Error Detection). To prevent a receive data overrun error, read the received data before the next serial transfer ends.

Transmit buffer empty interrupt and receive buffer full interrupt or the corresponding IRn.IR flags (where n is the interrupt vector number) in the ICU can be used to confirm the states of the transmit and receive buffers. The status of the transmit and receive buffers can be also confirmed by the SPTEF and SPRF flags.

41.3.8 Idle Interrupts

When the SPSSR.SPCP[2:0] bits becomes 000b, if the next transmit data is not set, the SPSR.IDLNF flag is set to 0 and an idle interrupt request is generated. An interrupt request is also made by setting the SPCR.SPE bit to 0.

[Motorola SPI]

Figure 41.34 shows an example of idle interrupt operation during normal operation.

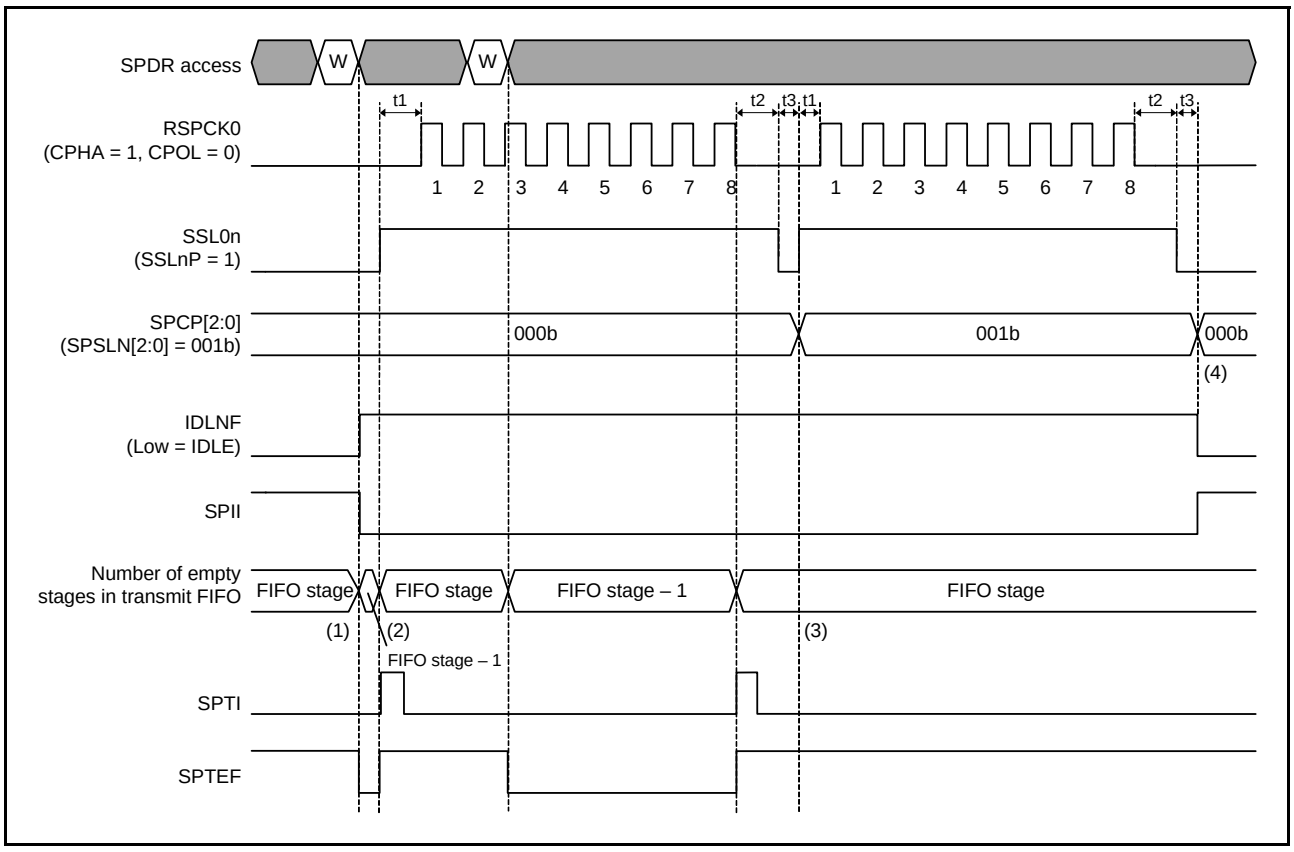


Figure 41.34 Operation Example of Idle Interrupts (Master Mode, Motorola SPI)

(1) At the start of transmission, if the next transmit data is not set in the transmission buffer, the SPSR.IDLNF flag becomes 0 (IDLE). Writing transmit data sets the SPSR.IDLNF flag to 1 (BUSY). When the SPCR.SPIIE bit is set to 1 before transmit data is written, interrupt processing is required before transmission start. For this reason, set the SPCR.SPIIE bit to 0 before starting transmission.

(2) After transmission has started, the SPSR.IDLNF flag remains 1 (BUSY) regardless of the transmit buffer state.

(3) The SPSSR.SPCP[2:0] bits change the command to the next one at the end of t3 cycle. When the next command is not 000b, the SPSR.IDLNF flag remains unchanged even when the next transmit data has not been written.

(4) The SPSR.IDLNF flag becomes 0 (IDLE) at the end of t3 cycle because the next command is 000b and the next transmit data is not present. When the SPCR.SPIIE bit is 1, an SPII interrupt is output.

[TI SSP]

Figure 41.35 shows an example of idle interrupt operation during normal operation.

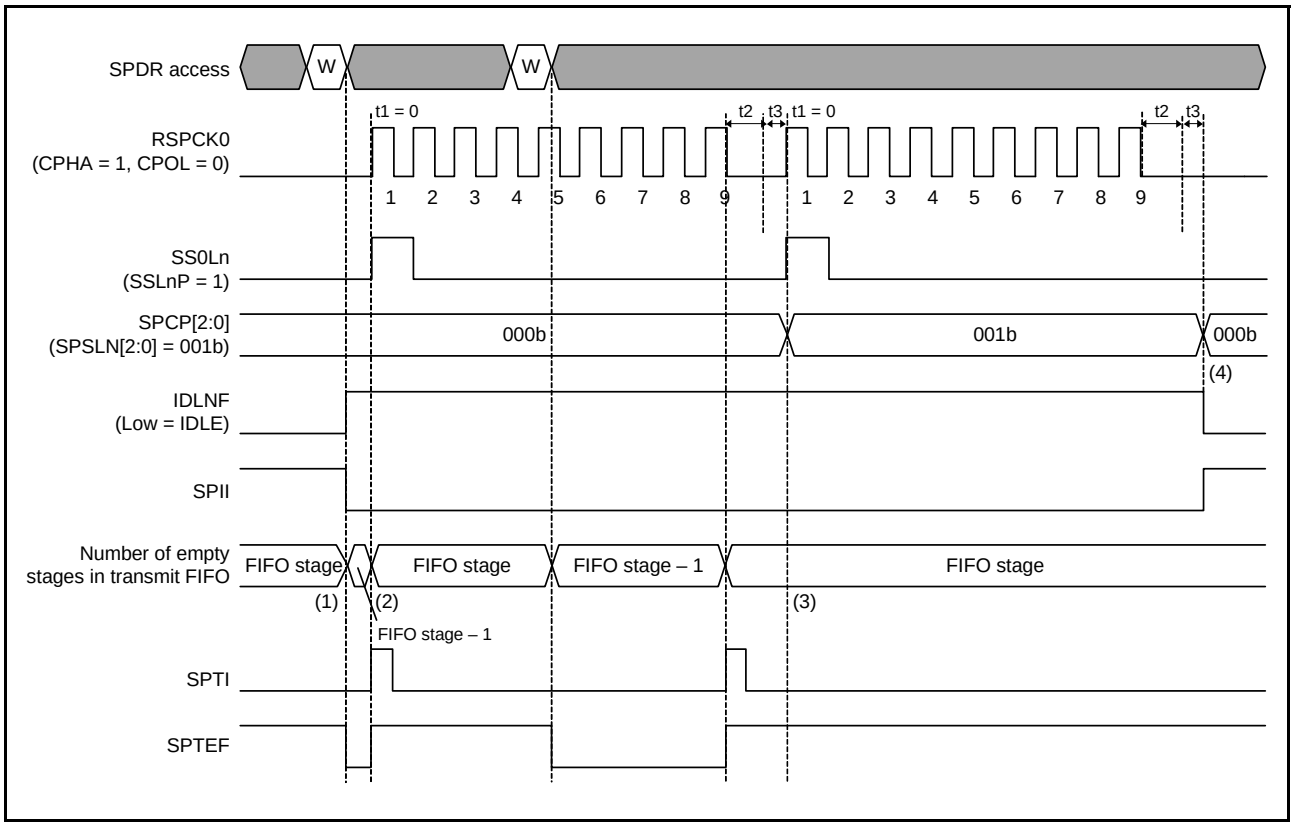


Figure 41.35 Operation Example of Idle Interrupts (Master Mode, TI SSP)

(1) At the start of transmission, if the next transmit data is not set in the transmission buffer, the SPSR.IDLNF flag becomes 0 (IDLE). Writing transmit data sets the SPSR.IDLNF flag to 1 (BUSY). When the SPCR.SPIIE bit is set to 1 before transmit data is written, interrupt processing is required before transmission start. For this reason, set the SPCR.SPIIE bit to 0 before starting transmission.

(2) After transmission has started, the SPSR.IDLNF flag remains 1 (BUSY) regardless of the transmit buffer state.

(3) The SPSSR.SPCP[2:0] bits change the command to the next one at the end of t3 cycle. When the next command is not 000b, the SPSR.IDLNF flag remains unchanged even when the next transmit data has not been written.

(4) The SPSR.IDLNF flag becomes 0 (IDLE) the end of t3 cycle because the next command is 000b and the next transmit data is not present. When the SPCR.SPIIE bit is 1, an SPII interrupt is output.

An idle interrupt during slave mode operation depends only on the value of the SPCR.SPE bit.

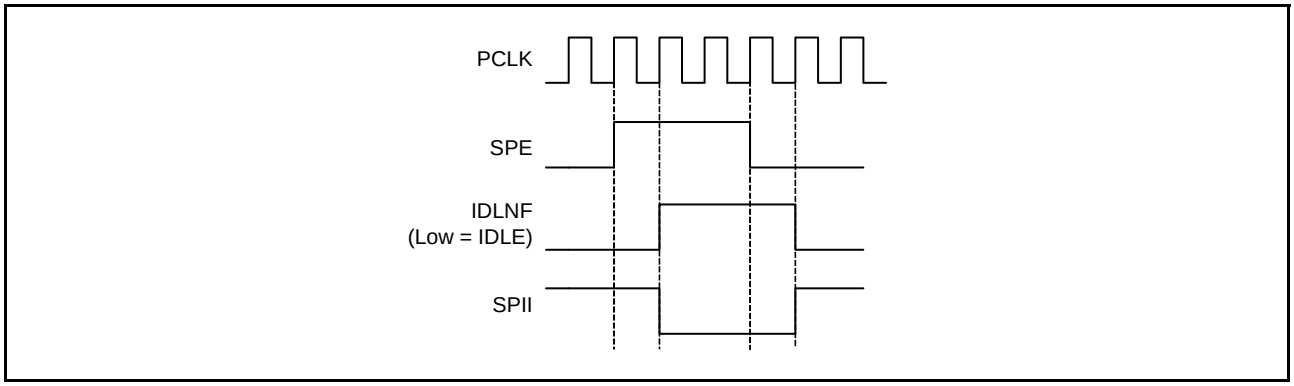


Figure 41.36 Operation Example of Idle Interrupts (Slave Mode)

41.3.9 Communication End Interrupts

41.3.9.1 Transmit-Receive Master Mode or Transmit-Only Master Mode

Refer to the description of the SPCF Flag (Communication End Flag) in section 41.2.16, RSPI Status Register (SPSR) for setting/clearing conditions of the communication end flag during transmit-receive master mode or transmit-only master mode.

[Motorola SPI]

Figure 41.37 shows an example of the SPCI interrupt operation during transmit-receive master mode/transmit-only master mode.

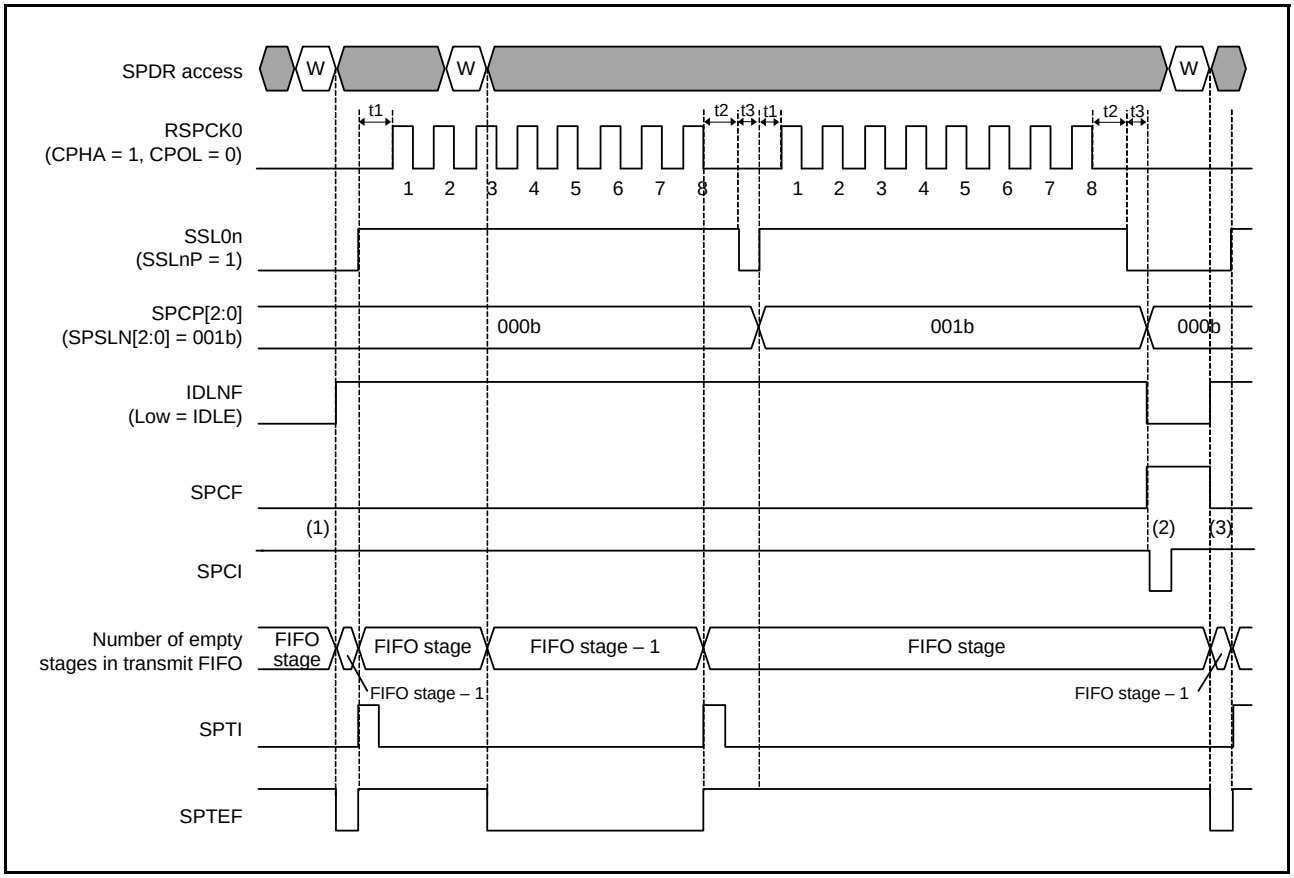


Figure 41.37 Operation Example of SPCI Interrupts (Transmit-Receive Master Mode/Transmit-Only Master Mode, Motorola SPI)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) The SPCF flag becomes 1 (communication end) at the end of t3 cycle, because the next command is 000b and there is no next transmit data. And then the SPCI interrupt outputs with PCLK 1 cycle width if the SPCR.SPCIE bit is 1.

(3) The SPCF flag becomes 0 when the next transmit data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

[TI SSP]

Figure 41.38 shows an example of the SPCI interrupt operation during transmit-receive master mode/transmit-only master mode.

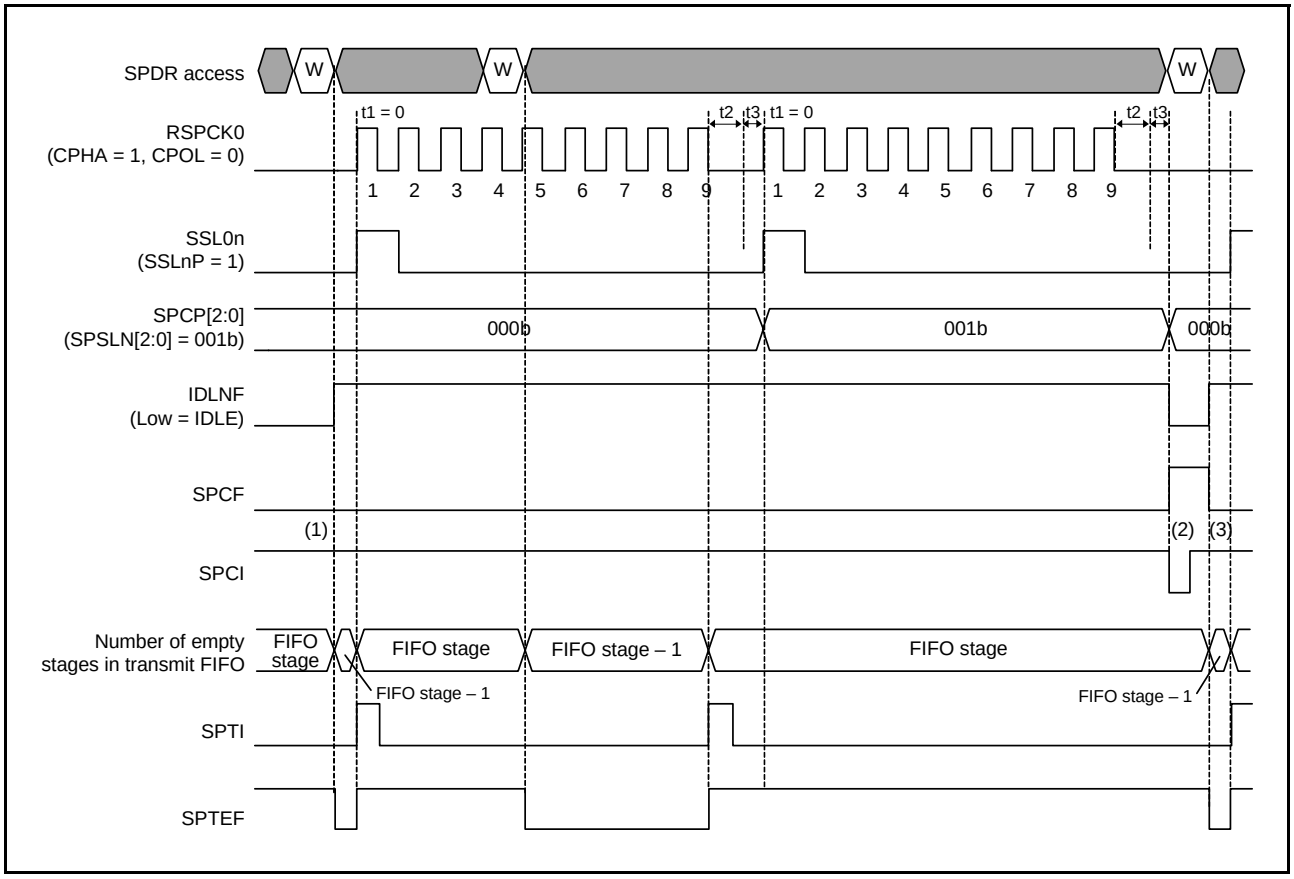


Figure 41.38 Operation Example of SPCI Interrupts (Transmit-Receive Master Mode/Transmit-Only Master Mode, TI SSP)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) The SPCF flag becomes 1 (communication end) at the end of t3 cycle, because the next command is 000b and there is no next transmit data. And then the SPCI interrupt outputs with PCLK 1 cycle width if the SPCR.SPCIE bit is 1.

(3) The SPCF flag becomes 0 when the next transmit data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

41.3.9.2 Receive-Only Master Mode

Refer to the description of the SPCF Flag (Communication End Flag) in section 41.2.16, RSPI Status Register (SPSR) for setting/clearing conditions of the communication end flag during receive-only master mode.

Figure 41.39 shows an example of the SPCI interrupt operation during receive-only master mode when the SPRMCR.RFC[4:0] bits are 00000b.

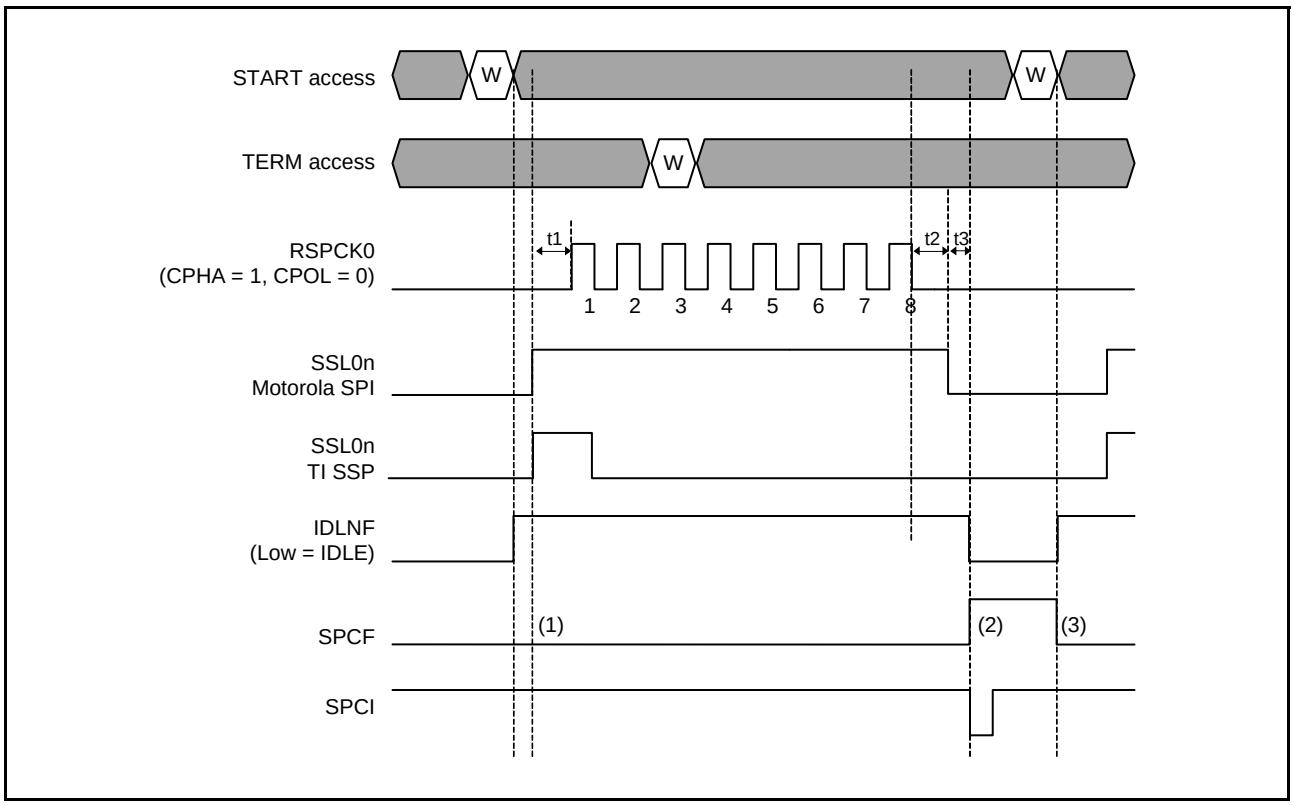


Figure 41.39 Operation Example of SPCI Interrupts (RFC[4:0] = 00000b) (Receive-Only Master Mode)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) When the SPRMCR.TERM bit is set to 1 during communication, the SPCF flag becomes 1 (communication end) at the end of t3 cycle. And then the SPCI interrupt outputs with PCLK 1 cycle width if the SPCR.SPCIE bit is 1.

(3) The SPCF flag becomes 0 when 1 is written to the SPRMCR.START bit. Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

Figure 41.40 shows an example of the SPCI interrupt operation during receive-only master mode when the SPRMCR.RFC[4:0] bits are not 00000b.

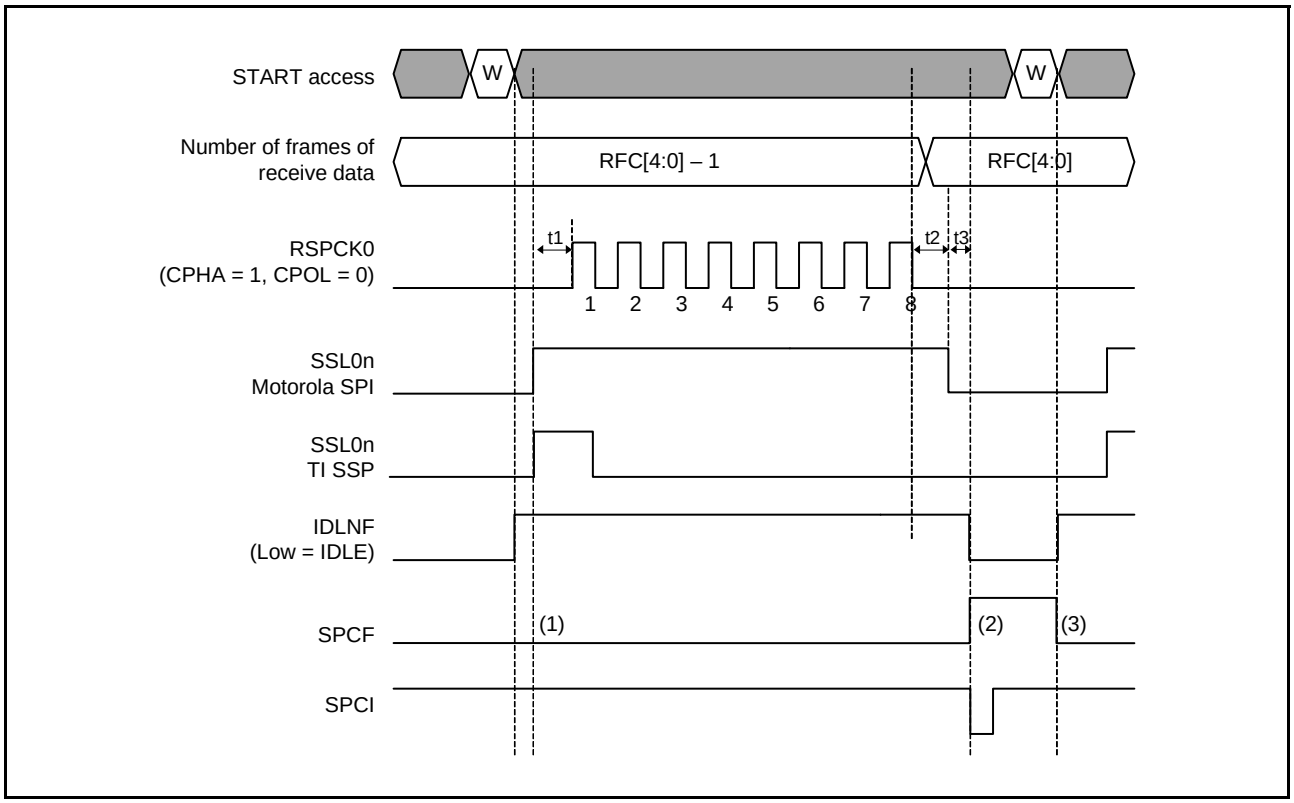


Figure 41.40 Operation Example of SPCI Interrupts (RFC[4:0] ≠ 00000b) (Receive-Only Master Mode)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) The SPCF flag becomes 1 (communication end) at the end of t3 cycle after receiving the number of frames set by the SPRMCR.RFC[4:0] bits. And then the SPCI interrupt outputs with PCLK 1 cycle width if the SPCR.SPCIE bit is 1.

(3) The SPCF flag becomes 0 when 1 is written to the SPRMCR.START bit. Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

In slave mode operation, the output timing of the SPCI interrupt is deferent due to the value of the SPCR.SPMS bit (RSPI mode select bit). And the clear timing of the communication end interrupt is deferent due to the communication mode (transmit-receive, transmit-only, or receive-only).

41.3.9.3 Transmit-Receive Slave Mode or Transmit-Only Slave Mode on SPI Operation (4-Wire Method)

Refer to the description of the SPCF Flag (Communication End Flag) in section 41.2.16, RSPI Status Register (SPSR) for setting/clearing conditions of the communication end flag during transmit-receive slave mode or transmit-only slave mode on SPI operation (4-wire method).

[Motorola SPI]

Figure 41.41 shows an example of the SPCI interrupt operation during transmit-receive slave mode/transmit-only slave mode on SPI operation.

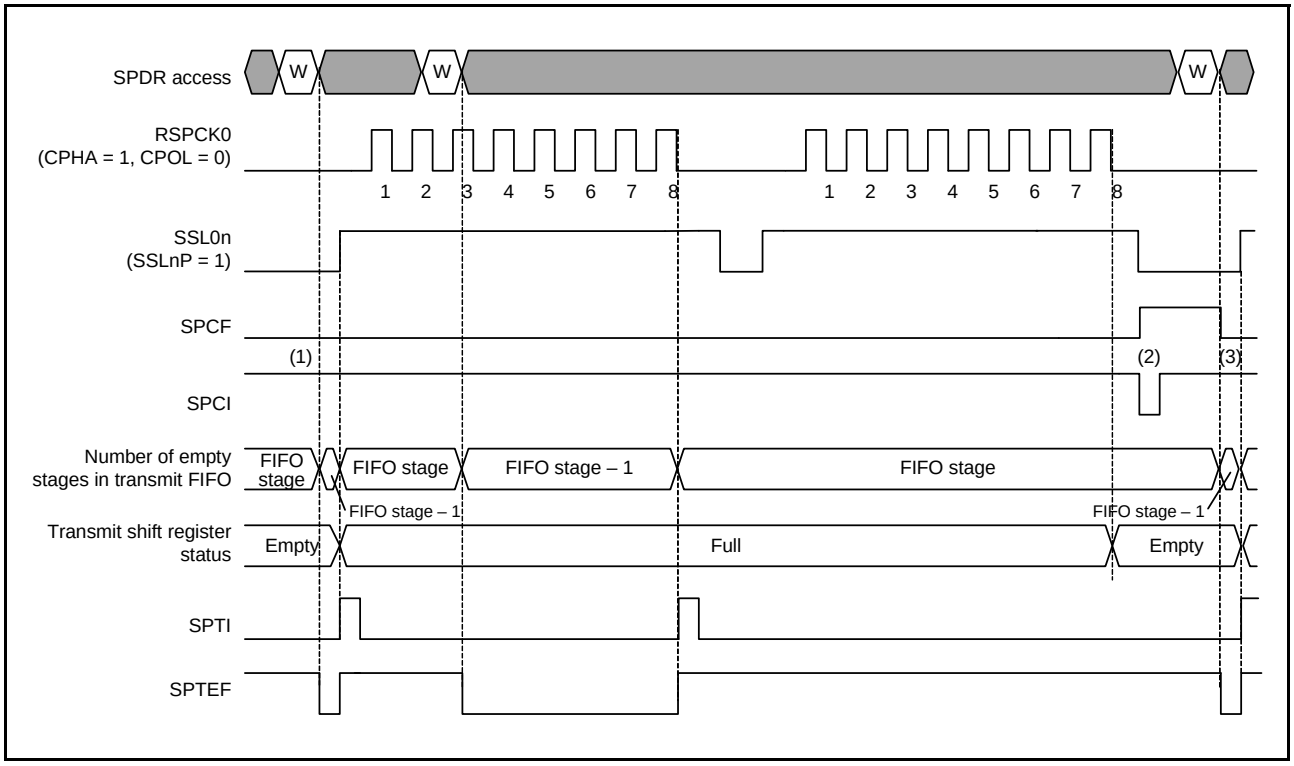


Figure 41.41 Operation Example of SPCI Interrupts (Transmit-Receive Slave Mode/Transmit-Only Slave Mode, Motorola SPI)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) The SPCF flag becomes 1 (communication end) at the timing of SSL0n negation, when the next transmit data is not set in the transmit FIFO and the transmit shift register is empty. And then the SPCI interrupt outputs with PCLK 1 cycle width if the SPCR.SPCIE bit is 1.

(3) The SPCF flag becomes 0 when the next transmit data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

[TI SSP]

Figure 41.42 shows an example of the SPCI interrupt operation during transmit-receive slave mode/transmit-only slave mode on SPI operation.

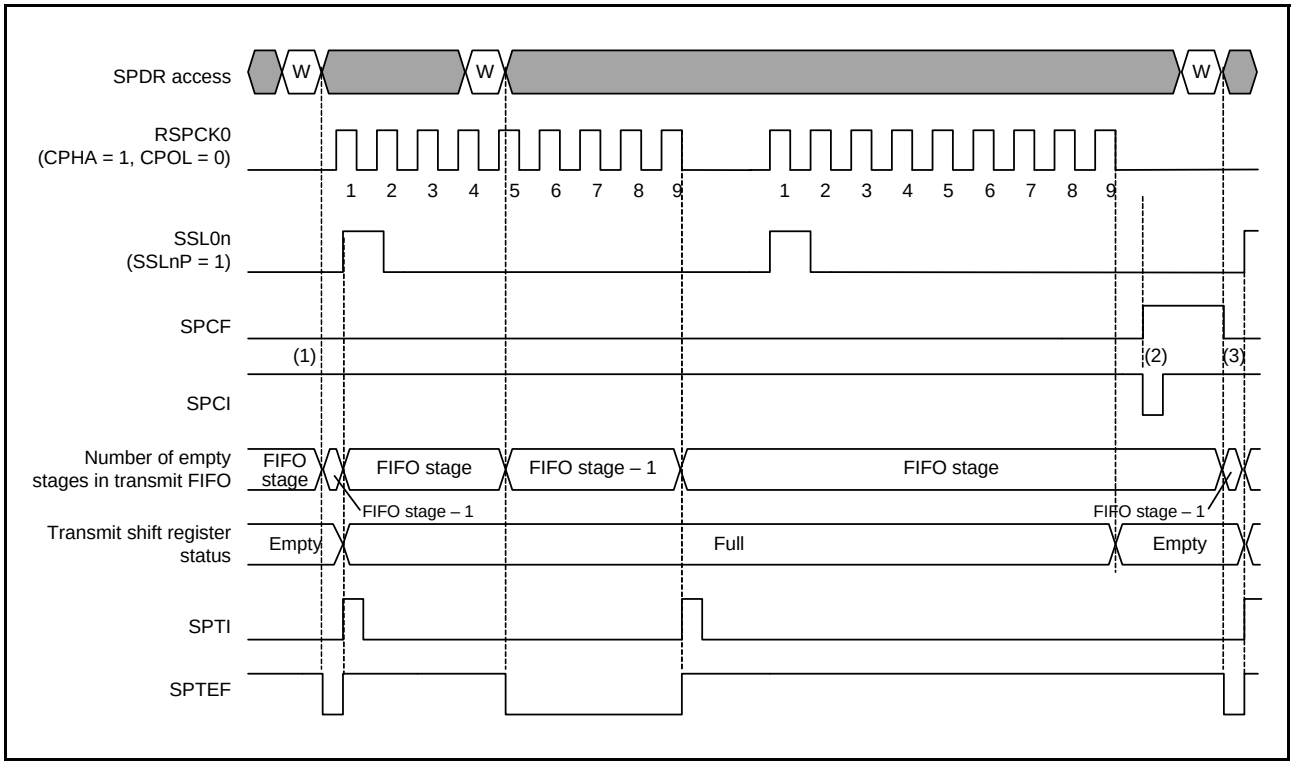


Figure 41.42 Operation Example of SPCI Interrupts (Transmit-Receive Slave Mode/Transmit-Only Slave Mode, TI SSP)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) The SPCF flag becomes 1 (communication end) when the next transmit data is not set in the transmit FIFO, the transmit shift register is empty, and SSL negation delay is completed. And then the SPCI interrupt outputs with PCLK 1 cycle width if the SPCR.SPCIE bit is 1.

(3) The SPCF flag becomes 0 when the next transmit data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

41.3.9.4 Receive-Only Slave Mode on SPI Operation (4-Wire Method)

Refer to the description of the SPCF Flag (Communication End Flag) in section 41.2.16, RSPI Status Register (SPSR) for setting/clearing conditions of the communication end flag during receive-only slave mode on SPI operation (4-wire method).

[Motorola SPI]

Figure 41.43 shows an example of the SPCI interrupt operation during receive-only slave mode on SPI operation.

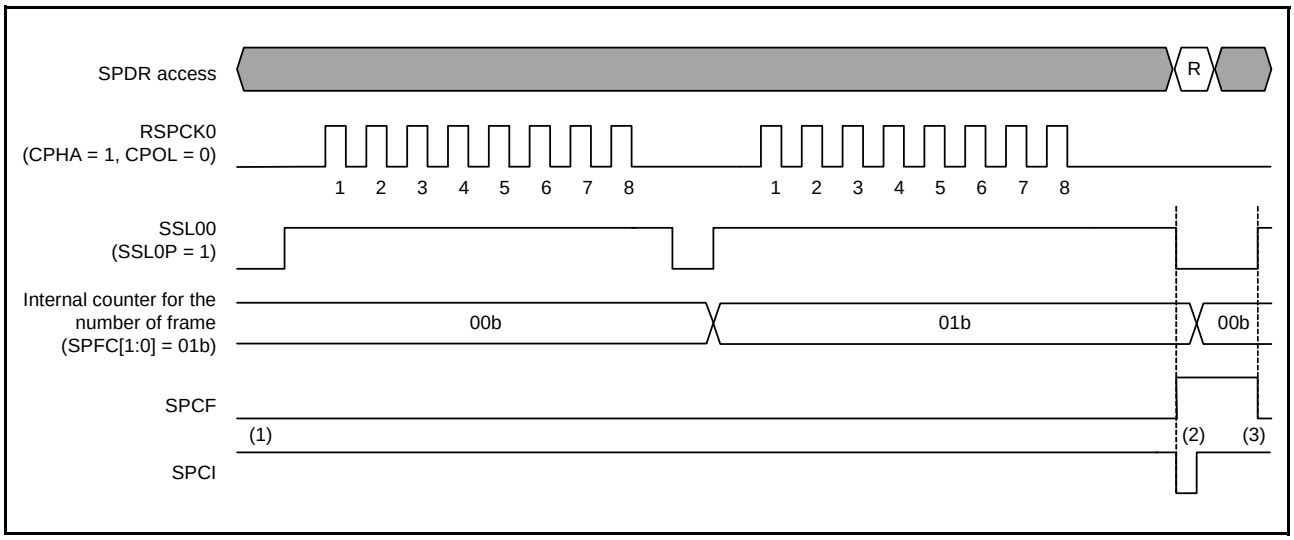


Figure 41.43 Operation Example of SPCI Interrupts (Receive-Only Slave Mode, Motorola SPI)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) After the frames of the value set in the SPDCR.SPFC[1:0] bits are stored in the receive buffer, the SPCF flag becomes 1 (communication end) at the timing of SSL00 negation. And then the SPCI interrupt outputs with PCLK 1 cycle width if the SPCIE bit is 1.

(3) The SPCF flag becomes 0 at the SSL00 assertion when the next transmission start. Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

[TI SSP]

Figure 41.44 shows an example of the SPCI interrupt operation during receive-only slave mode on SPI operation.

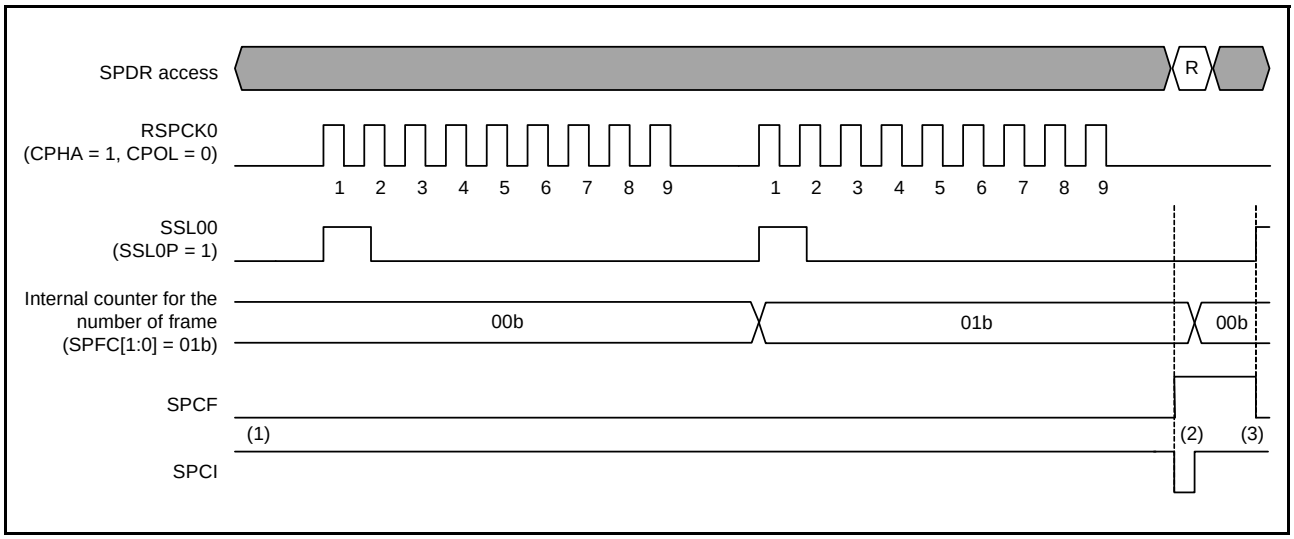


Figure 41.44 Operation Example of SPCI Interrupts (Receive-Only Slave Mode, TI SSP)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) After the frames of the value set in the SPDCR.SPFC[1:0] bits are stored in the receive buffer, the SPCF flag becomes 1 (communication end) at the timing when SSL00 negation delay is completed. And then the SPCI interrupt outputs with PCLK 1 cycle width if the SPCIE bit is 1.

(3) The SPCF flag becomes 0 at the SSL00 assertion when the next transmission start. Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

41.3.9.5 Transmit-Receive Slave Mode or Transmit-Only Slave Mode on Clock Synchronous Operation (3-Wire Method)

Refer to the description of the SPCF Flag (Communication End Flag) in section 41.2.16, RSPI Status Register (SPSR) for setting/clearing conditions of the communication end flag during transmit-receive slave mode or transmit-only slave mode on clock synchronous operation (3-wire method).

Figure 41.45 shows an example of the SPCI interrupt operation during transmit-receive slave mode/transmit-only slave mode on clock synchronous operation.

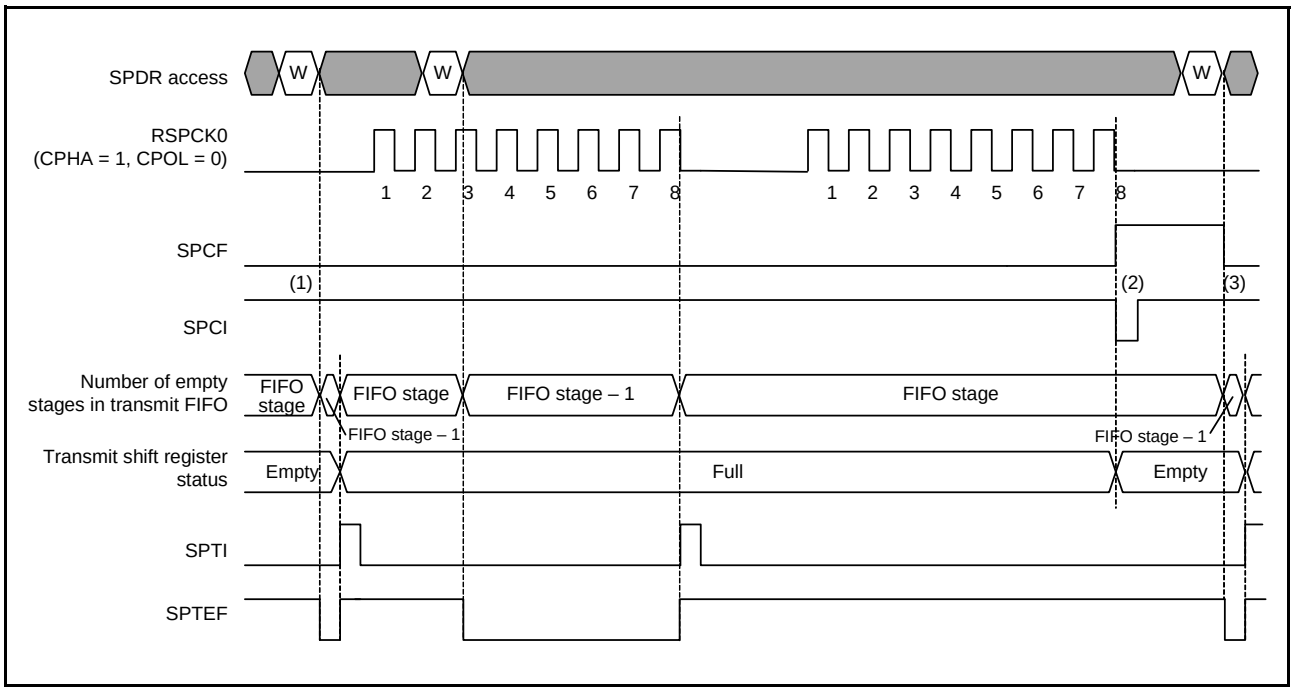


Figure 41.45 Operation Example of SPCI Interrupts (Transmit-Receive Slave Mode/Transmit-Only Slave Mode, Clock Synchronous Operation)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) The SPCF flag becomes 1 (communication end) at the sampling timing of the final data bit of RSPCK0, when the next transmit data is not set in the transmit buffer and the transmit shift register is empty. And then the SPCI interrupt outputs with PCLK 1 cycle width if the SPCR.SPCIE bit is 1.

(3) The SPCF flag becomes 0 when the next transmit data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

41.3.9.6 Receive-Only Slave Mode on Clock Synchronous Operation (3-Wire Method)

Refer to the description of the SPCF Flag (Communication End Flag) in section 41.2.16, RSPI Status Register (SPSR) for setting/clearing conditions of the communication end flag during receive-only slave mode on clock synchronous operation (3-wire method).

Figure 41.46 shows an example of the SPCI interrupt operation during receive-only slave mode on clock synchronous operation.

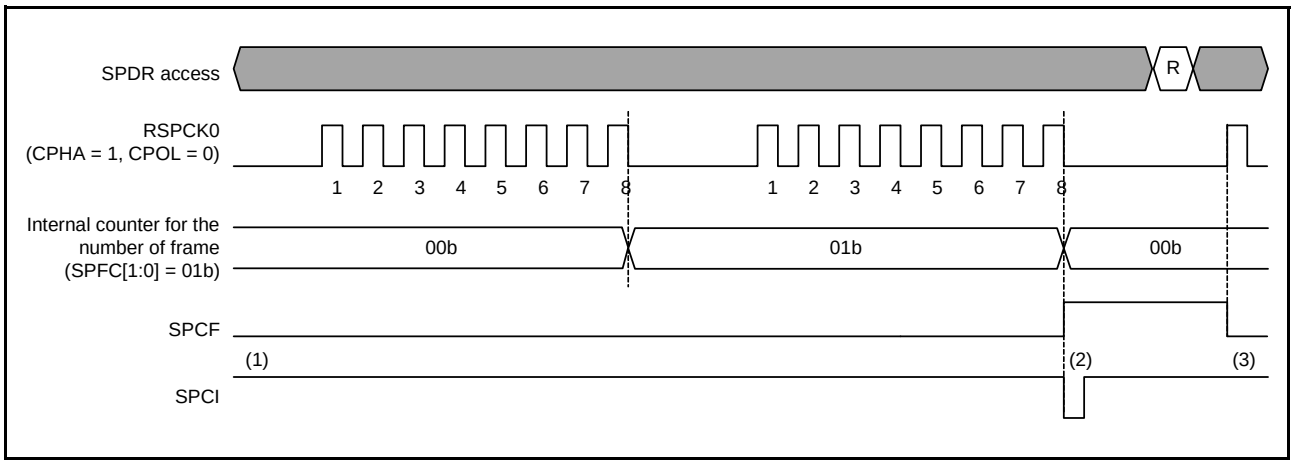


Figure 41.46 Operation Example of SPCI Interrupts (Receive-Only Slave Mode, Clock Synchronous Operation)

(1) The SPCF flag is 0 and the level of the SPCI interrupt is 1 before starting communication. And these have kept during communication.

(2) The SPCF flag becomes 1 (communication end) at the sampling timing of the final data bit of RSPCK0, when the final frame of the value set in the SPDCR.SPFC[1:0] bits is received. And then the SPCI interrupt outputs with PCLK 1 cycle width if the SPCR.SPCIE bit is 1.

(3) The SPCF flag becomes 0 at the first edge of RSPCK0 for the next transmission. Or when 1 is written to the SPSCLR.SPCFC bit, then the SPCF flag becomes 0.

41.3.10 Error Detection

In the normal serial transfer of the RSPIA, the data written to the transmit buffer of SPDR is transmitted, and the received data can be read from the receive buffer of SPDR. If access is made to SPDR, depending on the status of the transmit/receive buffer or the status of the RSPIA 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 RSPIA detects the event as an underrun error, overrun error, parity error, or mode fault error. Table 41.8 lists the relationship between non-normal transfer operations and the RSPIA's error detection function.

Table 41.8 Relationship between Non-Normal Transfer Operations and Error Detection Function of RSPIA

	Occurrence Condition	RSPI Operation	Error Detection
1	The SPDR register is written while no empty stages in the transmit FIFO.	• The contents of the transmit buffer are kept. • Missing write data.	None
2	The SPDR register is read while no empty stages in the receive FIFO.	If reception has completed, then the received data is output to the bus. Otherwise, data received previously is output instead.	None
3	Serial transfer is started in transmit-receive slave mode or transmit-only slave mode when transmit data is not ready to output.	• Serial transfer is suspended • Transmit/receive data is missing • The MISO0 output signal is disabled • RSPI function is disabled	Underrun error Refer to section 41.3.10.4, Underrun Error for underrun error.
4	Serial transfer terminates when data is stored in the receive FIFO with no empty FIFO stage.	• The contents of the receive FIFO are kept • Missing receive data	Overrun error Refer to section 41.3.10.1, Overrun Error for overrun error.
5	An incorrect parity bit is received with the parity function enabled in the transmit-receive mode or receive-only mode	The parity error flag is asserted.	Parity error Refer to section 41.3.10.2, Parity Error for parity error.
6	The SSL00 input signal is asserted when the serial transfer is idle in multi-master mode.	• Driving of the RSPCK0, MOSI0, SSL01 to SSL03 output signals is stopped. • RSPI function is disabled.	Mode fault error Refer to section 41.3.10.3, Mode Fault Error for mode fault error.
7	The SSL00 input signal is asserted during serial transfer in multi-master mode.	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the RSPCK0, MOSI0, SSL01 to SSL03 output signals is stopped. • RSPI function is disabled.	
8	The SSL00 input signal is negated during serial transfer in slave mode. [Motorola SPI]	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the MISO0 output signal is stopped. • RSPI function is disabled.	
9	The SSL00 input signal is asserted during serial transfer in slave mode. [TI SSP]	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the MISO0 output signal is stopped. • RSPI function is disabled.	
10	After data is stored in the receive FIFO with RDRIS = 1, the number of stored data is less than the threshold value and no receive data is written for the set value of the SPDRCSR register.	• The receive data ready flag is asserted.	Receive data ready Refer to section 41.3.11, Received Data Ready Detection for receive data ready

On operation 1 in Table 41.8, the RSPIA does not detect an error. To prevent data omission during the writing to SPDR, the SPDR register should be written when a transmit buffer empty interrupt request occurs or while the SPSR.SPTEF flag is 1.

Likewise, the RSPIA does not detect an error on operation 2. To prevent extraneous data from being read, the SPDR register should be read when a receive buffer full interrupt request occurs or while the SPSR.SPRF flag is 1.

For the transmit and receive interrupts, refer to section 41.3.7, Transmit Buffer Empty/Receive Buffer Full Interrupts.

41.3.10.1 Overrun Error

If a serial transfer ends while no empty stages in the receive FIFO, the RSPIA detects an overrun error, and sets the SPSR.OVRF flag to 1. When the OVRF flag is 1, the RSPIA 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 SPSR.OVRF flag to 0, issue a system reset or write 1 to the SPSRC.OVRFC bit.

Figure 41.47 shows an example of operations of the SPSR.SPRF and OVRF flags. The SPSCLR and SPDR accesses shown in Figure 41.47 indicate the condition of accesses to SPSCLR and SPDR, respectively, where W denotes a write cycle, and R a read cycle. In the example in Figure 41.47, the RSPIA 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 RSPCK0 waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

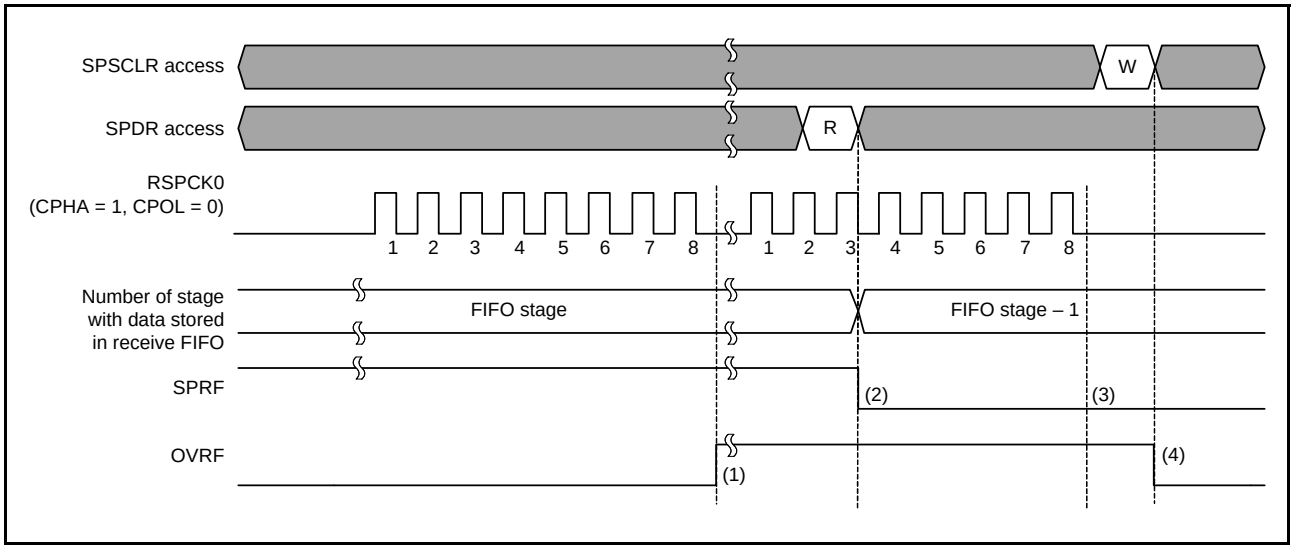


Figure 41.47 Operation Example of SPRF and OVRF Flags

The operation of the flags at the timing shown in steps (1) to (4) in the figure is described below.

(1) If a serial transfer terminates while no empty stages in the receive FIFO, the RSPIA detects an overrun error, and sets the OVRF flag to 1. The RSPIA 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 RSPIA copies the pointer value to SPCMDm register to the SPSSR.SPECM[2:0] bits.

(2) When SPDR is read, the RSPIA outputs the data in the receive buffer. At this time the SPRF flag becomes 0 at the last access when the received data is read from the SPDR register in one processing routine using DTC/DMAC. Even if the receive buffer becomes empty, the OVRF flag does not become 0.

(3) If the serial transfer ends with the OVRF flag being 1 (an overrun error occurs), the RSPIA does not copy the data in the shift register to the receive buffer (the SPRF flag remains 0). A receive buffer full interrupt is not generated. Even if the SPPE bit is 1, parity errors are not detected. When in master mode, the RSPIA does not update the SPSSR.SPECM[2:0] bits. When in an overrun error state and the RSPIA does not copy the received data from the shift register to the receive buffer, upon termination of the serial transfer, the RSPIA determines that the shift register is empty; in this manner, data transfer from the transmit buffer to the shift register is enabled.

(4) When 1 is written to the SPSCLR.OVRFC bit, the OVRF flag is set to 0.

The occurrence of an overrun can be checked either by reading the SPSR register or by using an error interrupt and reading the SPSR register. When executing a serial transfer without using an error interrupt, measures should be taken to ensure the early detection of overrun errors, such as reading the SPSR register immediately after SPDR is read. When the RSPIA is used in master mode, the pointer value to the SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits. If an overrun error occurs and the OVRF flag is set to 1, normal reception operations cannot be performed until the OVRF flag is set to 0.

When the RSPCK auto-stop function is enabled in master mode, an overrun error does not occur. Figure 41.48 and Figure 41.49 show the clock stop waveform when a serial transfer continues while no empty stages in the receive FIFO in master mode.

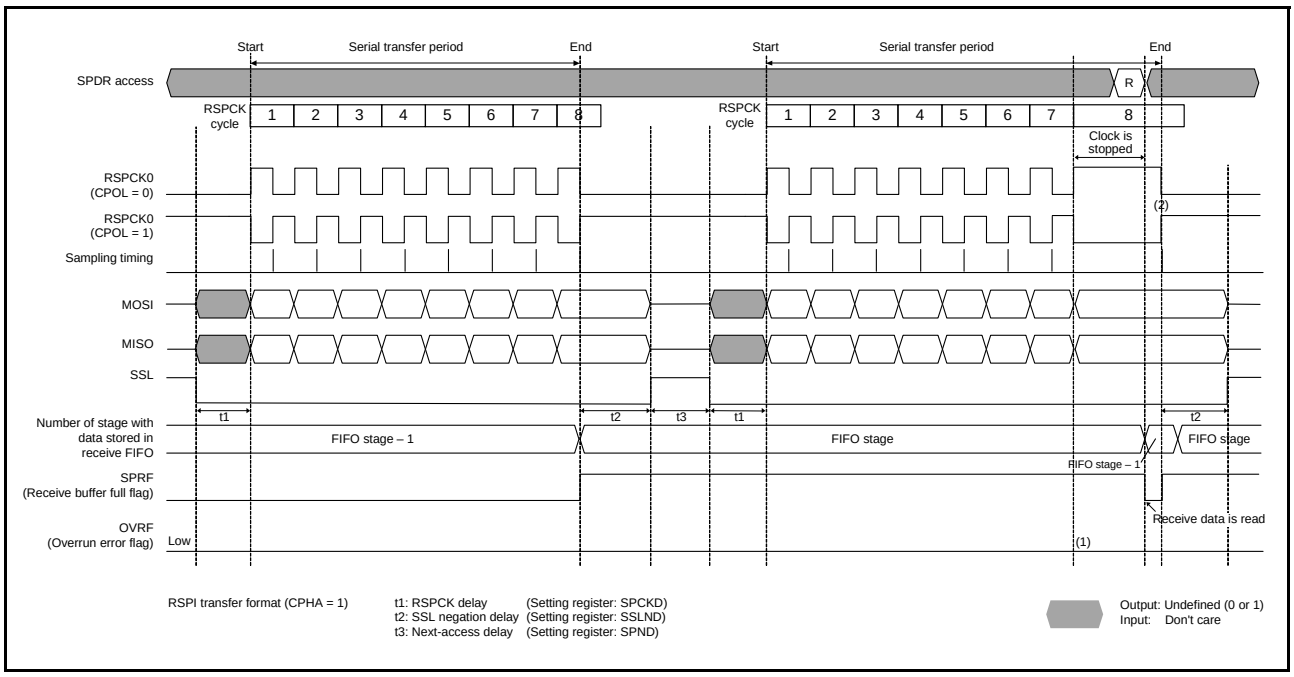


Figure 41.48 Clock Stop Waveform When a Serial Transfer Continues While No Empty Stages in the Receive FIFO in Master Mode (CPHA = 1)

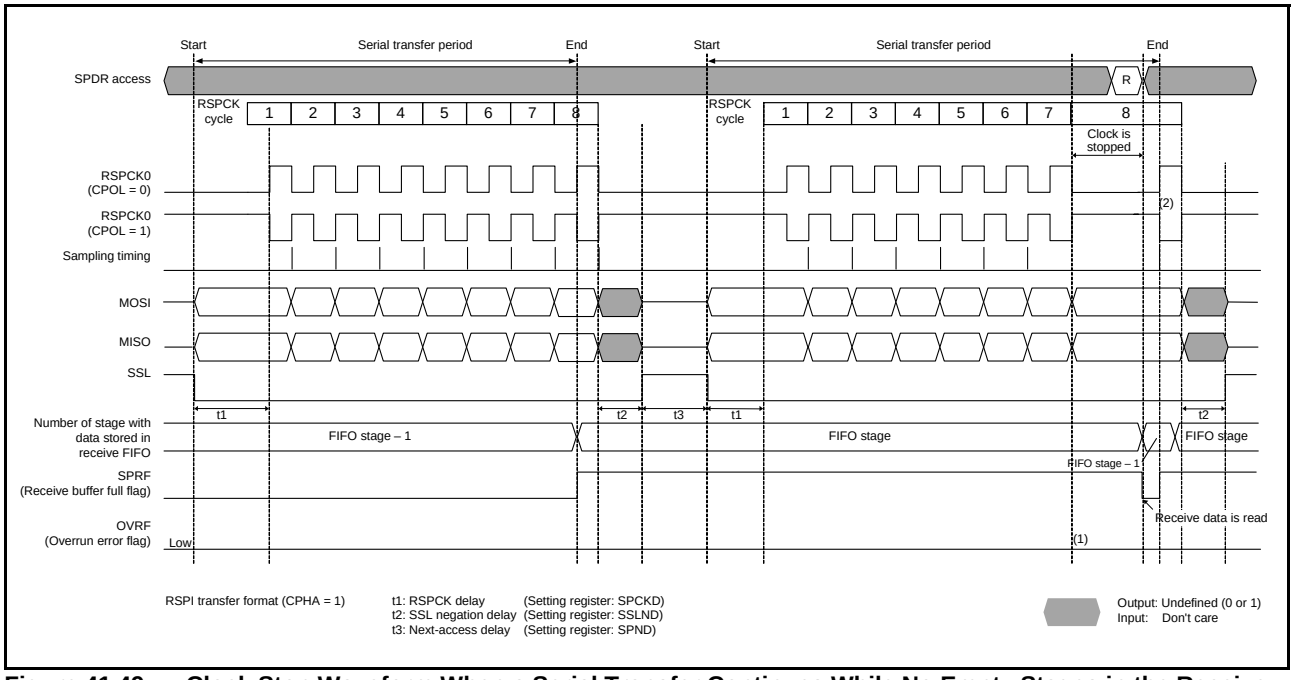


Figure 41.49 Clock Stop Waveform When a Serial Transfer Continues While No Empty Stages in the Receive FIFO in Master Mode (CPHA = 0)

The operation of the flags at the timings shown in steps (1) and (2) in the figure is described below.

(1) When no empty stages in the receive FIFO, an overrun error does not occur because the RSPCK0 is stopped.

(2) If SPDR is read while the clock is stopped, data in the receive buffer can be read. The RSPCK0 restarts after reading the receive buffer.

When the RSPCK auto-stop function is enabled for transfer with no delay of between frames during burst transfer in master mode, an overrun error does not occur. Figure 41.50 and Figure 41.51 show the clock stop waveform when there is no delay between frames during burst transfer and a serial transfer continues while no empty stages in the receive FIFO in master mode.

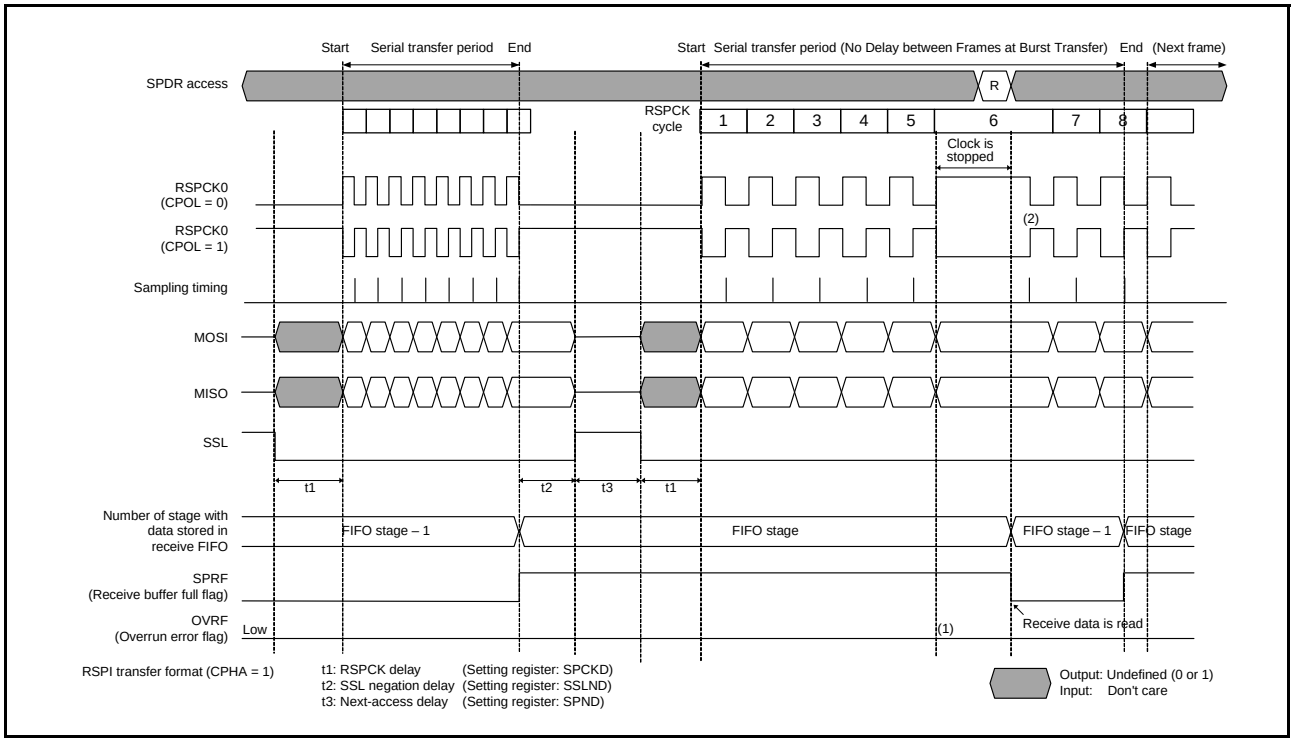


Figure 41.50 Clock Stop Waveform When a Serial Transfer Continues While No Empty Stages in the Receive FIFO in Master Mode (No Delay between Frames at Burst Transfer, CPHA = 1)

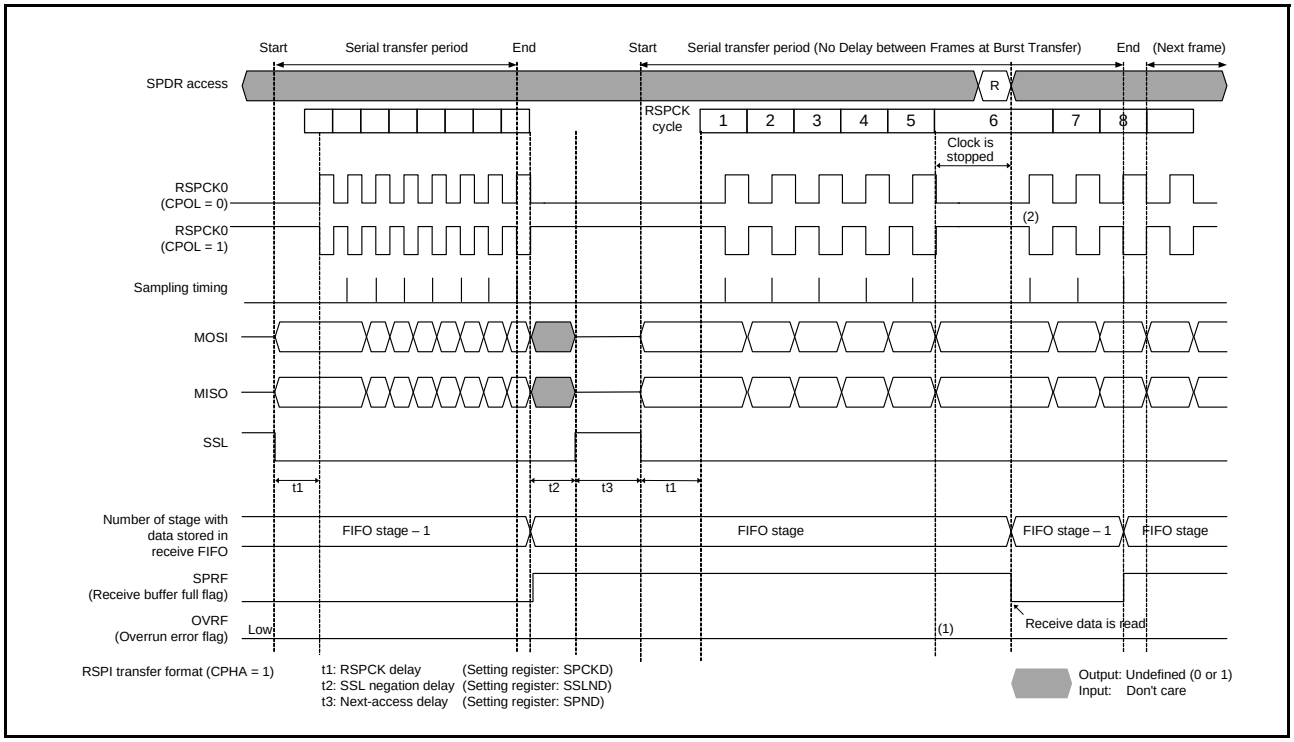


Figure 41.51 Clock Stop Waveform When a Serial Transfer Continues While No Empty Stages in the Receive FIFO in Master Mode (No Delay between Frames at Burst Transfer, CPHA = 0)

The operation of the flags at the timings shown in steps (1) and (2) in the figure is described below.

(1) When no empty stages in the receive FIFO, an overrun error does not occur because the RSPCK0 is stopped.

(2) If SPDR is read while the clock is stopped, data in the receive buffer can be read. The RSPCK0 restarts after reading the receive buffer.

41.3.10.2 Parity Error

If the SPCR.SPPE bit set to 1, when serial transfer in transmit-receive mode or receive-only mode terminates, the RSPIA checks whether there are parity errors. Upon detecting a parity error in the received data, the RSPIA sets the SPSR.PERF flag to 1. Since the RSPIA 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, issue a system reset or write 1 to the SPSCLR.PERFC bit.

Figure 41.52 shows an example of operation of the OVRF and PERF flags. The SPSR access shown in Figure 41.52 indicates the condition of access to SPSR register, where W denotes a write cycle, and R a read cycle. In the example of Figure 41.52, serial communications is performed while the SPCR.SPPE bit is 1. The RSPIA 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 RSPCK0 waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

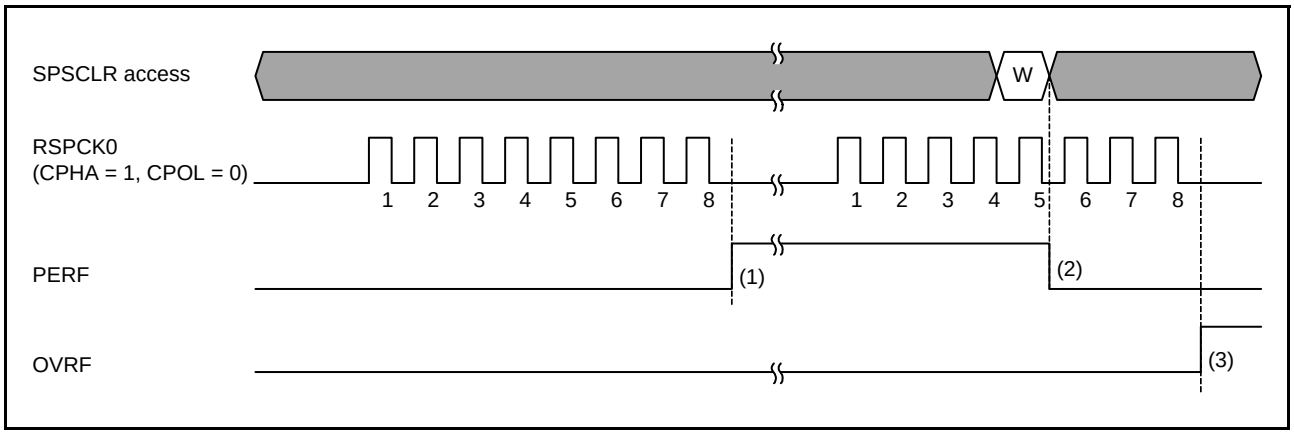


Figure 41.52 Operation Example of PERF Flag

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 RSPIA not detecting an overrun error, the RSPIA copies the data in the shift register to the receive buffer. The RSPIA judges the received data at this timing, and sets the PERF flag to 1 if a parity error is detected. In master mode, the RSPIA copies the pointer value to SPCMDm register to the SPSSR.SPECM[2:0] bits.

(2) When 1 is written to the SPSCLR.PERFC bit, the PERF flag is set to 0.

(3) When the RSPIA detects an overrun error and serial transfer is terminated, the data in the shift register is not copied to the receive buffer. The RSPIA does not perform parity error detection at this timing.

The occurrence of a parity error can be checked either by reading the SPSR register or by using an error interrupt and reading the SPSR register. To use an error interrupt, set the SPCR.SPEIE bit to 1. When executing a serial transfer without using an error interrupt, measures should be taken to ensure the early detection of parity errors, such as reading the SPSR register. When the RSPIA 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.

41.3.10.3 Mode Fault Error

The RSPIA 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 SSL00 input signal of the RSPIA in multi-master mode, the RSPIA detects a mode fault error irrespective of the status of the serial transfer, and sets the SPSR.MODF flag to 1. Upon detecting the mode fault error, the RSPIA copies the value of the pointer to SPCMDm to the SPSSR.SPECM[2:0] bits. The active level of the SSL00 signal is determined by the SSLP.SSL0P bit.
When the MSTR bit is 0, the RSPIA operates in slave mode. The RSPIA detects a mode fault error if the MODFEN bit of the RSPIA in slave mode is 1, and the SPMS bit is 0, and the following conditions are satisfied during the serial transfer period (from the time the driving of valid data is started to the time the final valid data is fetched).

- In the Motorola SPI mode, when the SSL00 input signal is negated during the serial transfer period
- In the TI SSP mode, when the SSL00 input signal is asserted during the serial transfer period
  However, during a burst transfer, no error is detected even if the SSL00 input signal is asserted during the last bit of frame.

Upon detecting a mode fault error, the RSPIA stops driving of the output signals and sets the SPCR.SPE bit to 0. When the SPE bit is cleared, the RSPI function is disabled (refer to section 41.3.12, Initializing RSPIA). 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 the SPSR register or by using an error interrupt and reading the SPSR register. Detecting mode fault errors without utilizing the error interrupt requires polling of the SPSR register. When using the RSPIA in master mode, the pointer value to SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.
When the MODF flag is 1, writing of the value 1 to the SPE bit is ignored by the RSPIA. To enable the RSPI function after the detection of a mode fault error, set the MODF flag to 0.

41.3.10.4 Underrun Error

If RSPIA operates in slave mode (the SPCR.MSTR bit is 0) and the communication mode select bits (CMMD[1:0]) in the RSPI control register (SPCR) is set to 00b or 01b, when serial transfer is started before transmit data output is ready with the SPCR.SPE bit is 1 (RSPI function is enabled), the RSPIA detects an underrun error, and sets the SPSR.MODF and SPSR.UDRF flags to 1.
Upon detecting an underrun error, the RSPIA stops driving of the output signals and sets the SPCR.SPE bit to 0. Clearing of the SPE bit disables the RSPI function. (Refer to section 41.3.12, Initializing RSPIA).
The occurrence of an underrun error can be checked either by reading the SPSR register or by using an error interrupt and reading the SPSR register. Detecting underrun errors without utilizing the error interrupt requires polling of the SPSR register.
When the MODF flag is 1, writing of the value 1 to the SPE bit is ignored by the RSPIA. To enable the RSPI function after the detection of an underrun error, set the MODF flag to 0.

41.3.11 Received Data Ready Detection

When the SPCR.CMMD[1:0] bits are 00b, 01b or 11b, and the value of the SPDRCSR register is not 00h, after receiving data in the receive FIFO during communication (SPE = 1), the SPSR.RRDYF flag is set to 1 when the received data is not stored even after the number of received FIFOs is equal to or less than the threshold value and the value set in the SPDRCSR register has elapsed.

When the receive data ready is detected, the interrupt and event link output can be selected as SPRI or SPEI with the SPCR.RDRIS bit.

Figure 41.53 shows an example of received data ready detection operation.

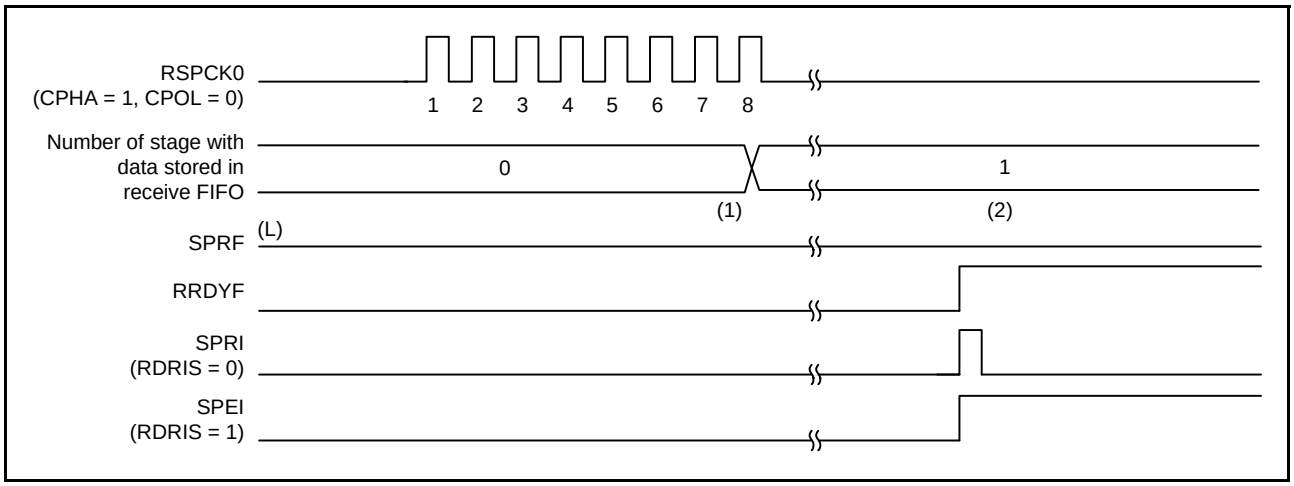


Figure 41.53 Received Data Ready Detection

The operation at the timing shown in steps (1) and (2) in the figure is described below.

(1) Store the received data to the receive FIFO. The SPSR.SPRF flag is 0, because the number of data stored in the receive FIFO is equal to or less than the value of the SPFCR.RTRG[1:0] bits.

(2) Set the RRDYF flag and assert SPRI or SPEI because data is not written to the receive FIFO for the value of the SPDRCSR register from (1).

41.3.12 Initializing RSPIA

If 0 is written to the SPCR.SPE bit or the RSPIA sets the SPE bit to 0 because of the detection of a mode fault error or an underrun error, the RSPIA disables the RSPI function, and initializes some of the module functions. When a system reset is generated, the RSPIA initializes all of the module functions. The following describes initialization by the clearing of the SPCR.SPE bit and initialization by a system reset.

41.3.12.1 Initialization by Clearing the SPE Bit

When the SPCR.SPE bit is set to 0, the RSPIA 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 RSPIA
- Set the SPSR.SPTEF flag to 1

Initialization by the clearing of the SPE bit does not initialize the control bits of the RSPIA. For this reason, the RSPIA can be started in the same transfer mode as prior to the initialization if the SPE bit is set to 1 again.
The SPSR.SPCF, SPRF, OVRF, MODF, PERF, and UDRF flags are not initialized, nor is the value of the RSPI sequence status register (SPSSR) initialized. For this reason, even after the RSPIA is initialized, data from the receive buffer, the communication end status, and the error status during RSPI transfer can be read to check them.
The SPSR.SPTEF flag is initialized to 1. Therefore, if the SPCR.SPTIE bit is set to 1 after RSPIA initialization, a transmit buffer empty interrupt is generated. When the CPU initializes the RSPIA, in order to disable any transmit buffer empty interrupt, 0 should be written to the SPTIE bit simultaneously with the writing of 0 to the SPE bit.

41.3.12.2 System Reset

The initialization by a system reset completely initializes the RSPIA through the initialization of all bits for controlling the RSPIA, initialization of the status bits, and initialization of data registers, in addition to the requirements described in section 41.3.12.1, Initialization by Clearing the SPE Bit.

41.3.13 SPI Operation

41.3.13.1 Master Mode Operation

The only difference between single-master mode operation and multi-master mode operation lies in mode fault error detection (refer to section 41.3.10, Error Detection). When operating in single-master mode, the RSPIA does not detect mode fault errors whereas the RSPIA 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 RSPIA updates the data in the transmit buffer (SPTXn, n = 0 to 3) 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 the RSPIA copies data from the transmit buffer to the shift register and starts serial transfer. Upon copying transmit data to the shift register, the RSPIA changes the status of the shift register to “full”, and upon termination of serial transfer, it changes the status of the shift register to “empty”. The status of the shift register cannot be referenced.
For details on the RSPI transfer format, refer to section 41.3.5, Transfer Format (Frame Format). The polarity of the SSL0n output pins depends on the SSLP register settings.

(2) Terminating a Serial Transfer

[Except receive-only operation in master mode]
Irrespective of the SPCMDm.CPHA bit, the RSPIA terminates the serial transfer after transmitting an RSPCK0 edge corresponding to the final sampling timing. If the number of data stored in the receive FIFO is less than the number of FIFO stages, after the termination of serial transfer, the RSPIA copies data from the shift register to the receive buffer of the SPDR register.
It should be noted that the final sampling timing varies depending on the bit length of transmit data. In master mode, the RSPI data length depends on the SPCMDm.SPB[4:0] bits setting. The polarity of the SSL0n output pin depends on the SSLP register settings.
For details on the RSPI transfer format, refer to section 41.3.5, Transfer Format (Frame Format).

[Receive-only operation in master mode]
The RSPIA terminates the serial transfer when any of the following conditions is satisfied.

- After the RSPIA detects the RSPCK0 edge corresponding to the final sampling timing irrespective of the SPCMDm.CPHA bit
- When writing 1 to the SPRMCR.TERM bit during the serial transfer period

If the number of data stored in the receive FIFO is less the number of FIFO stages, after termination of serial transfer, the RSPIA copies data from the shift register to the receive buffer of the SPDR register.
It should be noted that the final sampling timing varies depending on the bit length of transmit data. In master mode, the RSPI data length depends on the SPCMDm.SPB[4:0] bits setting. The polarity of the SSL0n output pin depends on the SSLP register settings.
For details on the RSPI transfer format, refer to section 41.3.5, Transfer Format (Frame Format).

(3) Sequence Control

The transfer format that is employed in master mode is determined by the following registers.

- RSPI sequence control register (SPSCR)
- RSPI command register m (SPCMDm) (m = 0 to 7)
- RSPI bit rate register (SPBR)
- RSPI clock delay register (SPCKD)
- RSPI slave select negation delay register (SSLND)
- RSPI next-access delay register (SPND)

SPSCR is a register used to determine the sequence configuration for serial transfers that are executed by the RSPIA in master mode. The following items are set in the SPCMDm register: SSL0n 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. The SPBR register holds some of the bit rate settings; the SPCKD register, an RSPI clock delay value; the SSLND register, an SSL negation delay; and the SPND register, a next-access delay value.

According to the sequence length that is assigned to the SPSCR register, the RSPIA makes up a sequence comprised of a part or all of the SPCMDm register. The RSPIA 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 RSPIA loads the pointer to the commands in SPCMD0, and incorporates the SPCMD0 settings into the transfer format at the beginning of serial transfer. The RSPIA 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 RSPIA sets the pointer in SPCMD0, and in this manner the sequence is executed repeatedly.

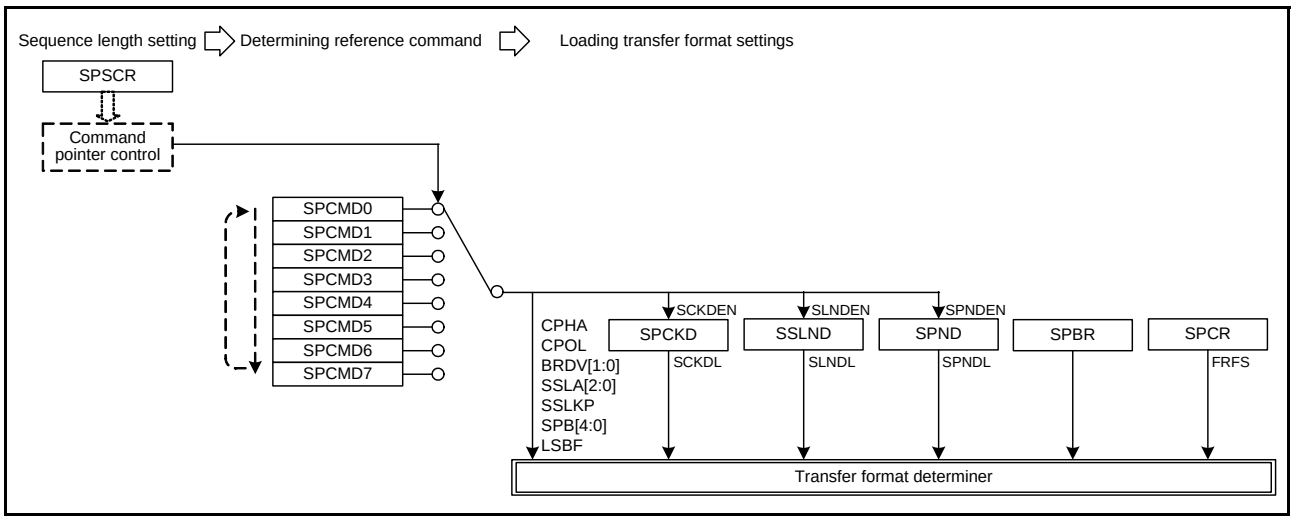


Figure 41.54 Procedure for Determining the Form of Serial Transfer in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

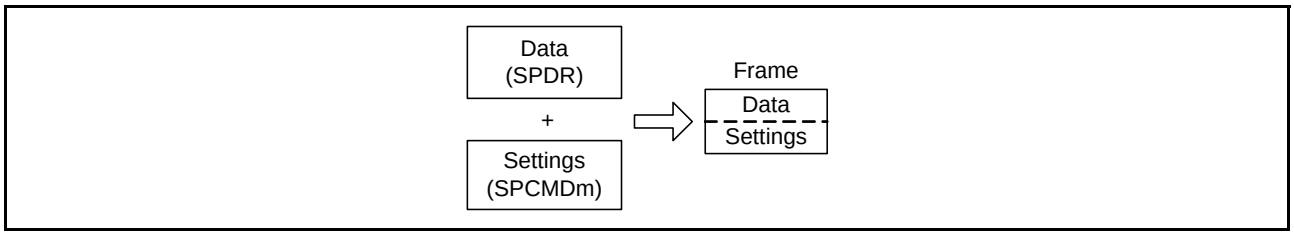


Figure 41.55 Concept of a Frame

Figure 41.56 shows the relationship between the command and the transmit buffer (SPTXn, n = 0 to 3) and receive buffer (SPRXn, n = 0 to 3) in sequence operations.

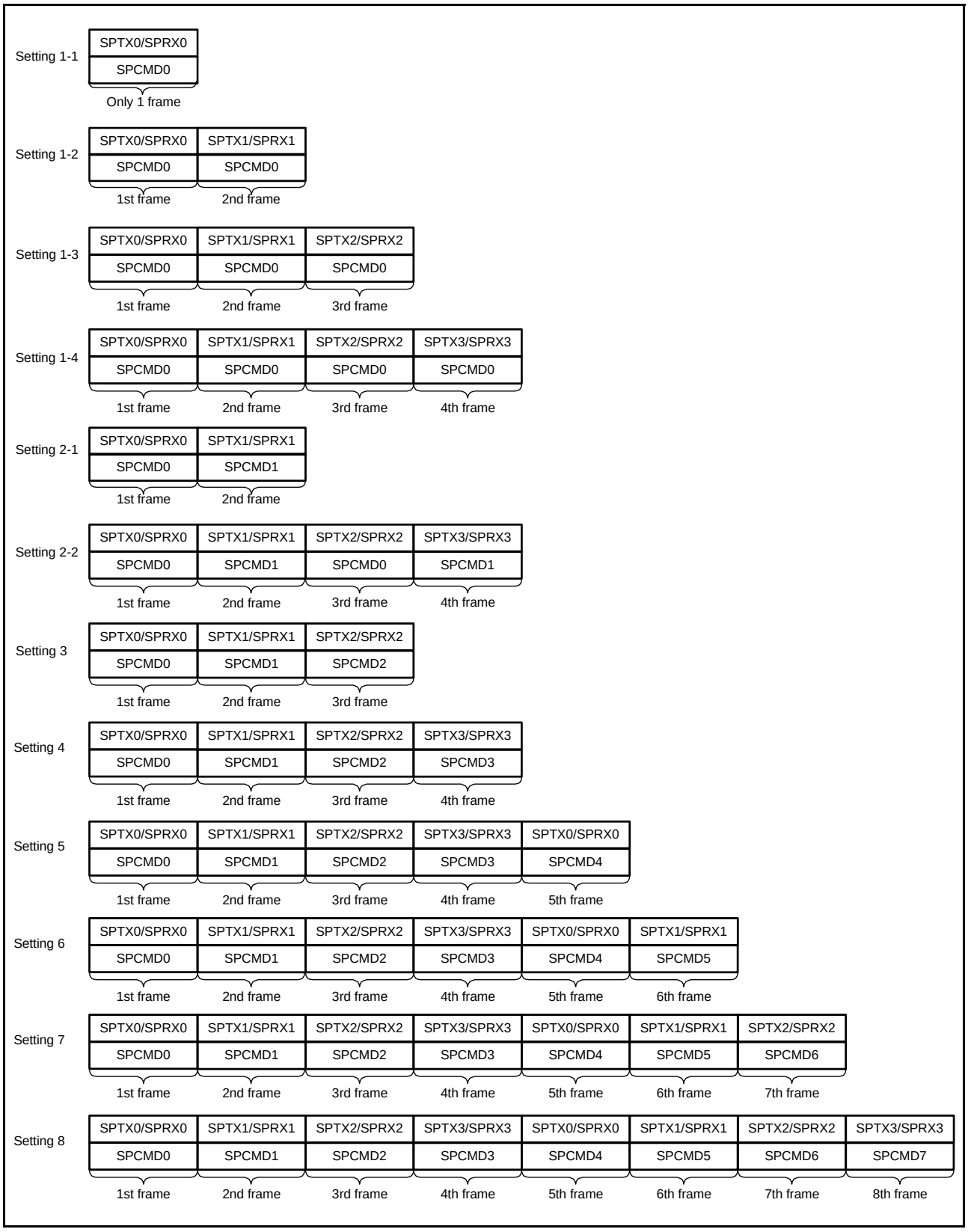


Figure 41.56 Correspondence between the RSPI Command Register m and Transmit/Receive Buffers in Sequence Operations

(4) Burst Transfer

[Motorola SPI]

If the SPCMDm.SSLKP bit that the RSPIA references during the current serial transfer is 1, the RSPIA keeps the SSL0n signal level during the serial transfer until the beginning of the SSL0n signal assertion for the next serial transfer. If the SSL0n signal level for the next serial transfer is the same as the SSL0n signal level for the current serial transfer, the RSPIA can execute continuous serial transfers while keeping the SSL0n signal assertion status (burst transfer).

- When the SPCR.SCKDDIS bit (RSPCK delay between data bytes disable bit) is 0

Figure 41.57 shows an example of an SSL0n 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 (8) as shown in Figure 41.57. It should be noted that the polarity of the SSL0n output signal depends on the SSLP register settings.

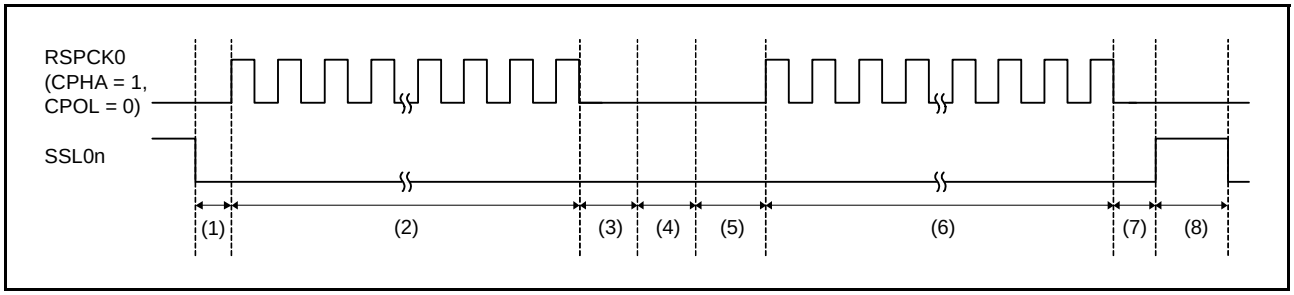


Figure 41.57 Example of Burst Transfer Operation Using SSLKP Bit (SCKDDIS = 0, FRFS = 0)

(1) Based on SPCMD0, the RSPIA asserts the SSL0n signal and inserts RSPCK delays.
(2) The RSPIA executes serial transfers according to SPCMD0.
(3) The RSPIA inserts SSL negation delays.
(4) In transmit-receive mode or transmit-only mode, since the SPCMD0.SSLKP bit is 1, the RSPIA keeps the SSL0n signal value on SPCMD0. This period is sustained, at the shortest, for a period equal to the next-access delay of SPCMD0 and additional 5 PCLKA cycles. 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.
In receive-only mode, since the SPCMD0.SSLKP bit is 1, the RSPIA keeps the SSL0n signal value on SPCMD0. This period is sustained for a period equal to the next-access delay of SPCMD0 and additional 5 PCLKA cycles.
(5) Based on SPCMD1, the RSPIA asserts the SSL0n signal and inserts RSPCK delays.
(6) The RSPIA executes serial transfers according to SPCMD1.
(7) The RSPIA inserts SSL negation delays.
(8) Since the SPCMD1.SSLKP bit is 0, the RSPIA negates the SSL0n signal. In addition, a next-access delay is inserted according to SPCMD1.

Note: If the SSL0n signal output settings in the SPCMDm register in which 1 is assigned to the SSLKP bit (to use for burst transfer) are different from the SSL0n signal output settings in the SPCMDm register to be used in the next transfer, the RSPIA switches the SSL0n signal status to SSL0n signal assertion ((5) in Figure 41.57) corresponding to the command for the next transfer. Note that if such an SSL0n signal switching occurs, the slaves that drive the MISO0 signal compete, and collision of signal levels may occur.

The RSPIA in master mode references the SSL0n signal operation within the module for the case where the SSLKP bit is not used. Even when the SPCMDm.CPHA bit is 0, the RSPIA can accurately start serial transfers by using the SSL0n signal assertion for the next transfer that is detected internally. For this reason, burst transfer in master mode is enabled regardless of the CPHA bit setting. (Refer to section 41.3.13, SPI Operation.)

- When the SPCR.SCKDDIS bit (RSPCK delay between data bytes disable bit) is 1

Figure 41.58 shows an example of an SSL0n 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 (6) as shown in Figure 41.58. It should be noted that the polarity of the SSL0n output signal depends on the SSLP register settings.

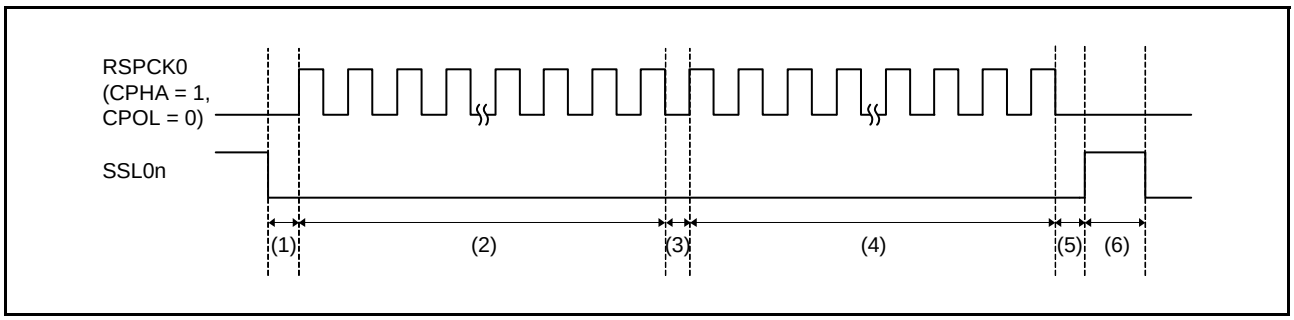


Figure 41.58 Example of Burst Transfer Operation Using SSLKP Bit (SCKDDIS = 1, FRFS = 0)

(1) Based on the SPCMD0 register, the RSPIA asserts the SSL0n signal and inserts RSPCK delays.
The RSPCK delay is inserted only the first frame of burst transfer.

(2) In transmit-receive mode or transmit-only mode, the RSPIA executes serial transfers according to the SPCMD0 register. The RSPIA waits for the last clock output until the next transmit data is stored in the shift register, if the shift register is empty during RSPCK negate period between frames.
In receive-only mode, the RSPIA executes serial transfers according to the SPCMD0 register.

(3) In transmit-receive mode or transmit-only mode, since the SPCMD0.SSLKP bit is 1, the RSPIA keeps the SSL0n signal value on the SPCMD0 register. If the shift register is not empty, RSPCK negate period between frames is 0.5 RSPCK.
In receive-only mode, since it is not the last frame*1, the RSPIA keeps the SSL0n signal value on the SPCMD0 register. RSPCK negate period between frames is 0.5 RSPCK.

(4) The RSPIA executes serial transfers according to the SPCMD1 register.

(5) The RSPIA inserts SSL negation delays.

(6) In transmit-receive mode or transmit-only mode, since the SPCMD1.SSLKP bit is 0, the RSPIA negates the SSL0n signal. In addition, a next-access delay is inserted according to the SPCMD1 register.
In receive-only mode, the RSPIA negates the SSL0n signal. In addition, a next-access delay is inserted according to the SPCMD1 register.

Note 1. In receive-only mode, the last frame is a frame set by the SPRMCR.RFC[4:0] bits when SPRMCR.RFC[4:0] ≠ 00000b or a frame in which SPRMCR.TERM = 1 has been accepted.

[TI SSP]
The RSPIA asserts the SSL0n signal for one cycle at the start of serial transfer.
The RSPIA can execute continuous serial transfers by asserting the SSL0n signal for one cycle at the start of serial transfer (burst transfer).

- When the SPCMDm.SSLKP bit (SSL signal level keeping bit) is 1, and the SPCR.SCKDDIS bit (RSPCK delay between data bytes disable bit) is 1

Figure 41.59 shows an example of an SSL0n signal operation and the serial data MISO0/MOSI0 for the case where a burst transfer is implemented using the SPCMD0 and SPCMD1 register settings. The text below explains the RSPI operations (1) to (6) as shown in Figure 41.59. It should be noted that the polarity of the SSL0n output signal depends on the SSLP register settings.

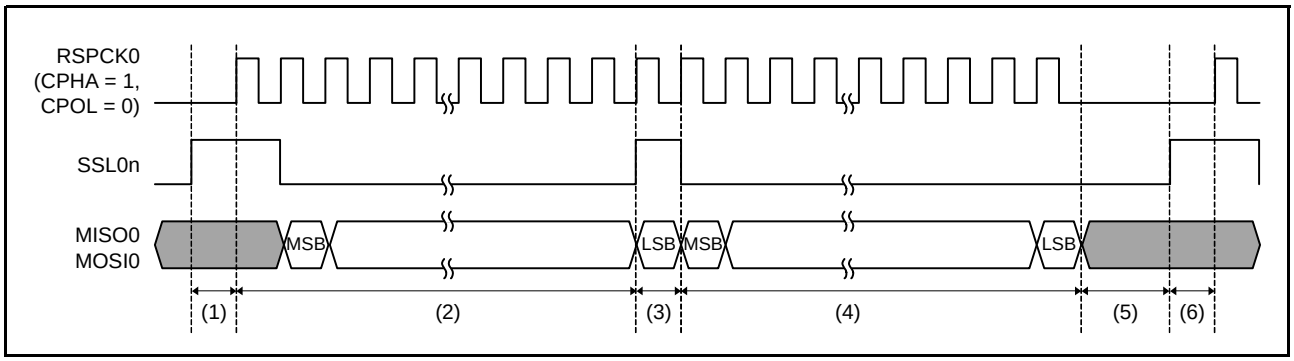


Figure 41.59 Example of Burst Transfer Operation (FRFS = 1)

(1) Based on the SPCMD0 register, the RSPIA asserts the SSL0n signal and inserts RSPCK delays.
The RSPCK delay is inserted only the first frame of burst transfer.
(2) The RSPIA executes serial transfers according to the SPCMD0 register.
(3) Final data transfer and the SSL0n assertion are performed simultaneously.
During transmit-receive/transmit-only operation, the RSPIA waits for the last clock output until the next transmit data is stored in the shift register, if the shift register is empty during RSPCK negate period between frames.
(4) The RSPIA executes serial transfers according to the SPCMD1 register.
(5) The RSPIA inserts SSL0n negation delays at the last frame*1.
(6) A next-access delay is inserted according to the SPCMD1 register.

Note 1. In receive-only mode, the last frame is a frame set by the SPRMCR.RFC[4:0] bits when SPRMCR.RFC[4:0] ≠ 00000b or a frame in which SPRMCR.TERM = 1 has been accepted.
Note: If the SSL0n signal output settings in the SPCMDm register in which 1 is assigned to the SSLKP bit are different from the SSL0n signal output settings in the SPCMDm register to be used in the next transfer, the RSPIA switches the SSL0n signal status to SSL0n signal assertion ((5) in Figure 41.59) corresponding to the command for the next transfer. Note that if such an SSL0n signal switching occurs, the slaves that drive the MISO0 signal compete, and collision of signal levels may occur.

(5) RSPCK Delay (t1)

The RSPCK delay value of the RSPIA in master mode depends on the SPCMDm.SCKDEN bit setting and the SPCKD register setting. The RSPIA 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 the SPCKD.SCKDL[2:0] bits, as listed in Table 41.9. For a definition of RSPCK delay, refer to section 41.3.5, Transfer Format (Frame Format).

The RSPCK delay is inserted only the first frame of burst transfer when data transfer is performed without delay between frames in burst transfer (SPCMDm.SSLKP = 1, SPCR.SCKDDIS = 1).

Table 41.9 Relationship among SCKDEN Bit, SCKDL[2:0] Bits, and RSPCK Delay Value

		RSPCK Delay Value	
SPCMDm.SCKDEN Bit	**SPCKD.SCKDL[2:0] Bits**	**Motorola SPI**	**TI SSP**
0	000b to 111b	1 RSPCK	0 RSPCK
1	000b	1 RSPCK	1 RSPCK
	001b	2 RSPCK	2 RSPCK
	010b	3 RSPCK	3 RSPCK
	011b	4 RSPCK	4 RSPCK
	100b	5 RSPCK	5 RSPCK
	101b	6 RSPCK	6 RSPCK
	110b	7 RSPCK	7 RSPCK
	111b	8 RSPCK	8 RSPCK

(6) SSL Negation Delay (t2)

The SSL negation delay value of the RSPIA in master mode depends on the SPCMDm.SLNDEN bit setting and the SSLND register setting. The RSPIA 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.SLNDL[2:0] bits, as listed in Table 41.10. For a definition of SSL negation delay, refer to section 41.3.5, Transfer Format (Frame Format).

The SSL negation delay is inserted only the last frame of burst transfer when data transfer is performed without delay between frames in burst transfer (SPCMDm.SSLKP = 1, SPCR.SCKDDIS = 1).

Table 41.10 Relationship among SLNDEN Bit, SLNDL[2:0] Bits, and SSL Negation Delay Value

SPCMDm.SLNDEN Bit	**SSLND.SLNDL[2:0] Bits**	**SSL Negation Delay Value**
0	000b to 111b	1 RSPCK
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

(7) Next-Access Delay (t3)

The next-access delay value of the RSPIA in master mode depends on the SPCMDm.SPNDEN bit setting and the SPND setting. The RSPIA 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 the SPND.SPNDL[2:0] bits, as listed in Table 41.11. For a definition of next-access delay, refer to section 41.3.5, Transfer Format (Frame Format).

The next-access delay is inserted only the last frame of burst transfer when data transfer is performed without delay between frames in burst transfer (SPCMDm.SSLKP = 1, SPCR.SCKDDIS = 1).

Table 41.11 Relationship among SPNDEN Bit, SPNDL[2:0], and Next-Access Delay Value

SPCMDm.SPNDEN Bit	SPND.SPNDL[2:0] Bits	Next-Access Delay Value
0	000b to 111b	1 RSPCK + 5 PCLKA
1	000b	1 RSPCK + 5 PCLKA
	001b	2 RSPCK + 5 PCLKA
	010b	3 RSPCK + 5 PCLKA
	011b	4 RSPCK + 5 PCLKA
	100b	5 RSPCK + 5 PCLKA
	101b	6 RSPCK + 5 PCLKA
	110b	7 RSPCK + 5 PCLKA
	111b	8 RSPCK + 5 PCLKA

(8) Initialization Flowchart

Figure 41.60 is a flowchart illustrating an example of initialization in SPI operation when the RSPIA is used in master mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

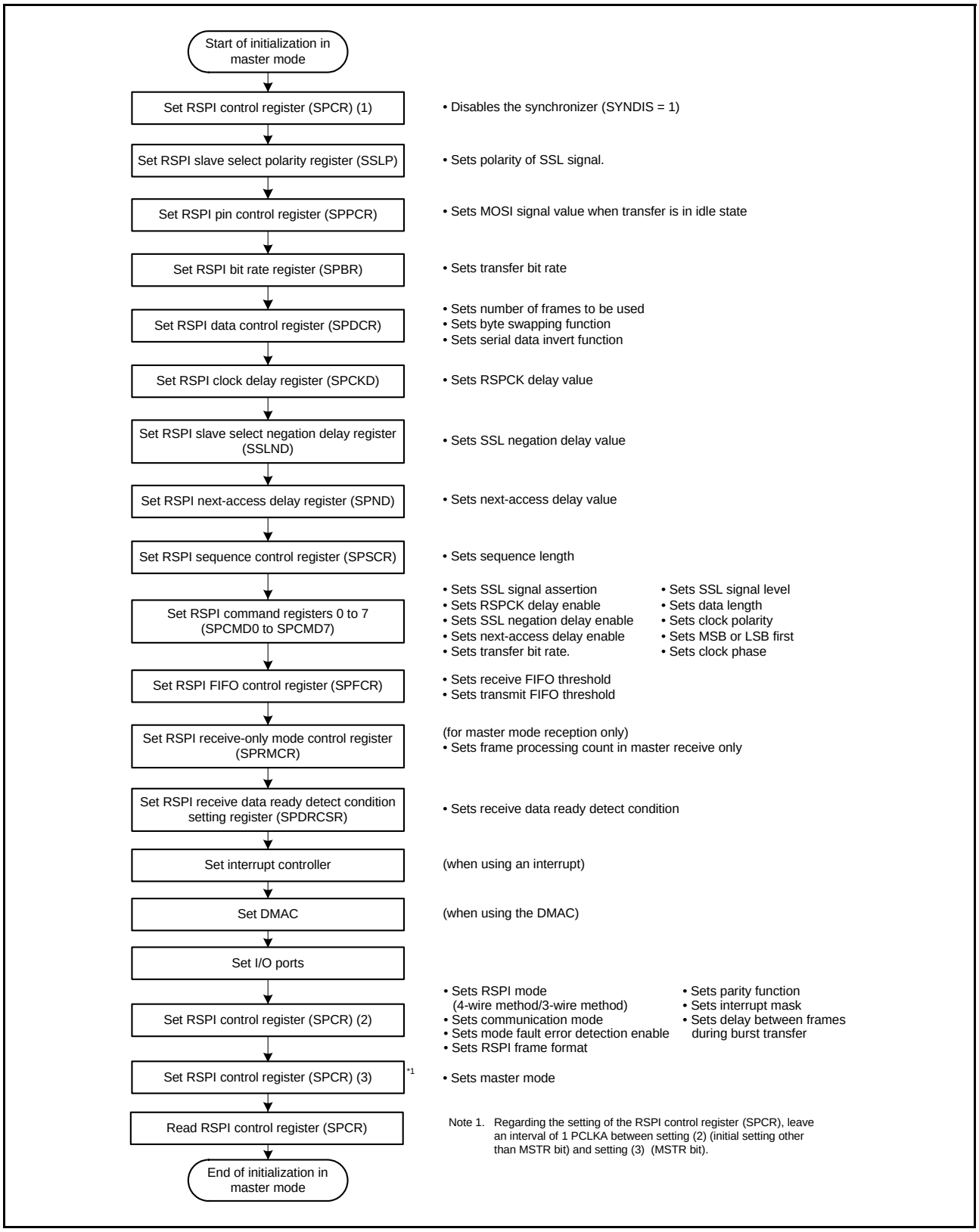


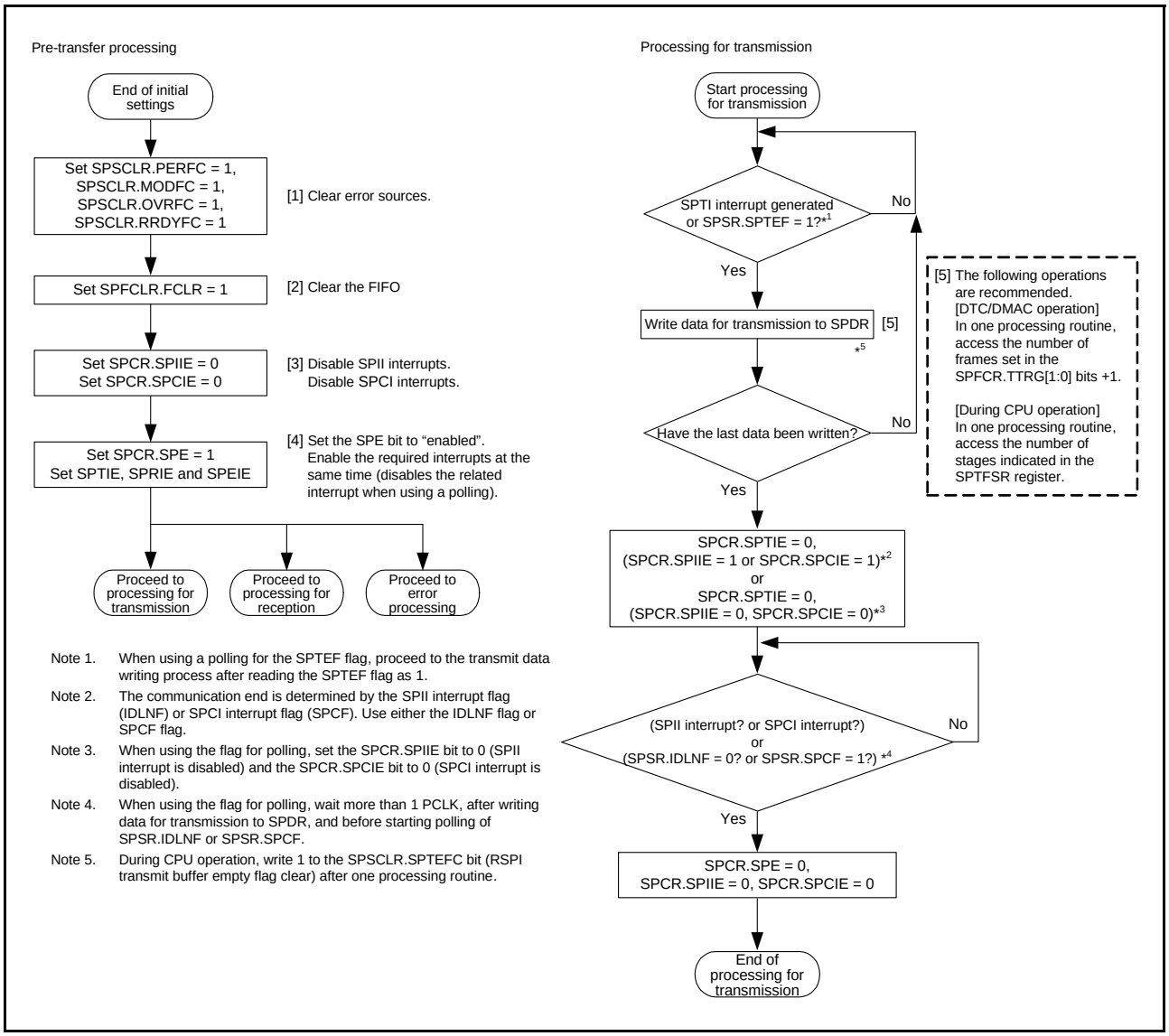
Figure 41.60 Example of Initialization Flowchart in Master Mode (SPI Operation)

(9) Software Processing Flow

Figure 41.61 to Figure 41.63 show examples of the flow of software processing.

(a) Transmit Processing Flow

Enabling SPII or SPCI interrupt after final transmit data is written makes it possible to notify the CPU that transmission of all data has been completed.



Note 1. When using a polling for the SPTEF flag, proceed to the transmit data writing process after reading the SPTEF flag as 1.

Note 2. The communication end is determined by the SPII interrupt flag (IDLNF) or SPCI interrupt flag (SPCF). Use either the IDLNF flag or SPCF flag.

Note 3. When using the flag for polling, set the SPCR.SPIIE bit to 0 (SPII interrupt is disabled) and the SPCR.SPCIE bit to 0 (SPCI interrupt is disabled).

Note 4. When using the flag for polling, wait more than 1 PCLK, after writing data for transmission to SPDR, and before starting polling of SPSR.IDLNF or SPSR.SPCF.

Note 5. During CPU operation, write 1 to the SPSCLR.SPTEFC bit (RSPI transmit buffer empty flag clear) after one processing routine.

Figure 41.61 Flowchart in Master Mode (Transmission)

(b) Receive Processing Flow

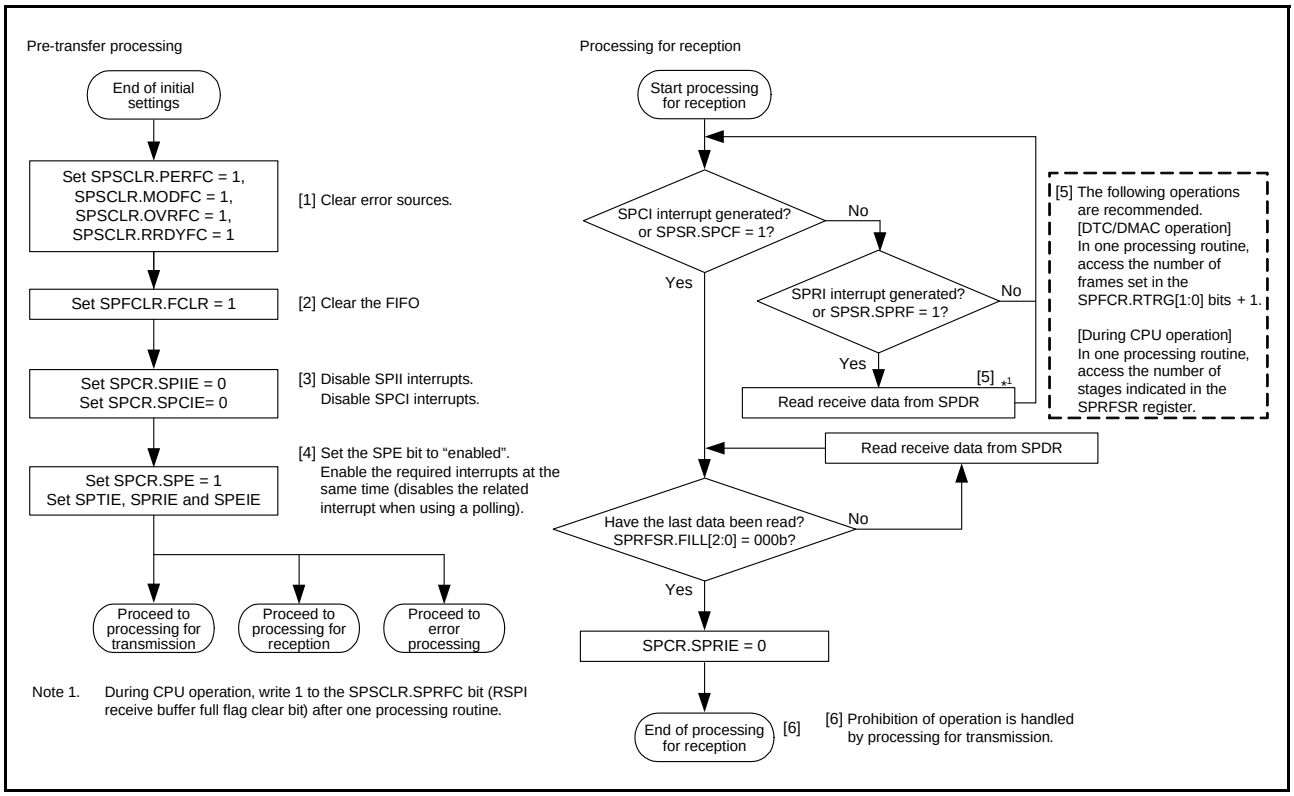


Figure 41.62 Flowchart in Master Mode (Reception)

(c) Receive-Only Processing Flow

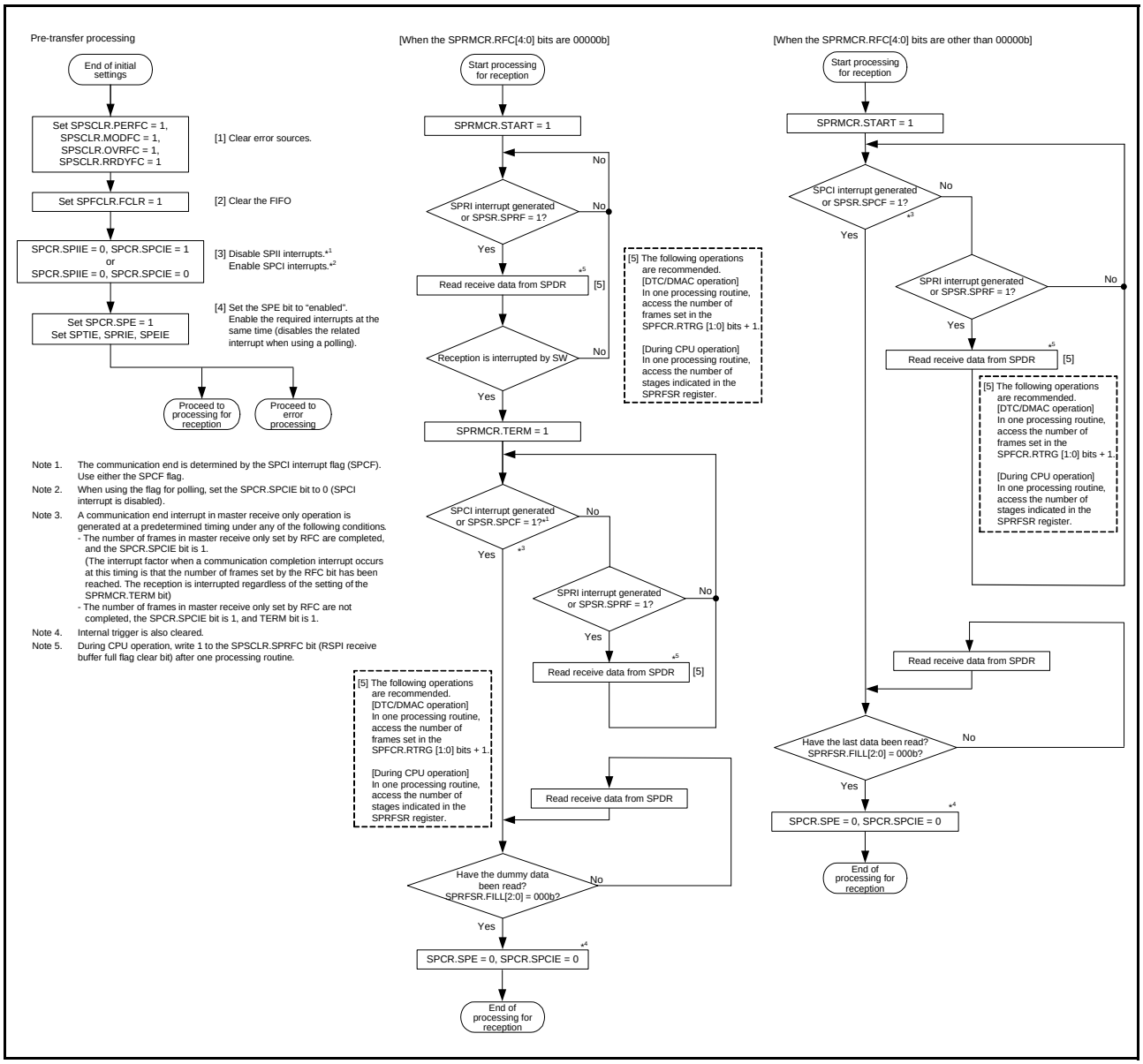


Figure 41.63 Flowchart in Master Mode (Receive-Only)

(d) Flow of Error Processing

When a mode fault error is generated, the SPCR.SPE bit is automatically cleared, stopping operations for transmission and reception. For errors from other sources, however, the SPCR.SPE bit is not cleared and operations for transmission and reception continue; accordingly, we recommend clearing of the SPCR.SPE bit to stop operations in the case of errors other than mode fault errors. Not doing so will lead to updating of the SPSSR.SPECM[2:0] bits.

When interrupts are used and an error occurs, if the ICU.IRn.IR flag for the SPTI or SPRI interrupt request is set to 1, clear the ICU.IRn.IR flag in the error processing routine. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPIA.

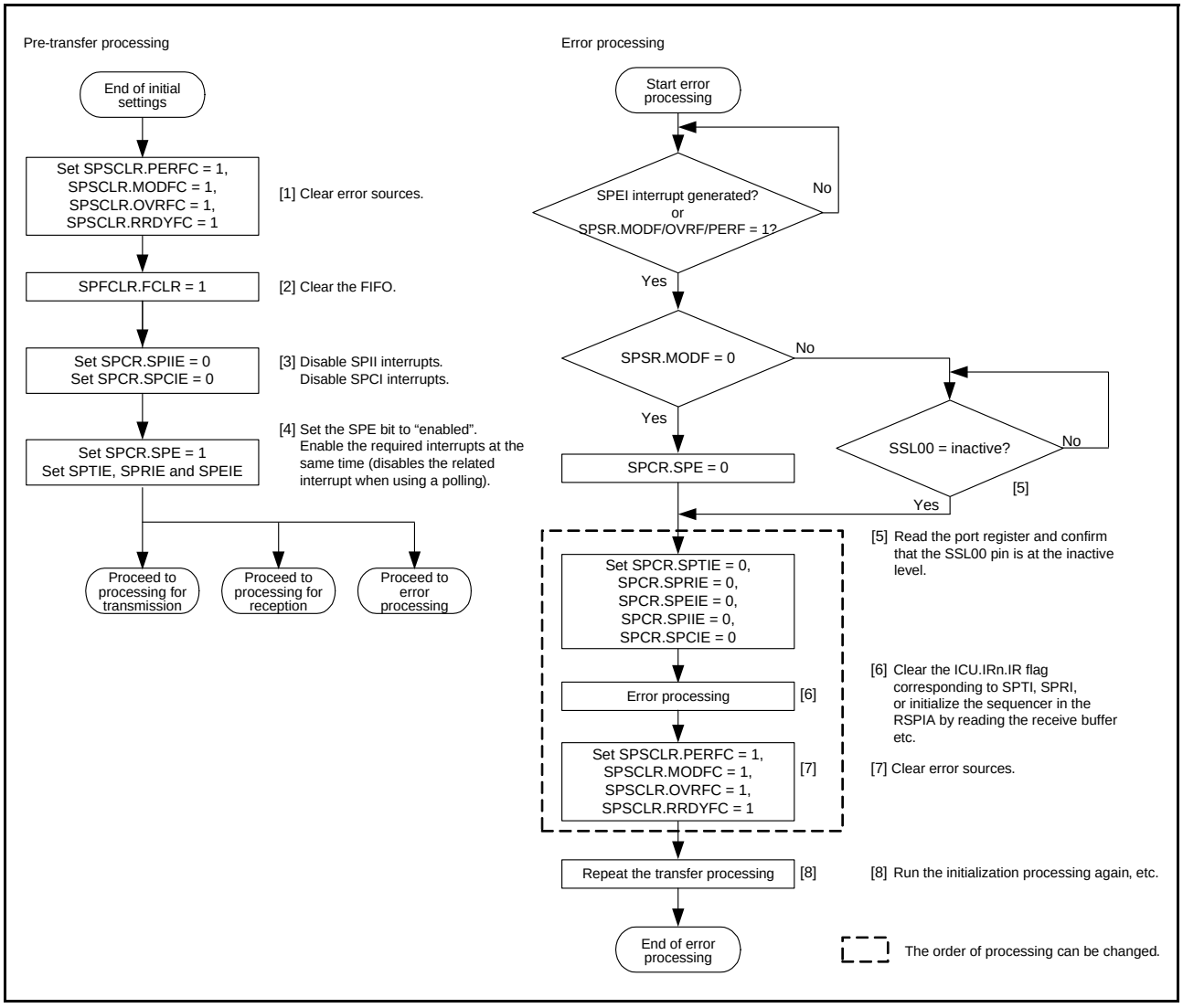


Figure 41.64 Flowchart for Master Mode (Error Processing)

41.3.13.2 Slave Mode Operation

(1) Starting a Serial Transfer

If the SPCMD0.CPHA bit is 0, when detecting an SSL00 input signal assertion, the RSPIA needs to start driving valid data to the MISO0 output signal. For this reason, when the CPHA bit is 0, the assertion of the SSL00 input signal triggers the start of a serial transfer.
If the CPHA bit is 1, when detecting the first RSPCK0 edge in an SSL00 signal asserted condition, the RSPIA needs to start driving valid data to the MISO0 output signal. For this reason, when the CPHA bit is 1, the first RSPCK0 edge in an SSL00 signal asserted condition triggers the start of a serial transfer.
Irrespective of the CPHA bit setting, the timing at which the RSPIA starts driving of the MISO0 output signal is the SSL00 signal assertion timing. The data which is output by the RSPIA is either valid or invalid, depending on the CPHA bit setting.
For details on the RSPI transfer format, refer to section 41.3.5, Transfer Format (Frame Format). The polarity of the SSL00 input signal depends on the setting of the SSLP.SSL0P bit.

(2) Terminating a Serial Transfer

Irrespective of the SPCMD0.CPHA bit, the RSPIA terminates the serial transfer after detecting an RSPCK0 edge corresponding to the final sampling timing. When the number of data stored in the receive FIFO is less than the number of FIFO stages, upon termination of serial transfer the RSPIA copies received data from the shift register to the receive buffer of the SPDR register. Upon termination of a serial transfer the RSPIA changes the status of the shift register to “empty”, regardless of the receive buffer state. A mode fault error occurs if the RSPIA detects an SSL00 input signal negation from the beginning of serial transfer to the end of serial transfer (refer to section 41.3.10, Error Detection).
The final sampling timing changes depending on the bit length of transmit data. In slave mode, the RSPI data length depends on the SPCMD0.SPB[4:0] bit setting. The polarity of the SSL00 input signal depends on the SSLP.SSL0P bit setting. For details on the RSPI transfer format, refer to section 41.3.5, Transfer Format (Frame Format).

(3) Notes on Single-Slave Operation

[Motorola SPI]
If the SPCMD0.CPHA bit is 0, when detecting an SSL00 input signal assertion edge, the RSPIA starts a serial transfer. Upon the SSL00 input signal is fixed to the active level when using the RSPIA in single slave mode in a configuration as shown in Figure 41.7, the RSPIA cannot start a serial transfer correctly while the CPHA bit is 0. To correctly perform transmission and reception of the RSPIA in slave mode in a configuration where the SSL00 input signal is fixed to the active level, set the CPHA bit to 1. If the CPHA bit must be set to 0, do not fix the SSL00 input signal to active level.

[TI SSP]
In the TI SSP mode, upon the SSL00 input signal is fixed to the inactive level when using the RSPIA in single slave mode in a configuration as shown in Figure 41.7, the RSPIA cannot start a serial transfer correctly.
When using the RSPIA in single slave mode, operate in a configuration as shown in Figure 41.6.

(4) Burst Transfer

[Motorola SPI]
When the SPCMD0.CPHA bit is 1, continuous serial transfer (burst transfer) is available while maintaining the SSL00 input signal assertion state. When the CPHA bit is 1, the period (from the first RSPCK0 edge in the SSL00 input signal active state until the sampling timing for receiving the final bit) corresponds to the serial transfer period. Even if the SSL00 input signal is always at the active level, start of an access can be detected, which allows burst transfer.
When the CPHA bit is 0, the second and subsequent serial burst transfers cannot be performed correctly for the same reason as (3).

[TI SSP]

In serial transfer, data transfer starts after the SSL0n input signal is asserted for RSPCK 1 cycle. Since frame transfer starts from the SSL0n input signal, SSL must be asserted between frames.

(5) Initialization Flowchart

Figure 41.65 shows an example of initialization flow when using the RSPIA in slave-mode SPI operation. For how to set the interrupt controller, DMAC, and input/output ports, see descriptions of each block.

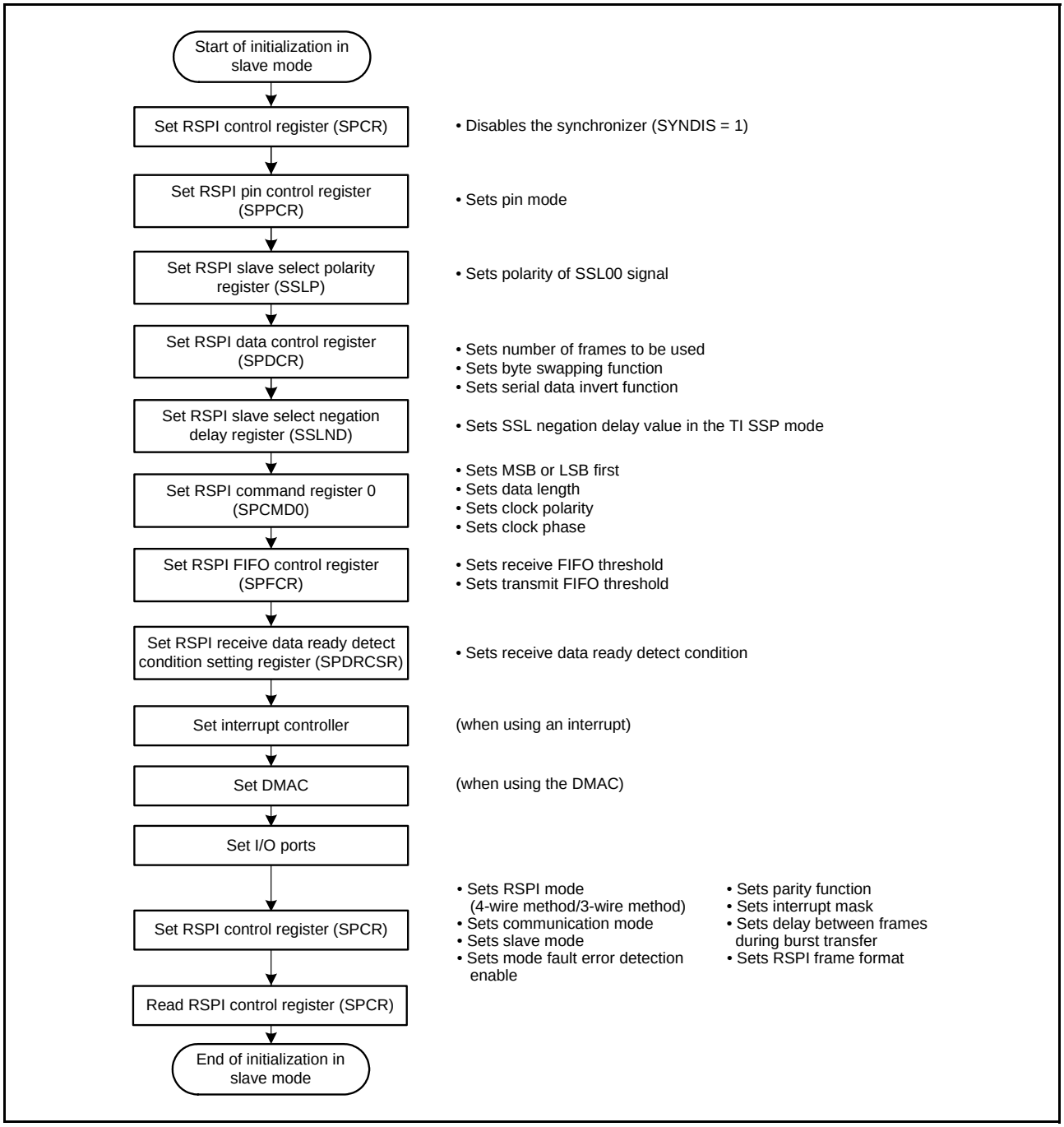
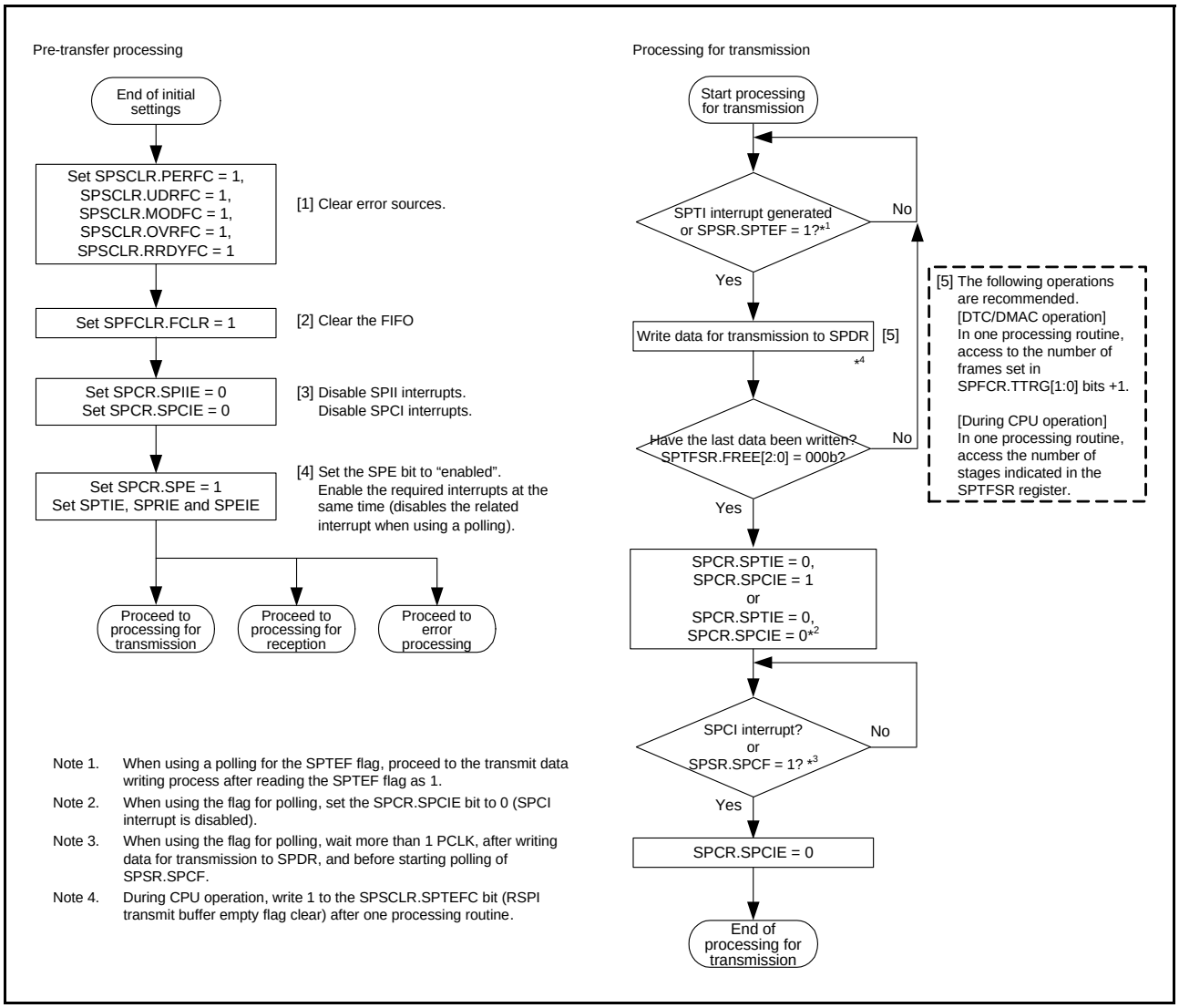


Figure 41.65 Example of Initialization Flow in Slave Mode

(6) Software Processing Flow

Figure 41.66 to Figure 41.69 show examples of the flow of software processing.

(a) Transmit Processing Flow



Note 1. When using a polling for the SPTEF flag, proceed to the transmit data writing process after reading the SPTEF flag as 1.

Note 2. When using the flag for polling, set the SPCR.SPCIE bit to 0 (SPCI interrupt is disabled).

Note 3. When using the flag for polling, wait more than 1 PCLK, after writing data for transmission to SPDR, and before starting polling of SPSR.SPCF.

Note 4. During CPU operation, write 1 to the SPSCLR.SPTEFC bit (RSPI transmit buffer empty flag clear) after one processing routine.

Figure 41.66 Flowchart in Slave Mode (Transmission)

(b) Receive Processing Flow

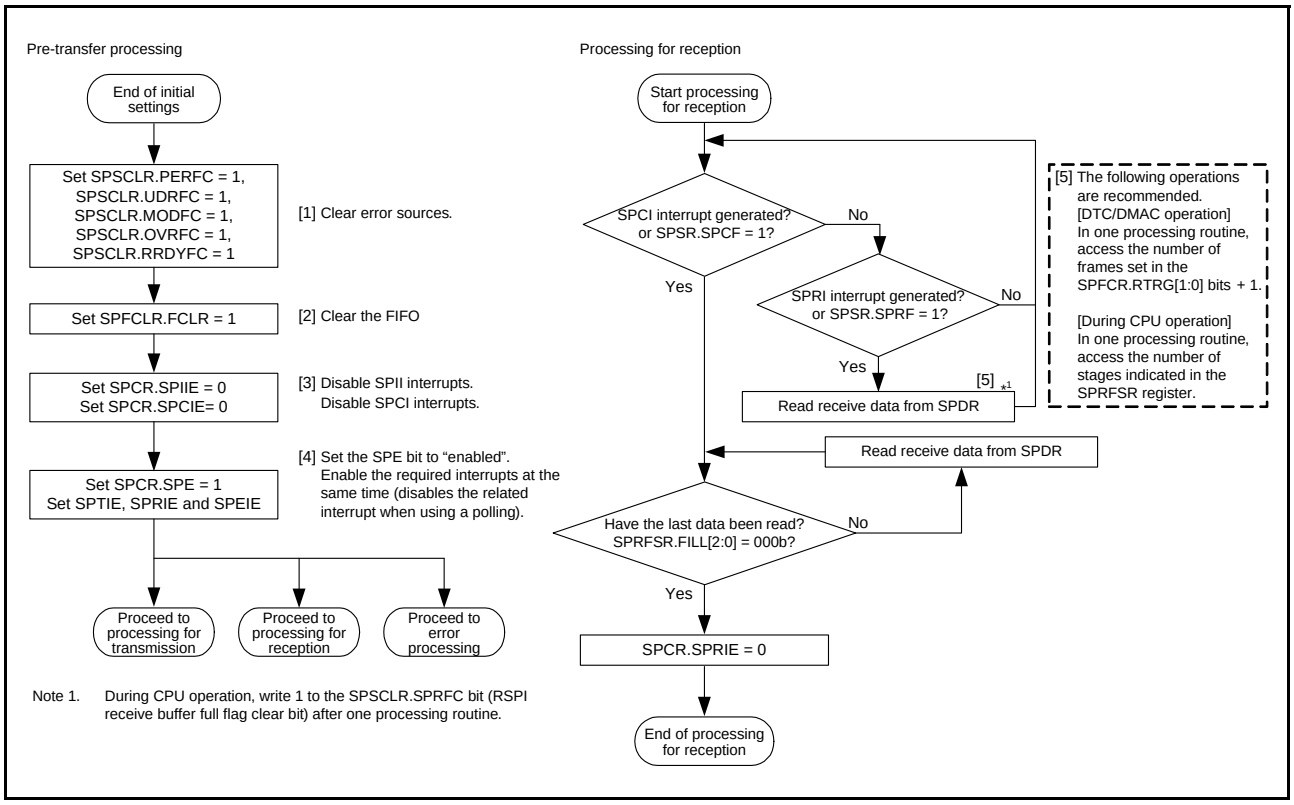
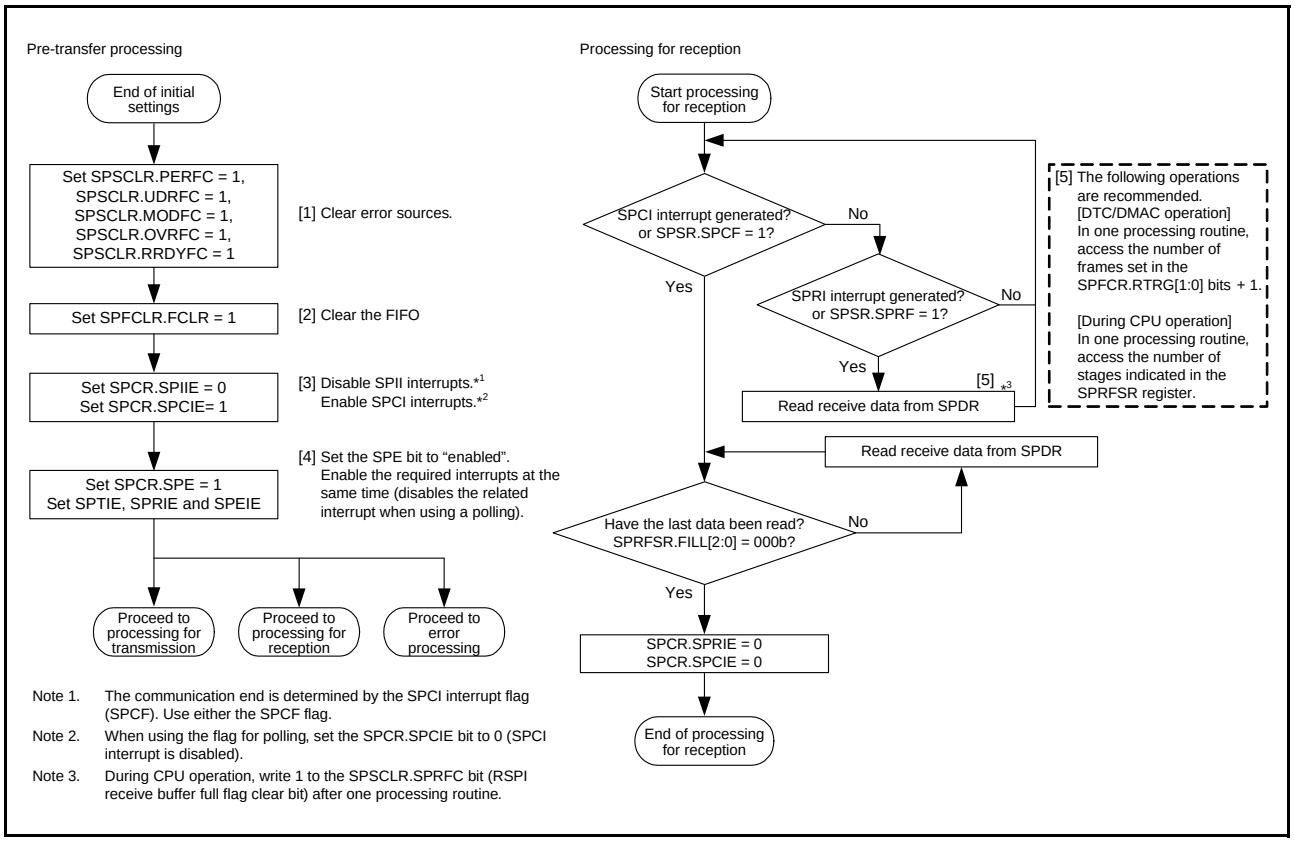


Figure 41.67 Flowchart in Slave Mode (Reception)

(c) Receive-Only Processing Flow



Note 1. The communication end is determined by the SPCI interrupt flag (SPCF). Use either the SPCF flag.
Note 2. When using the flag for polling, set the SPCR.SPCIE bit to 0 (SPCI interrupt is disabled).
Note 3. During CPU operation, write 1 to the SPSCLR.SPRFC bit (RSPI receive buffer full flag clear bit) after one processing routine.

Figure 41.68 Flowchart in Slave Mode (Receive-Only)

(d) Flow of Error Processing

In slave mode operation, the MODF flag can be cleared regardless of the SSL00 pin status even when a mode fault error is generated.

When interrupts are used and an error occurs, if the ICU.IRn.IR flag for the SPTI or SPRI interrupt request is set to 1, clear the ICU.IRn.IR flag in the error processing routine. If the SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the RSPIA.

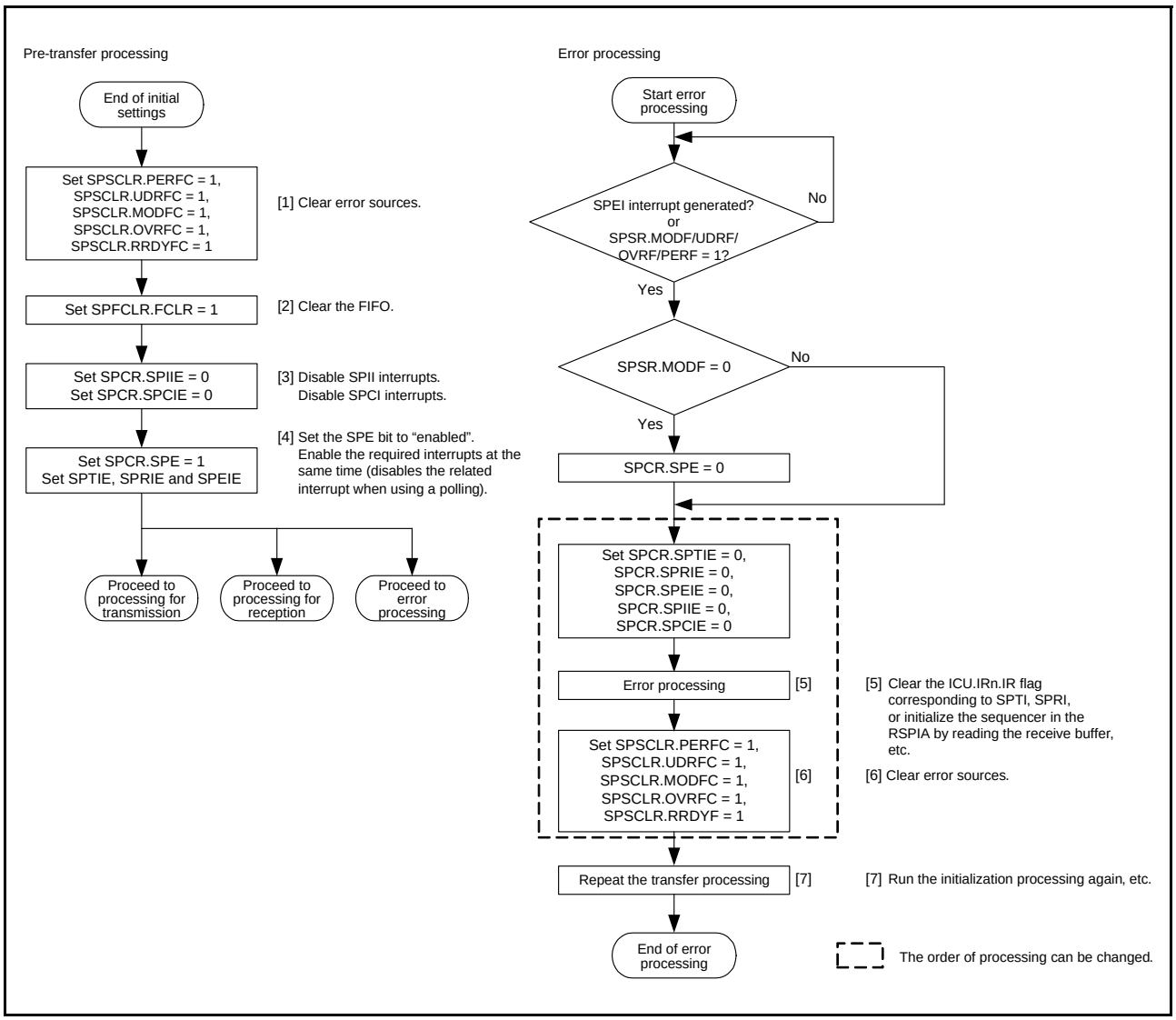


Figure 41.69 Flowchart in Slave Mode (Error Processing)

41.3.14 Clock Synchronous Operation

Setting the SPCR.SPMS bit to 1 selects clock synchronous operation of the RSPIA. In clock synchronous operation, the SSL0n pin is not used, and the three pins of RSPCK0, MOSI0, and MISO0 handle communications. The SSL0n pin is available as I/O port pins.
Although clock synchronous operation does not require use of the SSL0n pin, operation of the module is the same as in SPI operation. That is, in both master and slave mode operations, communications can be performed with the same flow as in SPI operation. However, mode fault errors are not detected because the SSL0n pin is not used.
Furthermore, do not set the SPCMDm.CPHA bit to 0 if clock synchronous operation is to proceed in slave mode (SPCR.MSTR = 0).

41.3.14.1 Master Mode Operation

(1) Starting a Serial Transfer

The RSPIA updates the data in the transmit buffer (SPTXn, n = 0 to 3) of the SPDR register when data is written to the SPDR register with the transmit buffer being empty (data for the next transfer is not set). When the shift register is empty, the RSPIA copies data from the transmit buffer to the shift register and starts serial transmission. Upon copying transmit data to the shift register, the RSPIA changes the status of the shift register to “full”, and upon termination of serial transfer, it changes the status of the shift register to “empty”. The status of the shift register cannot be referenced.
For details on the RSPI transfer format, refer to section 41.3.5, Transfer Format (Frame Format).
However, transfer in clock synchronous operation is conducted without the SSL0n output signal.

(2) Terminating a Serial Transfer

The RSPIA terminates the serial transfer after transmitting an RSPCK0 edge corresponding to the final sampling timing. If the number of data stored in the receive FIFO is less than the number of FIFO stages, upon termination of serial transfer, the RSPIA 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 transmit data. In master mode, the RSPI data length depends on the SPCMDm.SPB[4:0] bit setting. For details on the RSPI transfer format, refer to section 41.3.5, Transfer Format (Frame Format).
However, transfer in clock synchronous operation is conducted without the SSL0n output signal.

(3) Sequence Control

The transfer format employed in master mode is determined by the following registers:

- RSPI sequence control register (SPSCR)
- RSPI command register m (SPCMDm) (m = 0 to 7)
- RSPI bit rate register (SPBR)
- RSPI clock delay register (SPCKD)
- RSPI slave select negation delay register (SSLND)
- RSPI next-access delay register (SPND)

Although the SSL0n signals are not output in clock synchronous operation, these settings are valid.

The SPSCR register is a register used to determine the sequence configuration for serial transfers that are executed by the RSPIA in master mode. The following items are set in the SPCMDm register: MSB/LSB first, data length, some of the bit rate settings, RSPCK polarity/phase, whether the SPCKD register is to be referenced, whether the SSLND register is to be referenced, and whether the SPND register is to be referenced. The SPBR register holds some of the bit rate settings; the SPCKD register, an RSPI clock delay value; the SSLND register, an SSL negation delay; and the SPND register, a next-access delay value.
According to the sequence length that is assigned to the SPSCR register, the RSPIA makes up a sequence comprised of a part or all of the SPCMDm register. The RSPIA 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 RSPIA loads the pointer to the commands in the SPCMD0 register, and incorporates the SPCMD0 register setting into the transfer format at the beginning of serial transfer. The RSPIA 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 RSPIA sets the pointer in the SPCMD0 register, and in this manner the sequence is executed repeatedly.

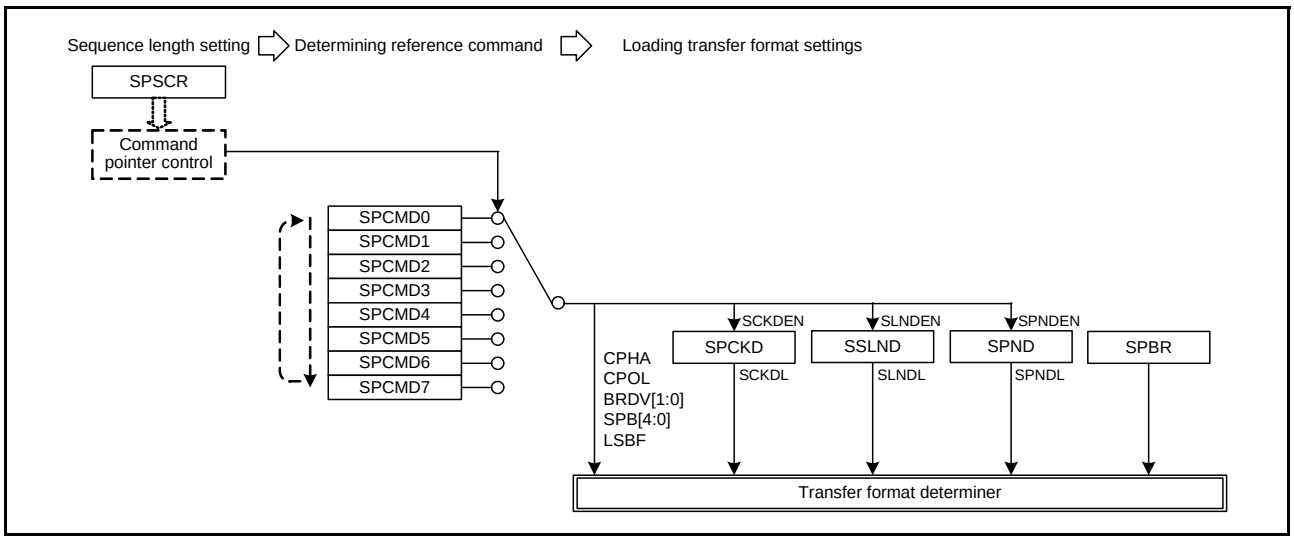


Figure 41.70 Procedure for Determining the Form of Serial Transfer in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

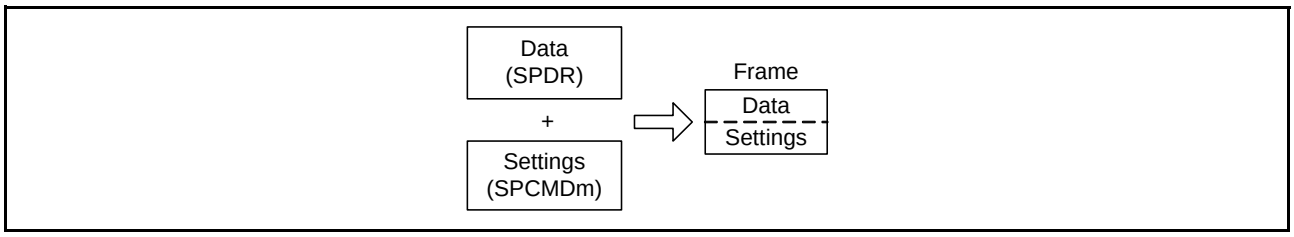


Figure 41.71 Concept of a Frame

Figure 41.72 shows the relationship between the command and the transmit buffer (SPTXn, n = 0 to 3) and receive buffer (SPRXn, n = 0 to 3) in the sequence of operations.

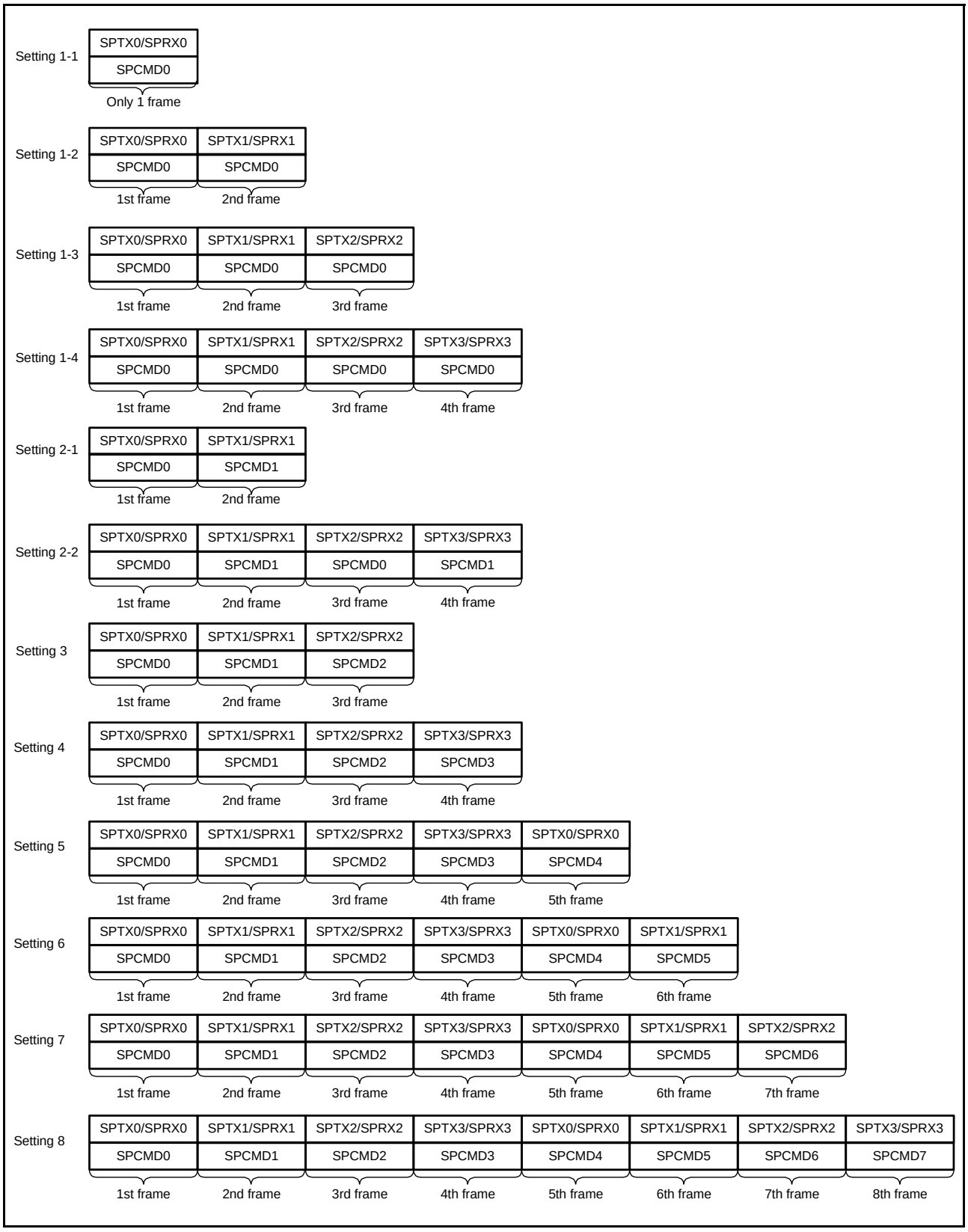


Figure 41.72 Correspondence between the RSPI Command Register m and Transmit/Receive Buffers in Sequence Operations

(4) Initialization Flowchart

Figure 41.73 is a flowchart illustrating an example of initialization in clock synchronous operation when the RSPIA is used in master mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

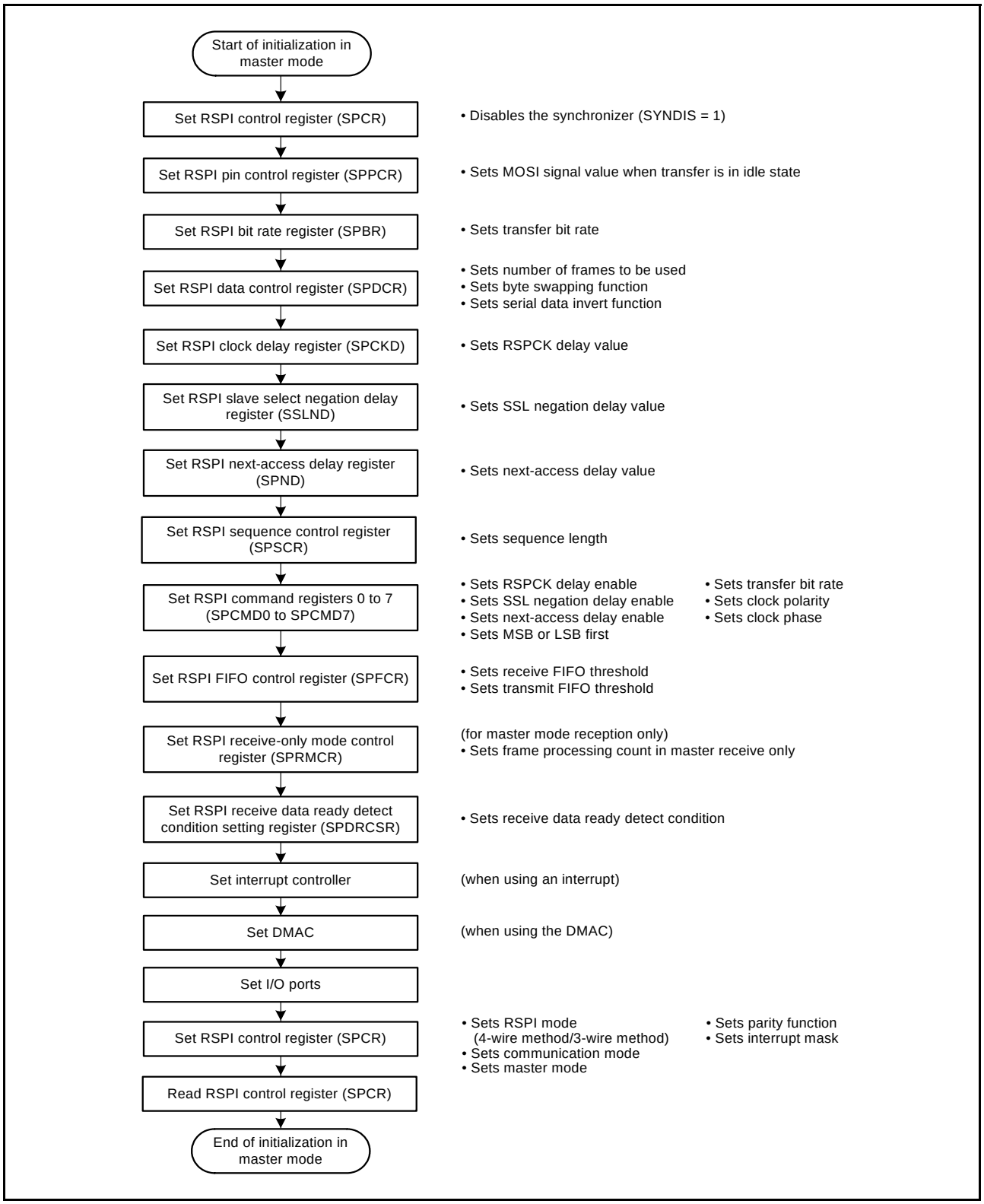


Figure 41.73 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation)

(5) Flow of Software Processing

Software processing during clock-synchronous operation when the RSPIA is used in master mode is the same as that for SPI master mode operation. For details, refer to section 41.3.13.1, (9) Software Processing Flow. Note that mode fault errors will not occur.

41.3.14.2 Slave Mode Operation

(1) Starting a Serial Transfer

When the SPCR.SPMS bit is 1, the first RSPCK0 edge triggers the start of a serial transfer in the RSPIA.
When the SPMS bit is 1, the RSPIA drives the MISO0 output signal.
For details on the RSPI transfer format, refer to section 41.3.5, Transfer Format (Frame Format).
It should be noted that the SSL00 input signal is not used in clock synchronous operation.

(2) Terminating a Serial Transfer

The RSPIA terminates the serial transfer after detecting an RSPCK0 edge corresponding to the final sampling timing. When the number of data stored in the receive FIFO is less than the number of FIFO stages, upon termination of serial transfer the RSPIA copies received data from the shift register to the receive buffer of the SPDR register. Upon termination of a serial transfer the RSPIA changes the status of the shift register to “empty” regardless of the receive buffer status. The final sampling timing changes depending on the bit length of transmit data. In slave mode, the RSPI data length depends on the SPCMD0.SPB[4:0] bit setting.
For details on the RSPI transfer format, refer to section 41.3.5, Transfer Format (Frame Format).

(3) Initialization Flowchart

Figure 41.74 is a flowchart illustrating an example of initialization in clock synchronous operation when the RSPIA is used in slave mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, refer to the descriptions given in the individual blocks.

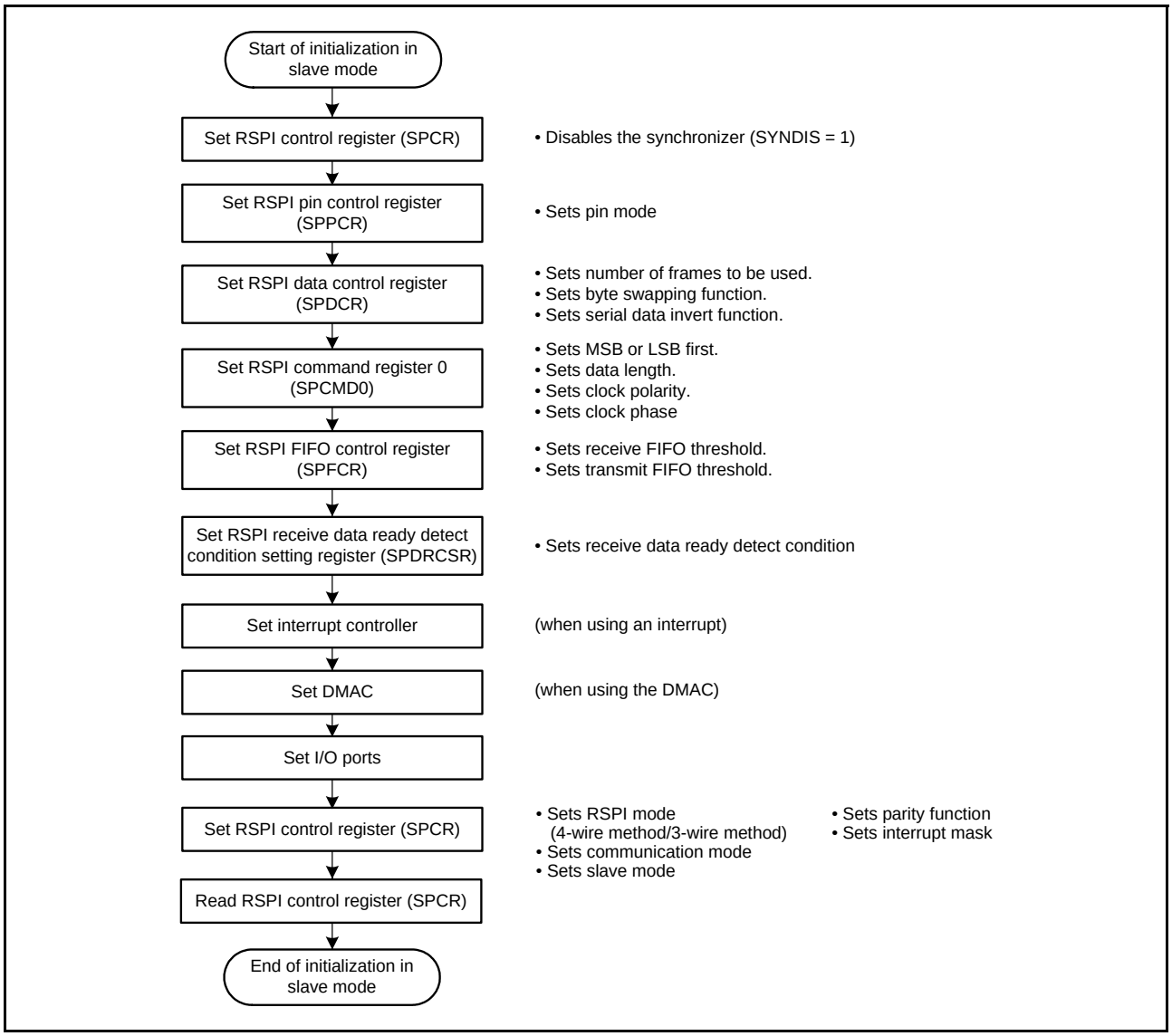


Figure 41.74 Example of Initialization Flowchart in Slave Mode (Clock Synchronous Operation)

(4) Flow of Software Processing

Software processing during clock-synchronous operation when the RSPIA is used in slave mode is the same as that for SPI slave mode operation. For details, refer to section 41.3.13.2, (6) Software Processing Flow. Note that mode fault errors will not occur.

41.3.15 Loopback Mode

When 1 is written to the SPPCR.SPLP2 bit or SPPCR.SPLP bit, the RSPIA shuts off the path between the MISO0 pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSI0 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 RSPIA does not shut off the path between the MOSI0 pin and the shift register if the SPCR.MSTR bit is 1, and between the MISO0 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 RSPIA or the reversed transmit data becomes the received data for the RSPIA.
Table 41.12 lists the relationship among the SPLP2 and SPLP bits and the received data.

Table 41.12 SPLP2 and SPLP Bit Settings and Received Data

SPPCR.SPLP2 Bit	SPPCR.SPLP Bit	Received Data
0	0	Input data from the MOSI0 pin or MISO0 pin
0	1	Inverted transmit data
1	x	Transmit data

x: Don’t care.

Figure 41.75 shows the configuration of the shift register I/O paths in loopback mode.

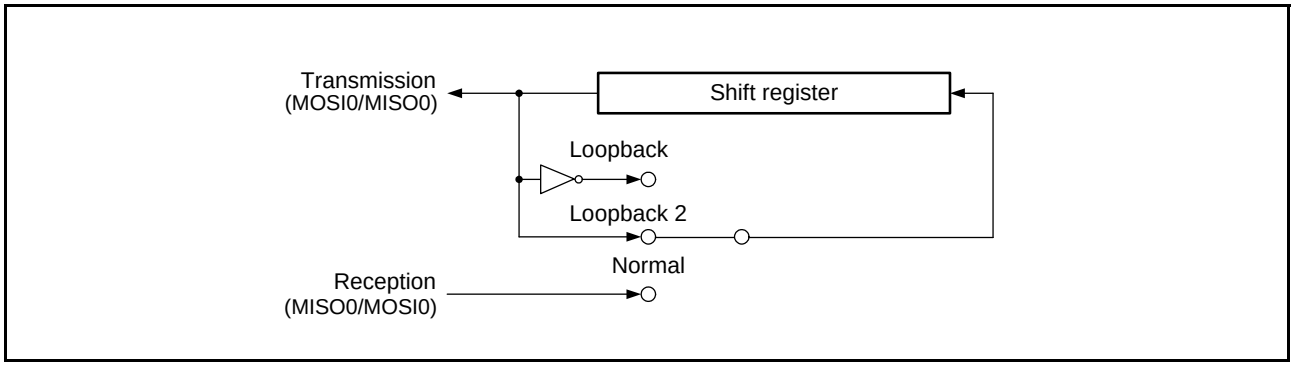


Figure 41.75 Configuration of Shift Register I/O Paths in Loopback Mode

41.3.16 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 41.76.

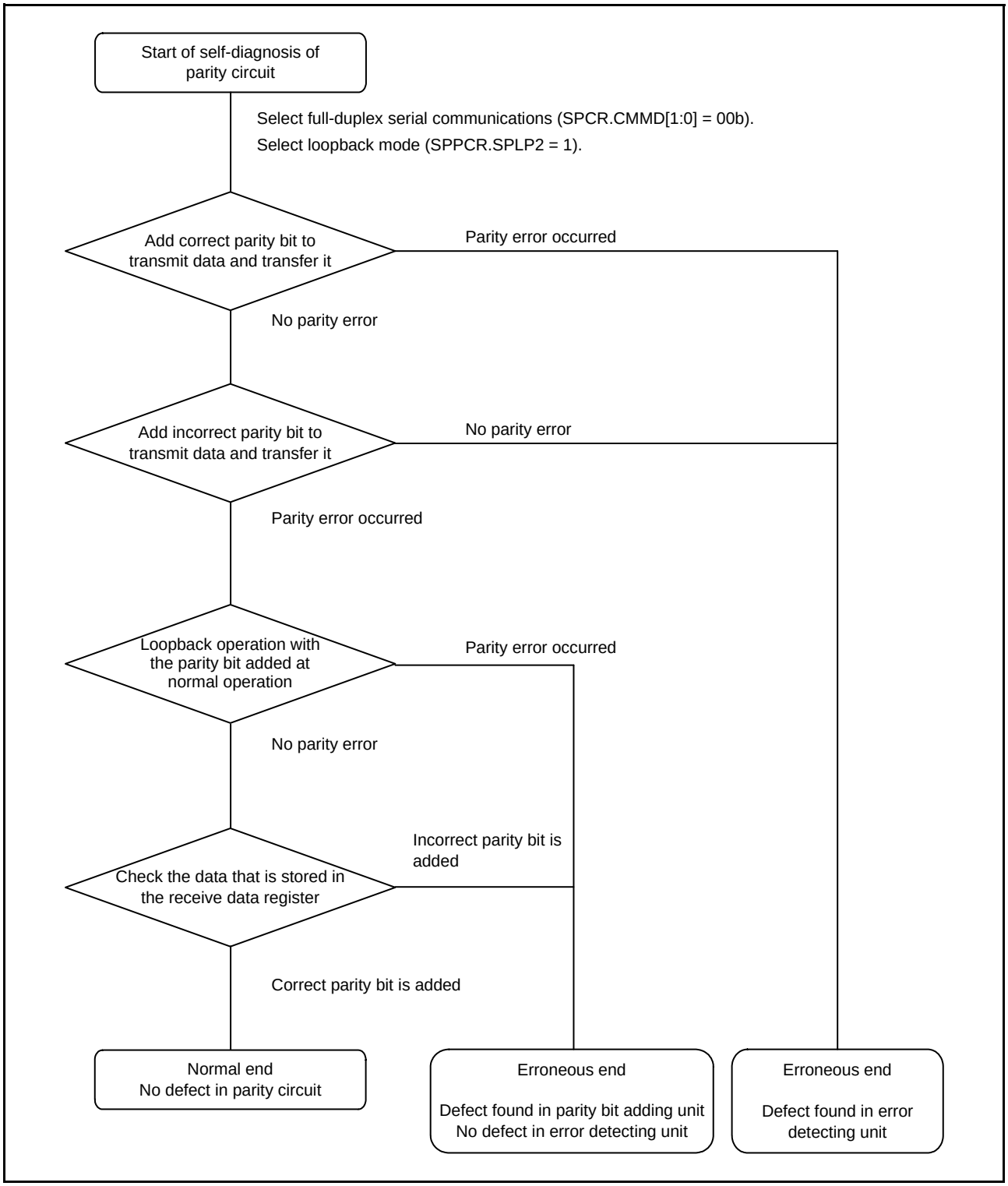


Figure 41.76 Flowchart for Self-Diagnosis of Parity Circuit

41.3.17 Interrupt Sources

The RSPIA has interrupt sources of receive buffer full, transmit buffer empty, error (mode fault, underrun, overrun, and parity error), idle, communication end, and receive data ready. 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, underrun, overrun, parity errors, and receive data ready (only when the RDRIS bit is 1), the actual interrupt source must be determined from the flags.

Interrupt sources for the RSPIA are listed in Table 41.13. An interrupt is generated on satisfaction of an interrupt condition in Table 41.13. 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 RSPIA settings. For the method for setting the DTC or DMAC, refer to section 18, DMA Controller (DMACAb), or section 20, Data Transfer Controller (DTCb).

If the conditions for generating a transmit buffer empty or receive buffer full interrupt are generated while the ICU.IRn.IR flag is 1, the interrupt is not output as a request for ICU but is retained internally (the capacity for retention is one request per source). A retained interrupt request is output when the ICU.IRn.IR flag becomes 0. A retained interrupt request is automatically discarded once it is output as an actual interrupt request. The interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) for an internally retained interrupt request can also be cleared to 0.

Table 41.13 Interrupt Sources of RSPIA

Interrupt Source	Symbol	Interrupt Condition	DMAC/DTC Activation
Receive buffer full Receive data ready	SPRI	The SPSR.SPRF flag is set to 1 while the SPCR.SPRIE bit is 1, or the RRDYF flag is set to 1 while the SPEIE is 1 and RDRIS bit is 0.	Possible
Transmit buffer empty	SPTI	The SPSR.SPTEF flag is set to 1 while the SPCR.SPTIE bit is 1.	Possible
Mode fault error Underrun error Overrun error Parity error Receive data ready	SPEI	The SPSR.MODF, OVRF, or PERF flag is set to 1 while the SPCR.SPEIE bit is 1, or the RRDYF flag is set to 1 while the SPEIE and RDRIS bits are 1.	Impossible
Idle	SPII	The SPSR.IDLNF flag is set to 0 while the SPCR.SPIIE bit is 1.	Impossible
Communication end	SPCI	The SPSR.SPCF flag is set to 1 while the SPCR.SPCIE bit is 1.	Impossible

41.3.18 Link Operation by Event Linking

The RSPIA supports the following event output for the event link controller (ELC). The event link output signal is output regardless of the interrupt enable bit setting.

41.3.18.1 Receive Buffer Full Event Output

This event signal is output when the number of data stored in the receive FIFO is greater than the threshold value, or when the SPCR.RDRIS bit = 0 and after writing to the receive FIFO, and the number of data stored in the receive FIFO is equal to or less than the threshold value, the value set in the SPDRCSR register has elapsed.
A receive buffer full event is output at the timing shown in Figure 41.77.

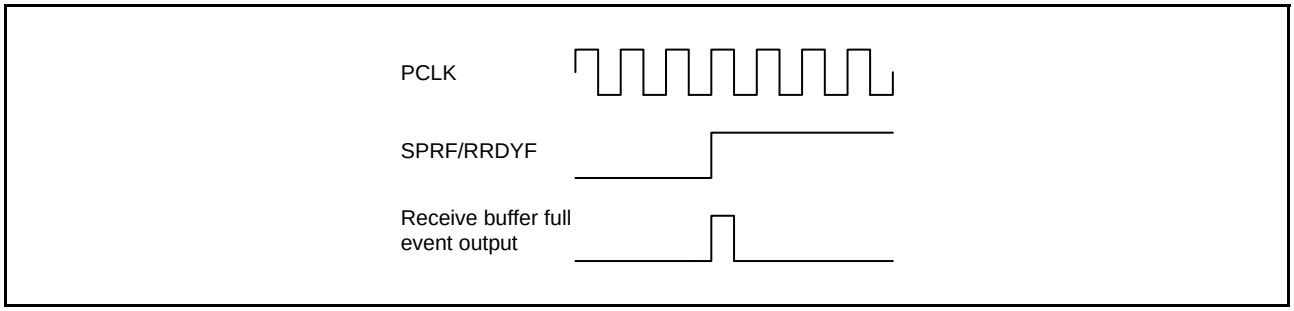


Figure 41.77 Event Output Timing of the Receive Buffer Full

41.3.18.2 Transmit Buffer Empty Event Output

This event signal is output when the number of empty stages in the transmit FIFO is greater than the threshold value or when the value of the SPCR.SPE bit has changed from 0 to 1.
A transmit buffer empty event is output at the timing shown in Figure 41.78.

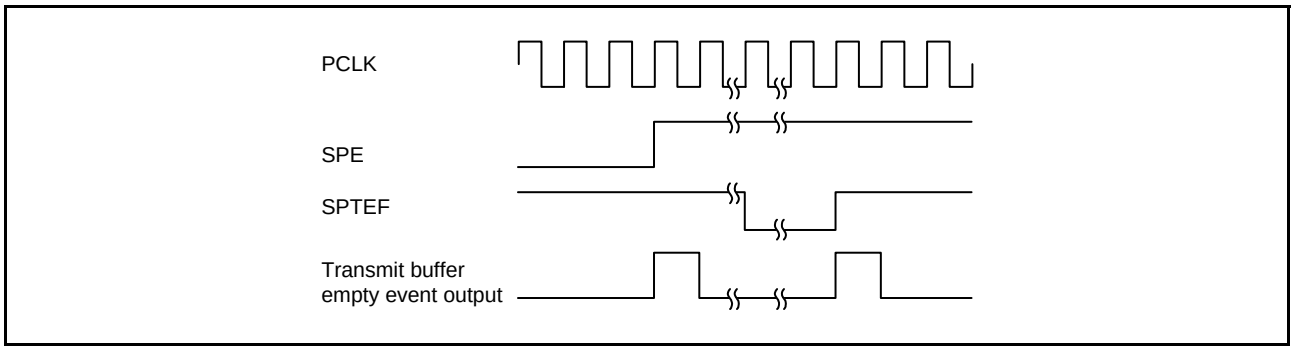


Figure 41.78 Event Output Timing of the Transmit Buffer Empty

41.3.18.3 Error Event Output

The error event signal is output due to five sources (mode fault, underrun, overrun, parity error, and receive data ready). For notes on error event output, see section 41.4.2, Note on Error Events Output.

(1) Mode Fault

A mode fault event occurs when all the following conditions are satisfied:

- SPI operation (SPCR.SPMS = 0)
- Slave mode (SPCR.MSTR = 0)
- The detection of mode fault error is enabled (SPCR.MODFEN = 1)
- Any of the following conditions are satisfied
  1. In the Motorola SPI mode, the SSL00 pin is deactivated during data transfer.
  2. In the TI SSP mode, the SSL00 pin is activated during data transfer.

(2) Underrun

The condition for this event signal being output in response to an underrun is the start of serial transfer while the value of the SPCR.MSTR bit is 0 (slave mode), the value of the SPCR.SPE bit is 1 (enables the RSPI function), and the transmit data are not ready for output.

(3) Overrun

The condition for this event signal being output in response to an overrun is completion of the serial transfer while the value of the SPCR.CMMD[1:0] bits are 00b or 10b, and the data is stored in the receive FIFO for the number of FIFO stages, in which case the SPSR.OVRF flag is set to 1.

(4) Parity Error

The condition for this event signal being output in response to a parity error is detection of a parity error on completion of the serial transfer while the value of the SPCR.CMMD[1:0] bits are 00b or 10b, and the value of the SPCR.SPE bit is 1.

(5) Receive Data Ready

The condition for this event signal being output in response to a receive data ready is elapsing of the value set by the SPDRSCR register when the number of data stored in the receive FIFO is less than or equal to the receive FIFO threshold value after the data has stored in the receive FIFO, while the value of the SPCR.CMMD[1:0] bits are 00b or 10b, the value of the SPCR.RDRIS bit is 1.

(6) Output Timing

Each error event signal is output at the timing shown in Figure 41.79.
Error event is output each time an error occurs.

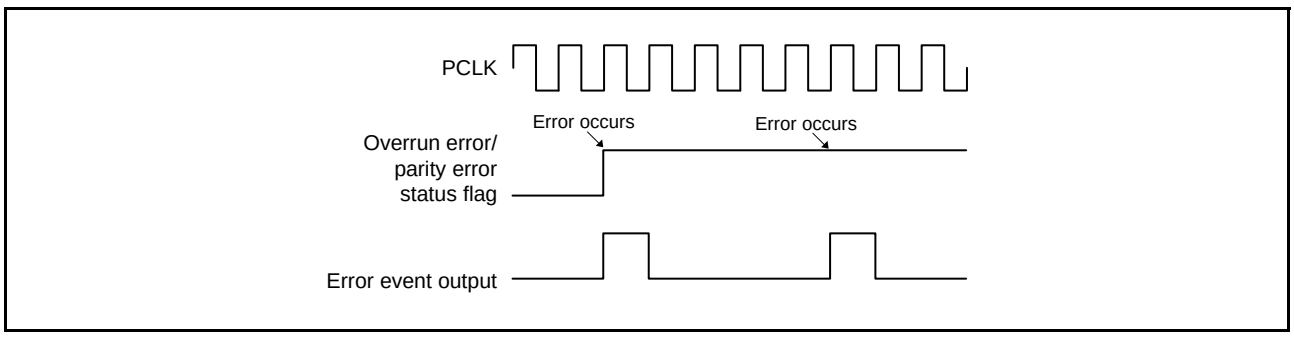


Figure 41.79 Error Event Output Timing

41.3.18.4 Idle Event Output

(1) In Master Mode

[Transmit-receive mode/transmit-only mode]

In transmit-receive master mode or transmit-only master mode, an event is output when the SPSR.IDLNF flag (idle flag) changes from 1 to 0.

The SPSR.IDLNF flag changes from 1 to 0 only when any of the following conditions are satisfied.

(a) The SPCR.SPE bit becomes 0 (RSPI is initialized) during transmission

(b) All the following conditions are satisfied:

- The next transmit data is not set in the transmission buffer (SPTXn, n = 0 to 3)
- The SPSSR.SPCP[2:0] bits are 000b (beginning of sequence control)
- The operations up to the next-access delay have finished (the master main state machine has entered the idle state)

[Receive-only mode]

In receive-only master mode, an event is output when any of the following conditions are satisfied.

(a) The SPCR.SPE bit is set to 0 (RSPI is initialized).

(b) Any the following conditions are satisfied:

- When the SPRMCR.RFC[4:0] bits are 00000b, after 1 is written to the SPRMCR.TERM bit, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)
- When the SPRMCR.RFC[4:0] bits are other than 00000b, after 1 is written to the SPRMCR.TERM bit, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)
- When the SPRMCR.RFC[4:0] bits are other than 00000b, after the operation is completed for the number of received frames set by the SPRMCR.RFC[4:0] bits, the operations up to the next-access delay have finished (the master main state machine has entered the idle state)

(2) In Slave Mode

In slave mode, an event is output when the SPCR.SPE bit is set to 0 (RSPI is initialized).

(3) Output Timing

An idle event is output at the timing shown in Figure 41.80.

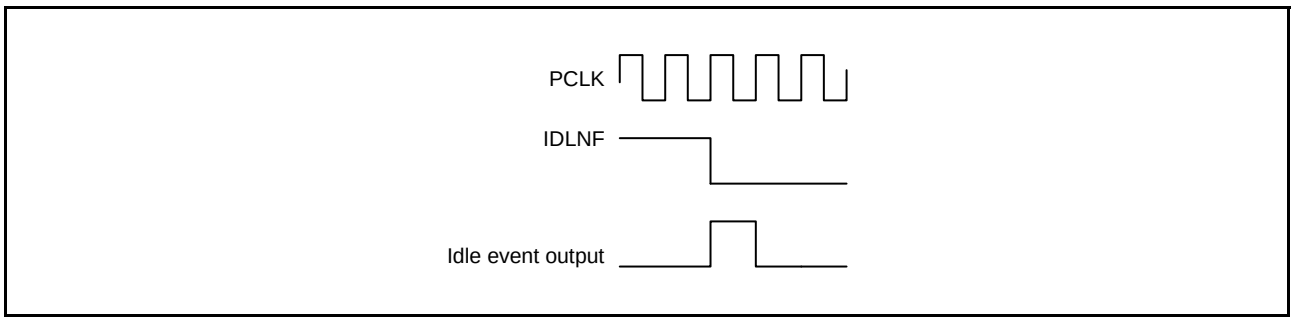


Figure 41.80 Idle Event Output Timing

41.3.18.5 Communication End Event Output

In master mode, an event is output under the condition for setting the SPSR.IDLNF flag (idle flag) from 1 to 0. In slave mode, an event is output under the conditions listed in Table 41.14 and Table 41.15.

Table 41.14 Generating Conditions of Communication End Event (Transmit-Receive Slave Mode/Transmit-Only Slave Mode)

RSPI Mode	Transmit Buffer State	Shift Register State	Others
SPI operation (SPMS = 0, FRFS = 0)	Empty	Empty	Negation of the SSL00 input
SPI operation (SPMS = 0, FRFS = 1)	Empty	Empty	Completion of the SSL negation delay
Clock synchronous operation (SPMS = 1)	Empty	Empty	Even edge detection of the last RSPCK0 for the last data

Table 41.15 Generating Conditions of Communication End Event (Receive-Only Slave Mode)

RSPI Mode	Condition
SPI operation (SPMS = 0, FRFS = 0)	Negation of the SSL00 input after storing the number of frames set by the SPDCR.SPFC[1:0] bits in the receive buffer
SPI operation (SPMS = 0, FRFS = 1)	Completion of the SSL negation delay after storing the number of frames set by the SPDCR.SPFC[1:0] bits in the receive buffer
Clock synchronous operation (SPMS = 1)	Even edge detection of the last RSPCK0 for the last frame set by the SPDCR.SPFC[1:0] bits

Whether the operation is in master mode or slave mode, an event is not output if 0 is written to the SPCR.SPE bit during communication or the SPCR.SPE bit is cleared by the mode fault error or underrun error.
A communication end event is output at timings shown in Figure 41.81 to Figure 41.86. The communication end event output timing in master operation is omitted because it is output at the same timing as an idle event.

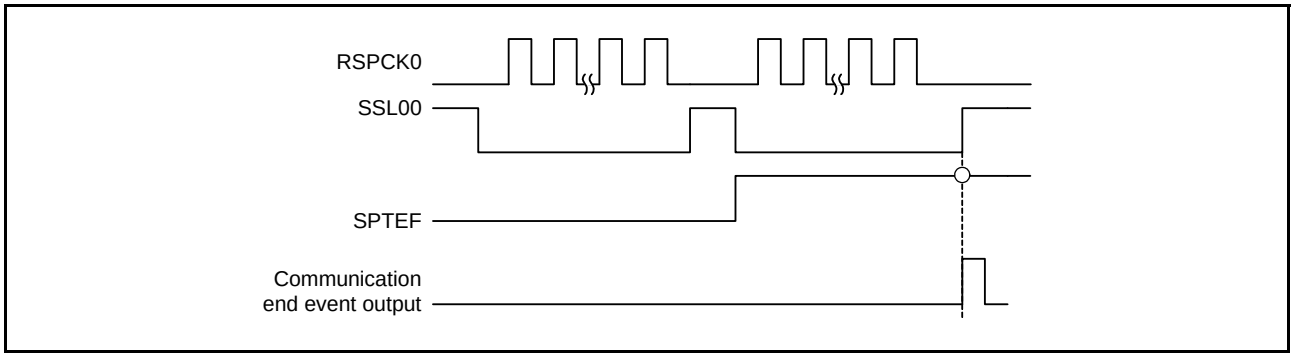


Figure 41.81 Communication End Event Output Timing (Transmit-Receive Slave Mode/Transmit-Only Slave Mode, Motorola SPI Operation)

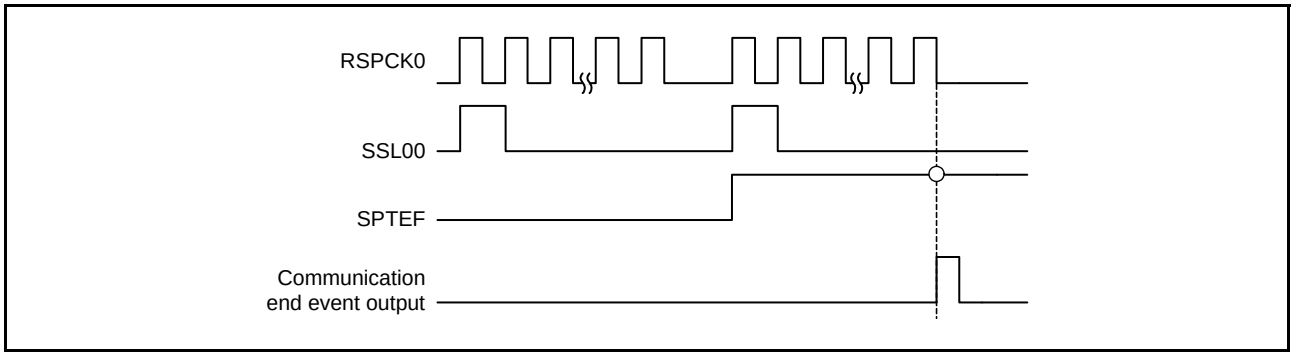


Figure 41.82 Communication End Event Output Timing (Transmit-Receive Slave Mode/Transmit-Only Slave Mode, TI SSP Operation)

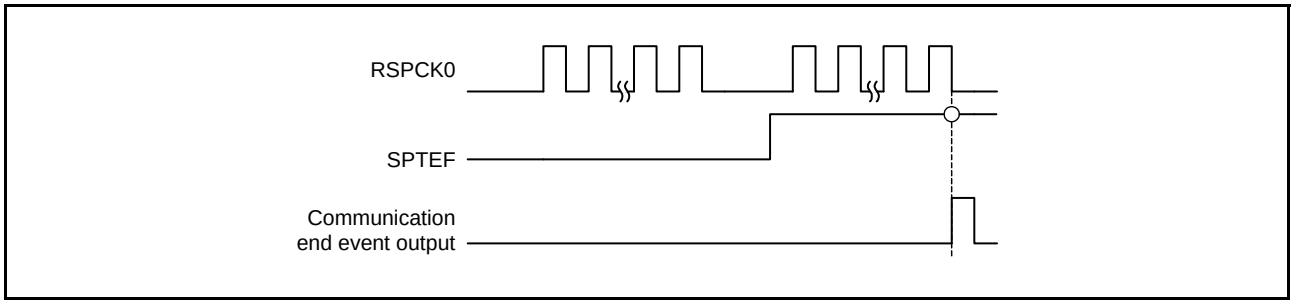


Figure 41.83 Communication End Event Output Timing (Transmit-Receive Slave Mode/Transmit-Only Slave Mode, Clock Synchronous Operation)

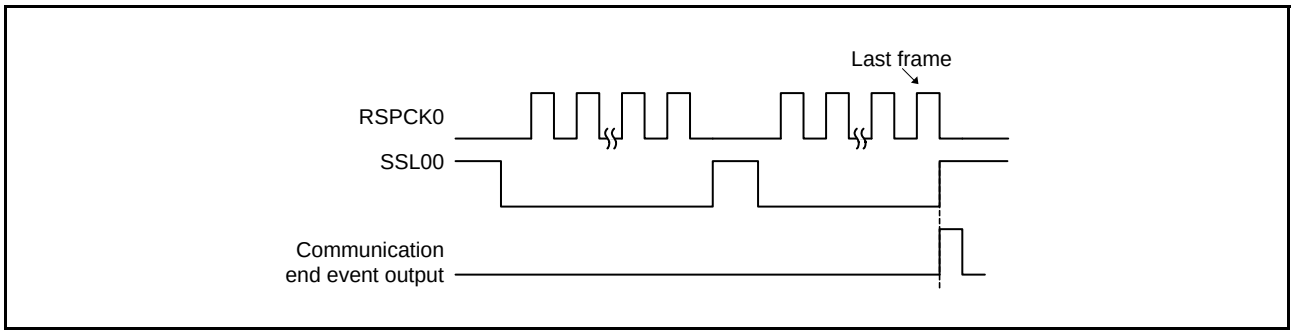


Figure 41.84 Communication End Event Output Timing (Receive-Only Slave Mode, Motorola SPI Operation)

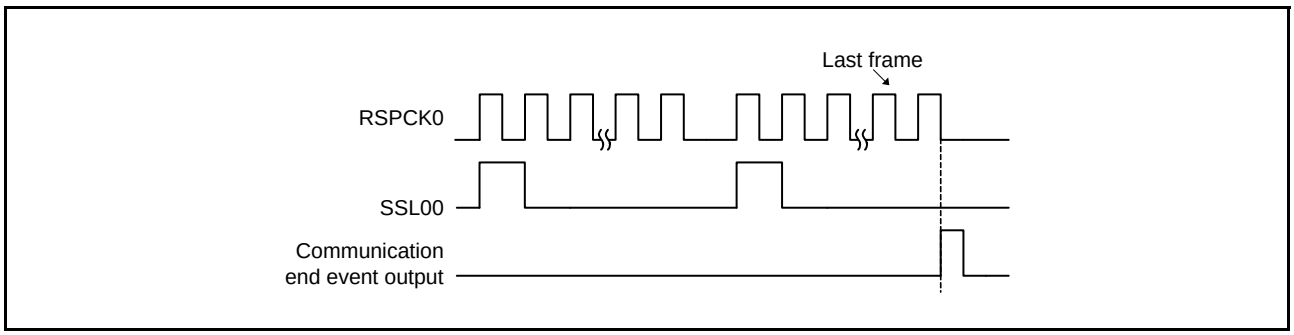


Figure 41.85 Communication End Event Output Timing (Receive-Only Slave Mode, TI SSP Operation)

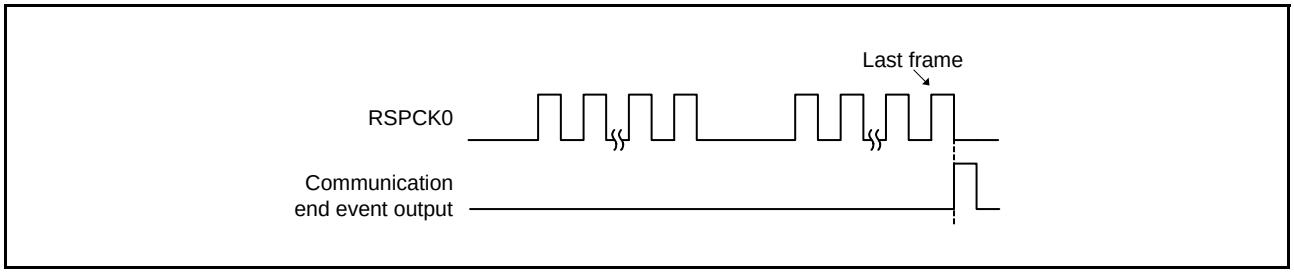


Figure 41.86 Communication End Event Output Timing (Receive-only slave mode, Clock Synchronous Operation)

41.4 Usage Notes

41.4.1 Notes on Starting Transfer

When the ICU.IRn.IR flag is 1 at the time transfer is to start, follow the procedure below to clear interrupt requests before enabling operations (by setting the SPCR.SPE bit to 1).

1. Confirm that transfer has stopped (i.e. that the SPCR.SPE bit is 0).
2. Set the relevant interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) to 0.
3. Read the relevant interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) and confirm that its value is 0.
4. Set the ICU.IRn.IR flag to 0.

41.4.2 Note on Error Events Output

To enable SPI operation of the RSPIA in master mode in the multi-master environment (SPCR.SPMS bit = 0, SPCR.MSTR bit = 1, and SPCR.MODFEN bit = 1), do not use any output of error events.

41.4.3 Note on Low Power Consumption Functions

To reduce power consumption of this module by using the low power consumption functions, set the SPCR.SPE bit to 0 to terminate communication, and then use the low power consumption functions.

41.4.4 Note on the SPRF and SPTEF flags

When polling the SPSR.SPRF flag and/or SPSR.SPTEF flag, set the SPCR.SPRIE bit and/or SPCR.SPTIE bit to 0.

41.4.5 Notes on Burst Transfer in Master Mode

During burst transfer, changing the following settings between the SPCMDm register with the SSLKP bit set to 1 and the SPCMDm register used in the next transfer is prohibited.

- SSL Signal Assertion setting (SSLA[2:0] bits)
- RSPCK output setting (CPHA, CPOL, BRDV[1:0] bits)

41.4.6 Notes on Operating in Slave, TI SSP Mode

In the slave TI SSP mode, the delay between frames must observe the following intervals. Secure the following intervals on the master side.

[Master side: interval from the RSPCK edge of last bit to the RSPCK edge of next SSL] > [Slave side: OE delay time = PCLKA * (1 to 2) + SLNDL[2:0] setting value]

41.4.7 Notes on the Data Length

Table 41.16 shows the relationship between the data length and operation of the RSPIA.

Table 41.16 Relationship between the Data Length and Operation of the RSPIA

Data Length										
4 bit	**5 bit**	**6 bit**	**7 bit**	**8 bit**	**9 bit**	**10 bit**	**11 bit**	**12 bit**	**13 bit**	**14 bit**
Conditionally possible	Conditionally possible	Conditionally possible	Possible	Possible	Possible	Possible	Possible	Possible	Possible	Possible

Possible: Operating possible
Conditionally possible: Operating possible when the SPCR.SCKASE or SCKDDIS bit is 0.

41.4.8 Notes when SPCR.SPE = 1

If the value of the registers listed below are changed while the SPCR.SPE bit is 1, the operation after the change is not guaranteed.

Table 41.17 List of Registers that Cannot Be Rewritten When SPCR.SPE = 1 (1/2)

Register	Bit
SPCKD	SCKDL[2:0]
SSLND	SLNDL[2:0]
SPND	SPNDL[2:0]
SPCR	SYNDIS
	MSTR
	CMMD[1:0]
	FRFS
	SPMS
	MODFEN
	SCKDDIS
	SCKASE
	PTE
	SPOE
	SPPE
SPRMCR	RFC[4:0]
SPDRCSR	—
SPPCR	MOIFE
	MOIFV
	SPLP2
	SPLP
SSLP	SSL3P
	SSL2P
	SSL1P
	SSL0P
SPBR	—
SPSCR	SPSLN[2:0]

Table 41.17 List of Registers that Cannot Be Rewritten When SPCR.SPE = 1 (2/2)

Register	Bit
SPCMD0*1	SSLA[2:0]
	SPB[4:0]
	SCKDEN
	SLNDEN
	SPNDEN
	LSBF
	SSLKP
	BRDV[1:0]
	CPOL
	CPHA
SPDCR	SPFC[1:0]
	DINV
	SPRDTD
	BYSW
SPFCR	TTRG[1:0]
	RTRG[1:0]
SPFCLR	FCLR

Note 1. Rewriting prohibited in slave mode. In master mode, rewriting is available only when there is no next transmit data in the transmit FIFO.

42. Quad-SPI Memory Interface (QSPIX)

The Quad-SPI memory interface (QSPIX) module is a memory controller for connecting serial ROM (nonvolatile memory such as a serial flash memory, serial EEPROM, or serial FeRAM) that has an SPI-compatible interface.

42.1 Overview

Table 42.1 lists the QSPIX specifications, Figure 42.1 shows a block diagram, and Table 42.2 lists the I/O pins.

Table 42.1 QSPIX Specifications

Parameter	Specifications
Number of channels	1 channel
SPI	• Support for Single/Extended SPI, Dual SPI, and Quad SPI protocols • Configurable to SPI mode 0 and SPI mode 3 • Address size selectable to 8, 16, 24, or 32 bits
Timing adjustment function	Timing corresponding to various serial ROMs can be generated.
Memory mapped mode	The data in the serial ROM can be read by accessing the memory space of the MCU. • Support for Read, Fast Read, Fast Read Dual Output, Fast Read Dual I/O, Fast Read Quad Output, and Fast Read Quad I/O instructions • Substitutable instruction code • Adjustable number of dummy cycles • Prefetch function • Polling processing • SPI bus cycle extension function • XIP (execute in place) mode
Indirect access mode	The data in the serial ROM can be accessed by accessing the data register of the QSPIX. • Flexible support for various serial flash instructions and functions including erase, program, ID read, and power-down control by software control
Interrupt source	Error interrupt
Low power consumption function	Module-stop state can be set to reduce power consumption

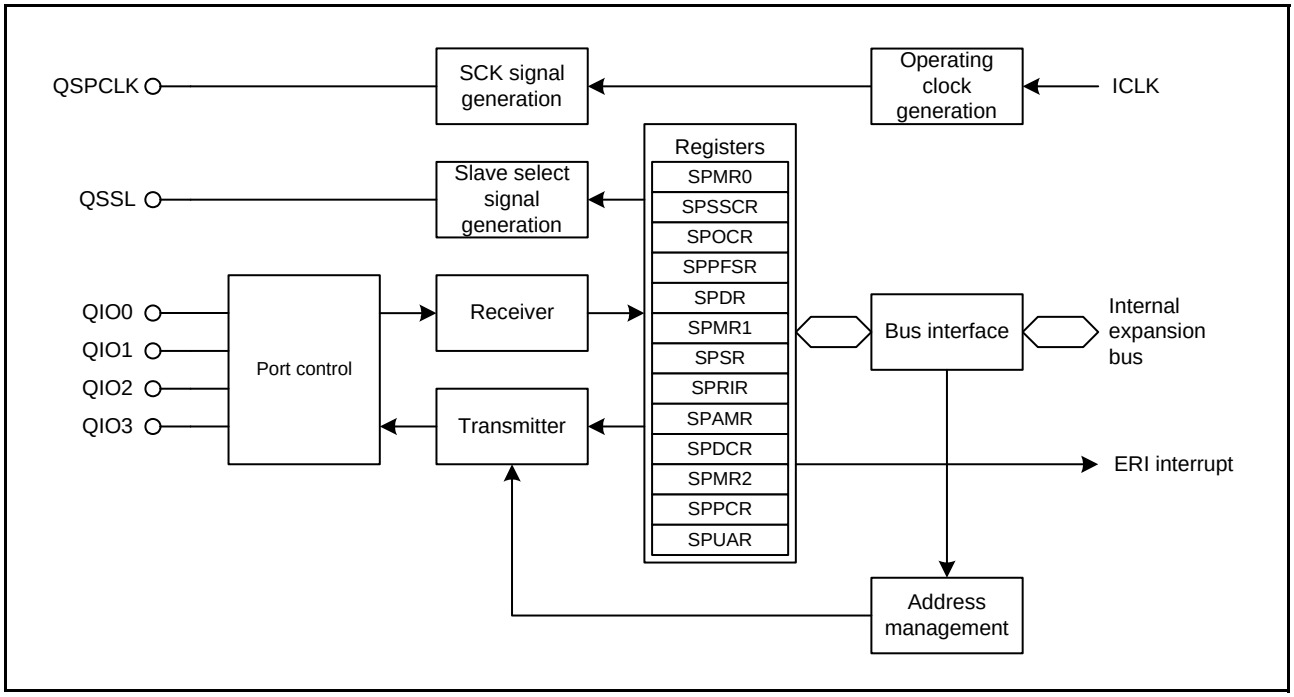


Figure 42.1 QSPIX Block Diagram

Table 42.2 QSPIX I/O Pins

Pin Name	I/O	Description
QSPCLK	Output	QSPIX clock output pin
QSSL	Output	QSPIX slave select pin
QIO0	I/O	Data 0 input/output pin
QIO1	I/O	Data 1 input/output pin
QIO2	I/O	Data 2 input/output pin or WP# signal output
QIO3	I/O	Data 3 input/output pin

42.2 Register Descriptions

42.2.1 Mode Register 0 (SPMR0)

Address(es): QSPIX.SPMR0 7400 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
	SRIS	—	—	—	—	—	DODE	CKMD	—	PFE	SSE[1:0]		—	RISEL[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	RISEL[2:0]	Read Instruction Select	b2 b0 0 0 0: Read (03h*1/13h) 0 0 1: Fast Read (0Bh/0Ch) 0 1 0: Fast Read Dual Output (3Bh/3Ch) 0 1 1: Fast Read Dual I/O (BBh/BCh) 1 0 0: Fast Read Quad Output (6Bh/6Ch) 1 0 1: Fast Read Quad I/O (EBh/ECh) 1 1 0: Setting prohibited 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	SSE[1:0]	Slave Select Extension	b5 b4 0 0: Do not extend QSSL 0 1: Extend QSSL by 33 QSPCLK cycles 1 0: Extend QSSL by 129 QSPCLK cycles 1 1: Extend QSSL until a discontinuous address is detected	R/W
b6	PFE	Prefetch Function Enable	0: Disable function 1: Enable function	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	CKMD	Clock Mode Select	0: SPI mode 0 1: SPI mode 3	R/W
b9	DODE	Data Output Drive Time Extension	0: Do not extend 1: Extend by 1 QSPCLK cycle	R/W
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	SRIS	Special Read Instruction Select	0: Use instruction code selected by RISEL[2:0] bits 1: Use instruction code set in the SPRIR register	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the SPAMR.SIZE[1:0] bits are 00b, code 0Bh may be sent because bit 8 of the address (A8) is assigned to bit 3 of the instruction code.

42.2.2 Slave Select Signal Control Register (SPSSCR)

Address(es): QSPIX.SPSSCR 7400 0004h

	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
	—	—	—	—	—	—	—	—	—	—	SSSU	SSHLD	SSHW[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SSHW[3:0]	Slave Select High Width Setting	b3 b0 0 0 0 0: 1 cycle of operating clock 0 0 0 1: 2 cycles of operating clock 0 0 1 0: 3 cycles of operating clock 0 0 1 1: 4 cycles of operating clock 0 1 0 0: 5 cycles of operating clock 0 1 0 1: 6 cycles of operating clock 0 1 1 0: 7 cycles of operating clock 0 1 1 1: 8 cycles of operating clock 1 0 0 0: 9 cycles of operating clock 1 0 0 1: 10 cycles of operating clock 1 0 1 0: 11 cycles of operating clock 1 0 1 1: 12 cycles of operating clock 1 1 0 0: 13 cycles of operating clock 1 1 0 1: 14 cycles of operating clock 1 1 1 0: 15 cycles of operating clock 1 1 1 1: 16 cycles of operating clock	R/W
b4	SSHLD	Slave Select Hold Time Setting	0: Negate QSSL 0.5 cycles of operating clock after the last rising edge of QSPCLK 1: Negate QSSL 1.5 cycles of operating clock after the last rising edge of QSPCLK	R/W
b5	SSSU	Slave Select Setup Time Setting	0: Assert QSSL 0.5 cycles of operating clock before the first rising edge of QSPCLK 1: Assert QSSL 1.5 cycles of operating clock before the first rising edge of QSPCLK	R/W
b31 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to configure the timing of the slave select (QSSL) signal.

42.2.3 Clock Control Register (SPOCR)

Address(es): QSPIX.SPOCR 7400 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
	—	—	—	—	—	—	—	—	—	—	DUTY	DIV[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	DIV[4:0]	Clock Divisor Select	b4 b0 0 0 0 0 0: ICLK divided by 2 0 0 0 0 1: ICLK divided by 3*1 0 0 0 1 0: ICLK divided by 4 0 0 0 1 1: ICLK divided by 5*1 0 0 1 0 0: ICLK divided by 6 0 0 1 0 1: ICLK divided by 7*1 0 0 1 1 0: ICLK divided by 8 0 0 1 1 1: ICLK divided by 9*1 0 1 0 0 0: ICLK divided by 10 0 1 0 0 1: ICLK divided by 11*1 0 1 0 1 0: ICLK divided by 12 0 1 0 1 1: ICLK divided by 13*1 0 1 1 0 0: ICLK divided by 14 0 1 1 0 1: ICLK divided by 15*1 0 1 1 1 0: ICLK divided by 16 0 1 1 1 1: ICLK divided by 17*1 1 0 0 0 0: ICLK divided by 18 1 0 0 0 1: ICLK divided by 20 1 0 0 1 0: ICLK divided by 22 1 0 0 1 1: ICLK divided by 24 1 0 1 0 0: ICLK divided by 26 1 0 1 0 1: ICLK divided by 28 1 0 1 1 0: ICLK divided by 30 1 0 1 1 1: ICLK divided by 32 1 1 0 0 0: ICLK divided by 34 1 1 0 0 1: ICLK divided by 36 1 1 0 1 0: ICLK divided by 38 1 1 0 1 1: ICLK divided by 40 1 1 1 0 0: ICLK divided by 42 1 1 1 0 1: ICLK divided by 44 1 1 1 1 0: ICLK divided by 46 1 1 1 1 1: ICLK divided by 48	R/W
b5	DUTY	Duty Cycle Correction	0: Make no correction 1: Correct the duty cycle when the divisor is odd number	R/W
b31 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Set the DUTY bit to 1

This register is used to configure the frequency and duty ratio of the operating clock (= QSPCLK).

42.2.4 Prefetch Status Register (SPPFSR)

Address(es): QSPIX.SPPFSR 7400 000Ch

	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
	—	—	—	—	—	—	—	—	PFOSF	PBFUL	—	PBLVL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PBLVL[4:0]	Prefetch Buffer Fill Level	b4 b0 0 0 0 0 0: 0 bytes 0 0 0 0 1: 1 byte 0 0 0 1 0: 2 bytes 0 0 0 1 1: 3 bytes 0 0 1 0 0: 4 bytes 0 0 1 0 1: 5 bytes 0 0 1 1 0: 6 bytes 0 0 1 1 1: 7 bytes 0 1 0 0 0: 8 bytes 0 1 0 0 1: 9 bytes 0 1 0 1 0: 10 bytes 0 1 0 1 1: 11 bytes 0 1 1 0 0: 12 bytes 0 1 1 0 1: 13 bytes 0 1 1 1 0: 14 bytes 0 1 1 1 1: 15 bytes 1 0 0 0 0: 16 bytes 1 0 0 0 1: 17 bytes 1 0 0 1 0: 18 bytes	R
b5	—	Reserved	This bit is read as 0.	R
b6	PBFUL	Prefetch Buffer Full Flag	0: Prefetch buffer has free space 1: Prefetch buffer is full	R
b7	PFOSF	Prefetch Function Operating Status Flag	0: Prefetch function operating 1: Prefetch function not enabled or not operating	R
b31 to b8	—	Reserved	These bits are read as 0.	R

42.2.5 SPI Data Register (SPDR)

Address(es): QSPIX.SPDR 7400 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
	—	—	—	—	—	—	—	—								
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
b7 to b0	—	—	Write to and read from this register is converted to an SPI bus cycle. This register is only accessible in indirect access mode (SPMR1.AMOD bit is 1). Access to this register is ignored in the memory mapped mode.	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

42.2.6 Mode Register 1 (SPMR1)

Address(es): QSPIX.SPMR1 7400 0014h

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AMOD
Value after 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	AMOD	Flash Memory Access Mode Select	0: Memory mapped mode 1: Indirect access mode*1	R/W
b31 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Overwrite this bit with 1 when terminating the SPI bus cycle in indirect access mode.

42.2.7 SPI Status Register (SPSR)

Address(es): QSPIX.SPSR 7400 0018h

	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
	—	—	—	—	—	—	—	—	ROMAE	—	—	—	—	—	—	BUSY
Value after 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	BUSY	Bus Busy Flag	0: No serial transfer being processed 1: Serial transfer being processed	R
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ROMAE	ROM Access Error Flag	0: ROM access error not detected 1: ROM access error detected	R/(W)*1
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to this flag.

This register is used to monitor the SPI bus status in the indirect access mode.

42.2.8 Special Read Instruction Setting Register (SPRIR)

Address(es): QSPIX.SPRIR 7400 0020h

	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
b7 to b0	—	—	When specifying a instruction code generated in memory mapped mode, set the instruction code to this field and set the SPMR0.SRIS bit to 1.	R/W
b31 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

42.2.9 Address Mode Register (SPAMR)

Address(es): QSPIX.SPAMR 7400 0024h

	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
	—	—	—	—	—	—	—	—	—	—	—	I4AE	—	—	SIZE[1:0]	
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
b1, b0	SIZE[1:0]	Address Size Setting	b1 b0 0 0: 1 byte (8-bit address) 0 1: 2 bytes (16-bit address) 1 0: 3 bytes (24-bit address) 1 1: 4 bytes (32-bit address)	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	I4AE	Instruction with 4-Byte Address Enable	0: Do not use 4-byte address instruction 1: Use 4-byte address instruction	R/W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

42.2.10 Dummy Cycle Control Register (SPDCR)

Address(es): QSPIX.SPDCR 7400 0028h

	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
	MODE[7:0]								XIPE	XIPS	—	—	DCYC[3:0]			
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DCYC[3:0]	Number of Dummy Cycle	b3 b0 0 0 0 0: Default dummy cycles for each instruction: - Read: 0 cycles (no dummy cycles) - Fast Read: 8 cycles - Fast Read Dual Output: 8 cycles - Fast Read Dual I/O: 4 cycles - Fast Read Quad Output: 8 cycles - Fast Read Quad I/O: 6 cycles 0 0 0 1: 3 cycles*1 0 0 1 0: 4 cycles 0 0 1 1: 5 cycles 0 1 0 0: 6 cycles 0 1 0 1: 7 cycles 0 1 1 0: 8 cycles 0 1 1 1: 9 cycles 1 0 0 0: 10 cycles 1 0 0 1: 11 cycles 1 0 1 0: 12 cycles 1 0 1 1: 13 cycles 1 1 0 0: 14 cycles 1 1 0 1: 15 cycles 1 1 1 0: 16 cycles 1 1 1 1: 17 cycles	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	XIPS	XIP Status Flag	0: Standard read mode 1: XIP mode	R
b7	XIPE	XIP Enable	0: Disable XIP mode 1: Enable XIP mode	R/W
b15 to b8	MODE[7:0]	Mode Data	Set the data to be output during the first dummy cycle in XIP mode.	R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Do not select this combination when the data output drive time is extended by setting the SPMR0.DODE bit to 1.

42.2.11 Mode Register 2 (SPMR2)

Address(es): QSPIX.SPMR2 7400 0030h

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IOMOD[1:0]	
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
b1, b0	IOMOD[1:0]	I/O Mode Select	b1 b0 0 0: Single/Extended SPI protocol 0 1: Dual SPI protocol 1 0: Quad SPI protocol 1 1: Setting prohibited	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

42.2.12 Port Control Register (SPPCR)

Address(es): QSPIX.SPPCR 7400 0034h

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	WP	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b2	WP	WP Pin Control	0: Low level 1: High level	R/W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

42.2.13 Upper Address Register (SPUAR)

Address(es): QSPIX.SPUAR 7400 0804h

	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
b25 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31 to b26	—	—	Set the missing upper 6 bits for the 26-bit offset address of the SPI space access window. These bits are effective in memory mapped mode.	R/W

42.3 Memory Map

Figure 42.2 shows the memory map in the QSPI area and the relation to the SPI space.

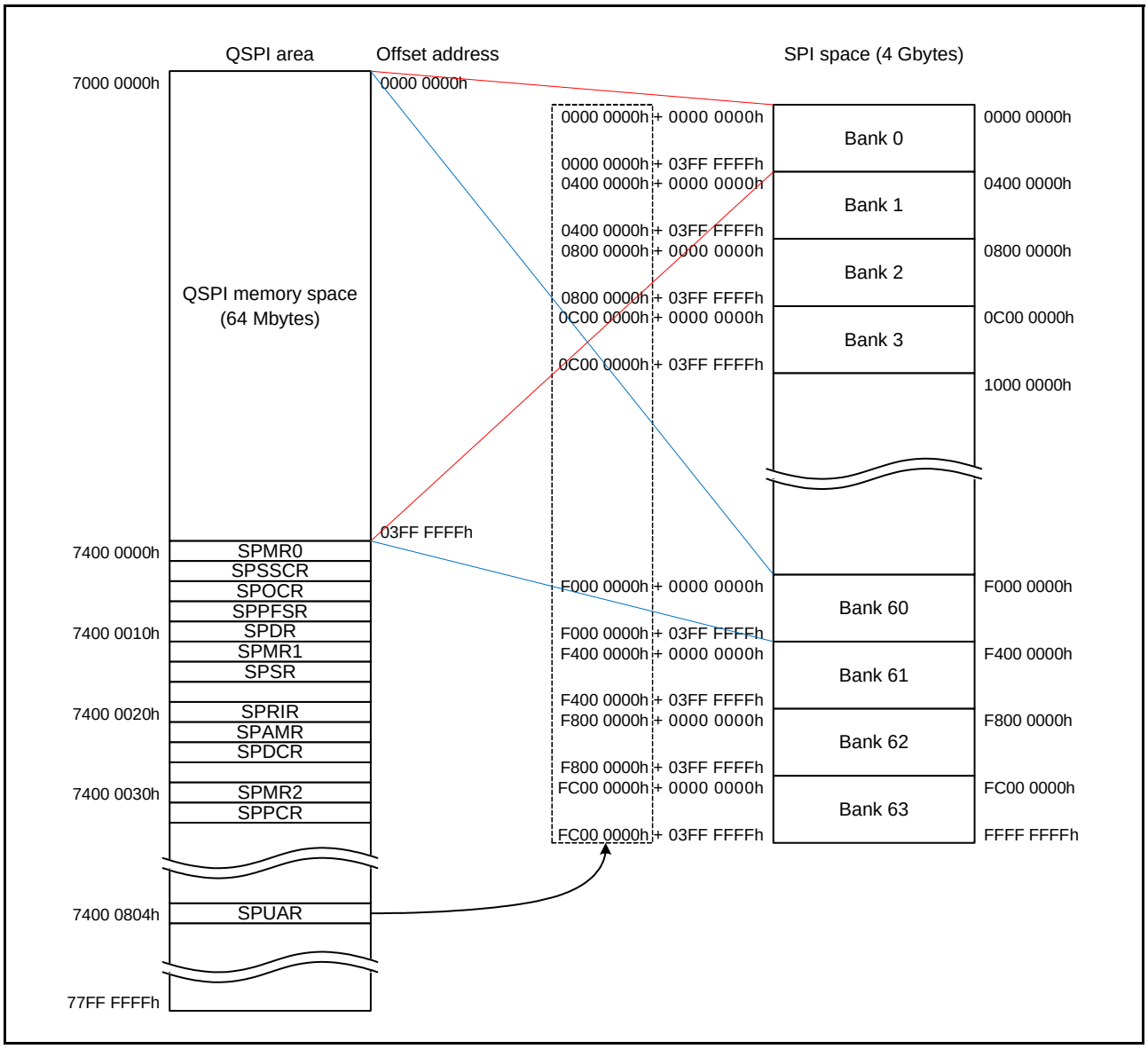


Figure 42.2 Memory Map in the QSPI Area and Relation to the SPI Space

When an address in the QSPI memory space is accessed for a read, an SPI bus cycle is automatically generated and data read from the serial ROM is returned.

The address to access the SPI space is generated from the upper 6 bits of the SPUAR register and the offset of the QSPI memory space. The address for the SPI space generated in this way has 32 bits, but the address sent to the SPI bus is selected from 8, 16, 24, and 32 bits by the SPAMR.SIZE[1:0] bits. When 8, 16, or 24 bits are selected, the lower bits of the address is sent to the SPI bus.

42.4 SPI Bus

42.4.1 SPI Protocol

The QSPIX supports the Single/Extended SPI, Dual SPI, and Quad SPI protocols. The initial state is Single/Extended SPI protocol. To change the protocol, set the SPMR2.IOMOD[1:0] bits.

42.4.1.1 Single/Extended SPI Protocol

The Single/Extended SPI protocol uses only the QIO0 pin to send instruction codes. Subsequent address and data I/O operations are performed using one to four pins, depending on the instruction code (setting of the SPMR0.RISEL[2:0] bits).

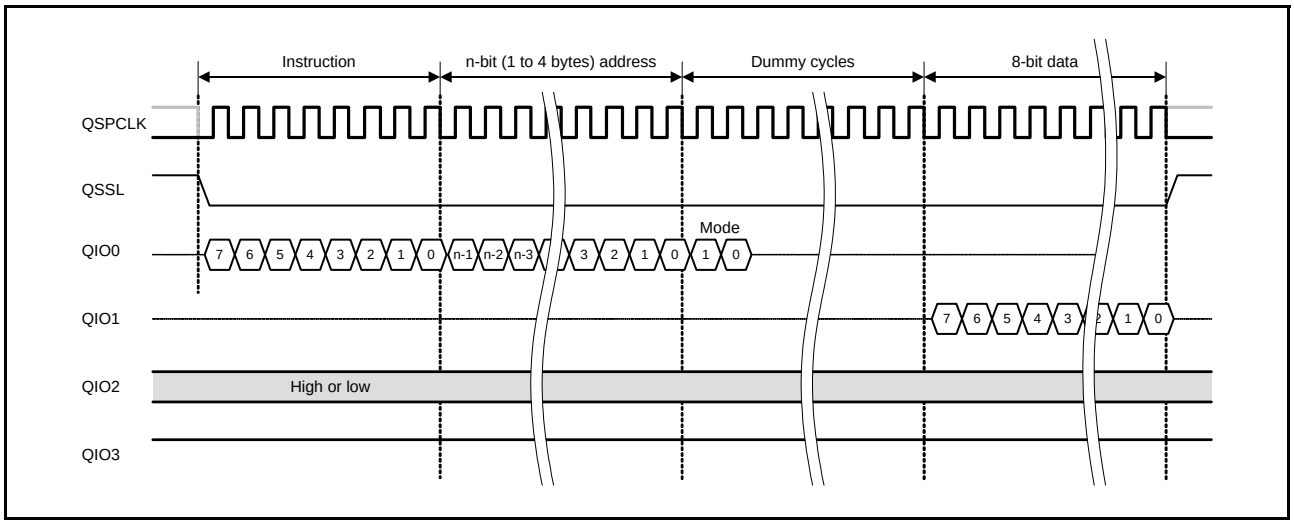


Figure 42.3 Single SPI Protocol Example for Fast Read

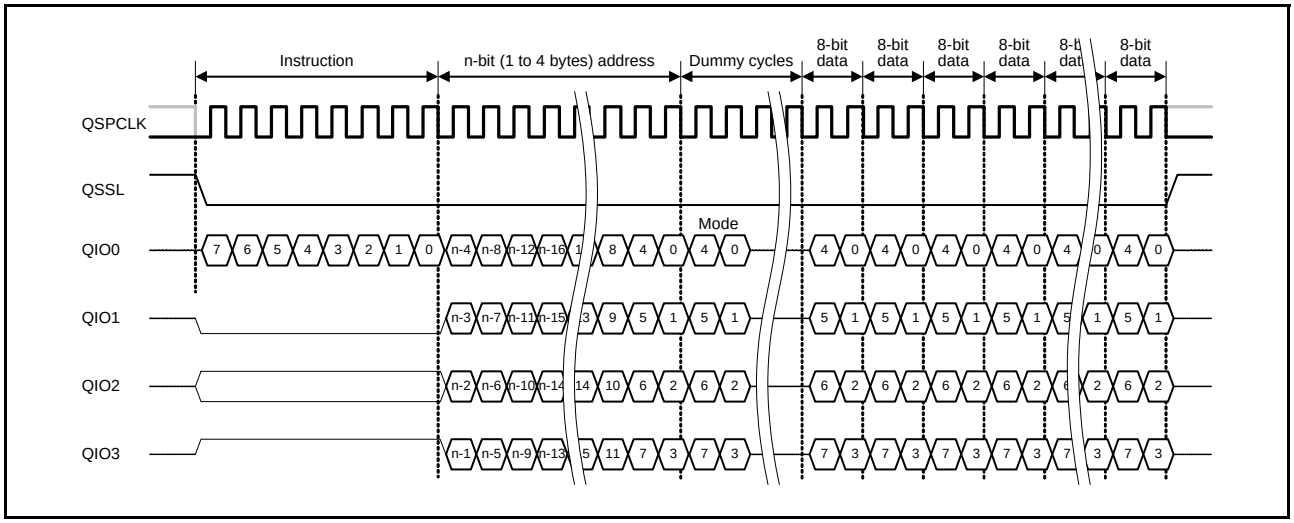


Figure 42.4 Extended SPI Protocol Example for Fast Read Quad I/O

42.4.1.2 Dual SPI Protocol

The Dual SPI protocol uses two pins, QIO0 and QIO1, to perform I/O operations for all signals such as instruction codes, addresses, and data.

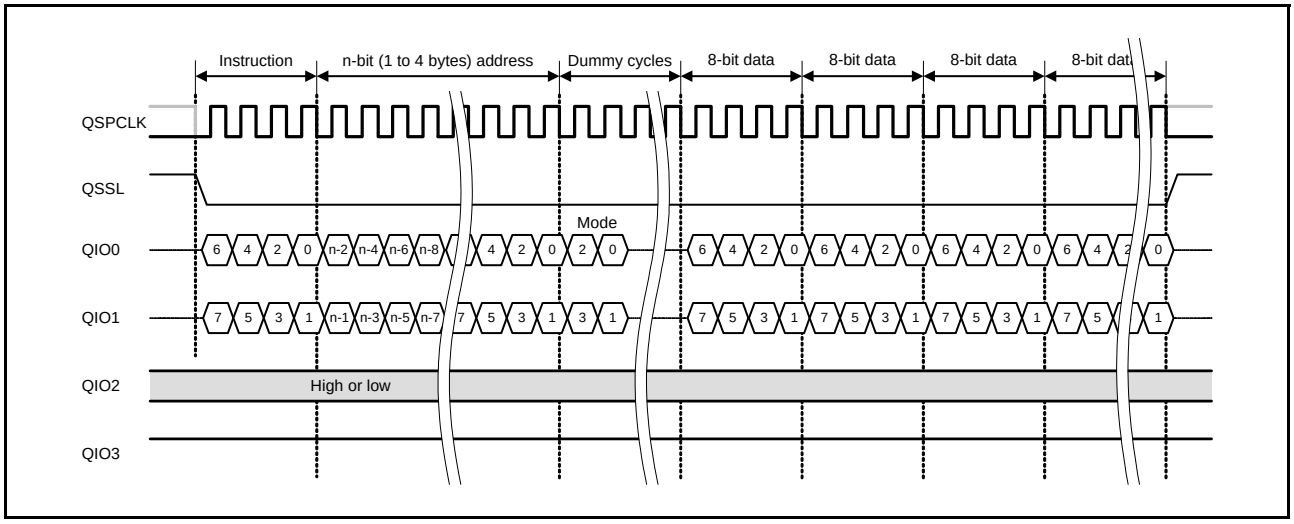


Figure 42.5 Dual SPI Protocol Example for Fast Read

42.4.1.3 Quad SPI Protocol

The Quad SPI protocol uses four pins, QIO0, QIO1, QIO2, and QIO3, to perform I/O operations for all signals such as instruction codes, addresses, and data.

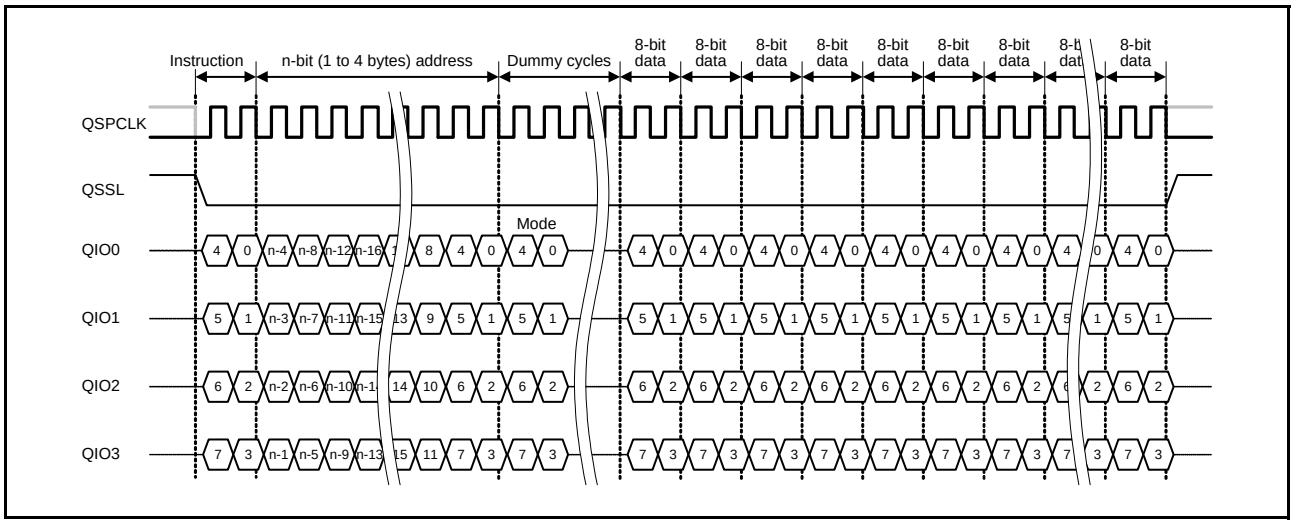


Figure 42.6 Quad SPI Protocol Example for Fast Read

42.4.2 SPI Mode

The SPI mode can be selected from either SPI mode 0 or SPI mode 3 by the SPMR0.CKMD bit. This can be switched by changing the register setting during operation. The difference between SPI modes 0 and SPI mode 3 is the level when the QSPCLK signal is stopped, which is low in SPI mode 0 and high in SPI mode 3.
Transmit data is output from the MCU at the falling edge of QSPCLK and read into the serial ROM at the rising edge of QSPCLK. Receive data is output from the serial ROM at the falling edge of QSPCLK and read into the MCU at the next falling edge of QSPCLK.

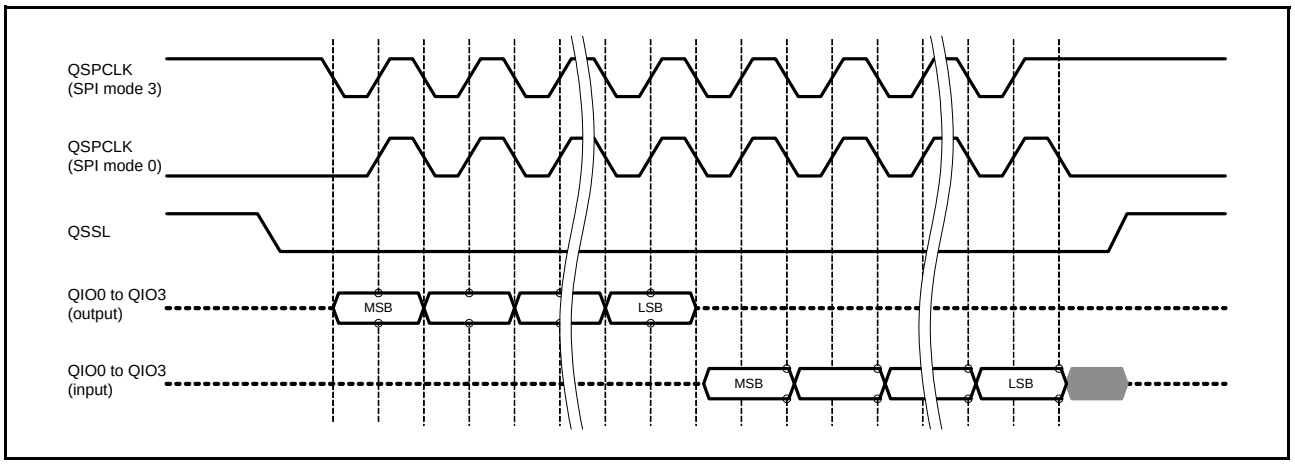


Figure 42.7 Basic Serial Interface Timing

42.5 SPI Bus Timing Adjustment

Various signal timings on the SPI bus can be adjusted by register settings. The configured timing applies to all SPI bus cycles, regardless of memory mapped or indirect access mode.

42.5.1 SPI Bus Operating Clock

The SPI bus operates according to the operating clock obtained by dividing ICLK. The operating clock frequency can be selected in the range of ICLK divided by 2 to 48 by the SPOCR.DIV[4:0] bits.

Table 42.3 Relationship among DIV[4:0] Bits, Frequency Divisor, and Operating Clock Frequencies when ICLK is 120 MHz

DIV[4:0]	Frequency divisor	Operating clock frequency (MHz)
11111b	48	2.50
11110b	46	2.61
11101b	44	2.73
11100b	42	2.86
11011b	40	3.00
11010b	38	3.16
11001b	36	3.33
11000b	34	3.53
10111b	32	3.75
10110b	30	4.00
10101b	28	4.29
10100b	26	4.62
10011b	24	5.00
10010b	22	5.45
10001b	20	6.00
10000b	18	6.67
01111b	17	7.06
01110b	16	7.50
01101b	15	8.00
01100b	14	8.57
01011b	13	9.23
01010b	12	10.00
01001b	11	10.91
01000b	10	12.00
00111b	9	13.33
00110b	8	15.00
00101b	7	17.14
00100b	6	20.00
00011b	5	24.00
00010b	4	30.00
00001b	3	40.00
00000b	2	60.00

42.5.2 QSPCLK Signal Duty Ratio

When the operating clock is configured as ICLK divided by an even number, the high and low widths of the QSPCLK signal match. However when it is configured as ICLK divided by an odd number, the high width is one ICLK cycle longer than the low width.

To bring the duty ratio of the QSPCLK signal closer to 50% when the operating clock is configured as ICLK divided by an odd number, set the SPOCR.DUTY bit to 1 to delay the rising edge of the QSPCLK output signal by 0.5 ICLK cycles. When the operating clock is configured as ICLK divided by an even number, the setting of the SPOCR.DUTY bit is ignored.

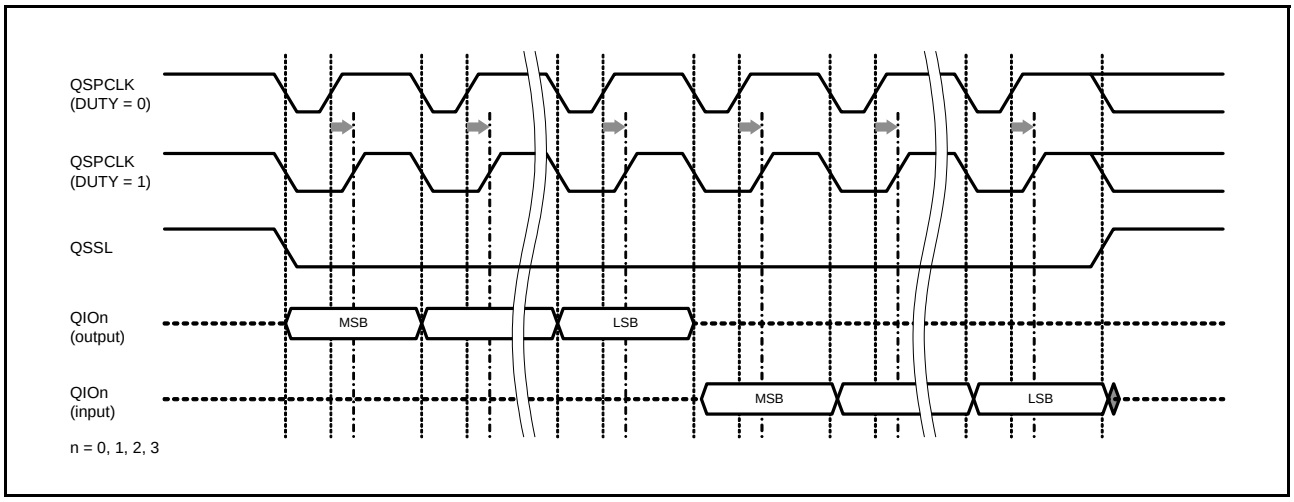


Figure 42.8 Example of Correction of the QSPCLK Signal Duty Ratio Using the DUTY Bit, When ICLK Is Divided by 3

42.5.3 Minimum High Width of the QSSL Signal

Between adjacent SPI bus cycles, the QSSL signal must be held high (inactive) for a sufficient time to ensure the deselect time required for serial ROM. The minimum high width of the QSSL signal can be selected from 1 to 16 cycles of the operating clock by the SPSSCR.SSHW[3:0] bits.

42.5.4 QSSL Signal Setup Time

The time from when the QSSL signal goes low to the first rising edge of QSPCLK signal (QSSL signal setup time) can be configured to satisfy the serial ROM requirements. The setup time can be selected from 0.5 or 1.5 cycles of the operating clock with the SPSSCR.SSSU bit.

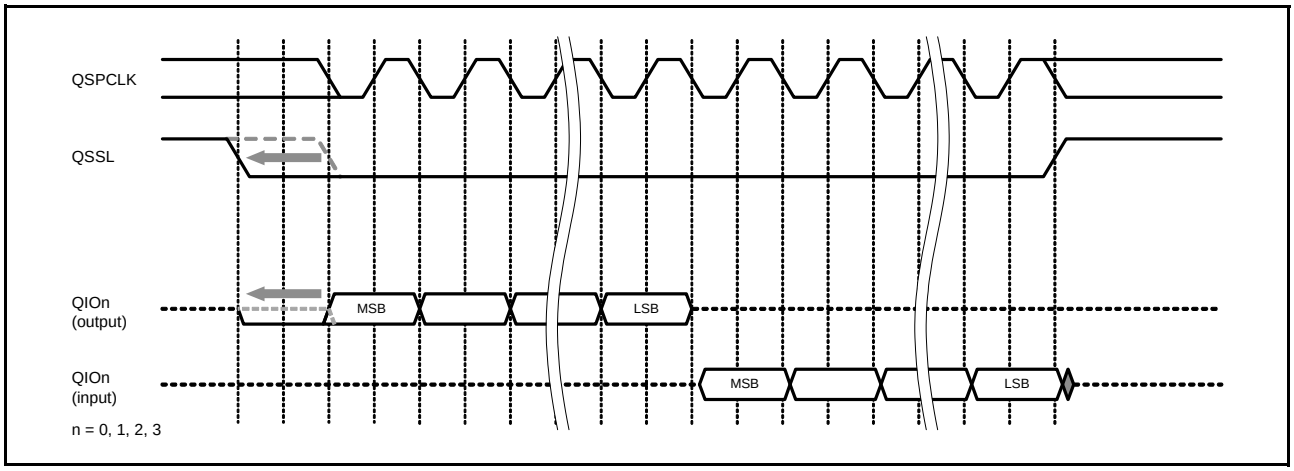


Figure 42.9 Setup Time Adjustment for the QSSL Signal Using the SSSU Bit

42.5.5 QSSL Signal Hold Time

The time from the last rising edge of QSPCLK signal until the QSSL signal goes high (QSSL signal hold time) can be configured to satisfy the serial ROM requirements. The hold time can be selected from 0.5 or 1.5 cycles of the operating clock with the SPSSCR.SSHLD bit.

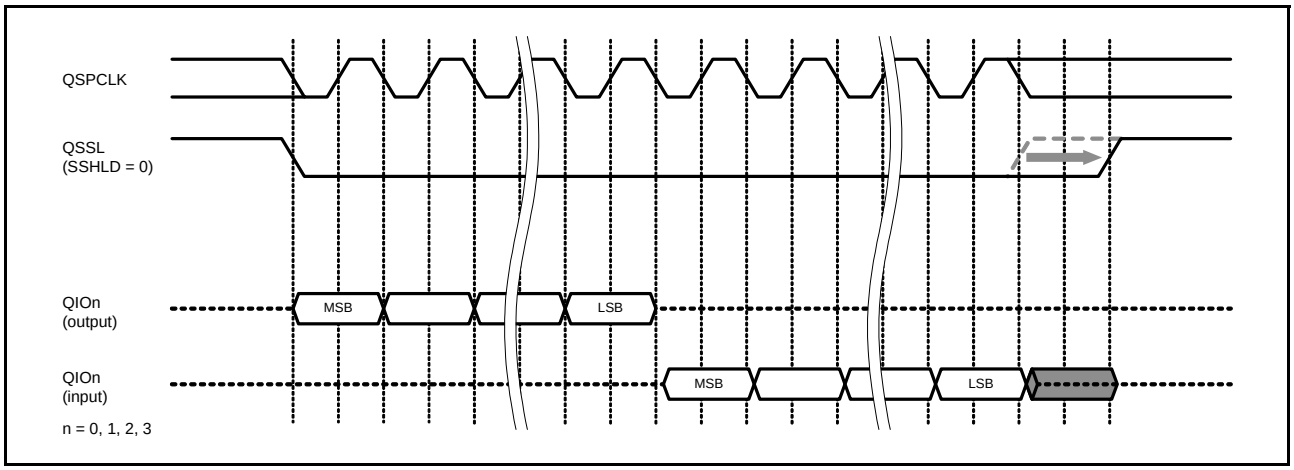


Figure 42.10 Hold Time Adjustment for the QSSL Signal Using the SSHLD Bit

42.5.6 Serial Data Output Drive Time

The output drive time of the QIO0 to QIO3 pins can be extended by one cycle of QSPCLK using the SPMR0.DODE bit. When the DODE bit is set to 1, the first cycle after the address output for the Read instruction and the third cycle of the dummy cycles for the other read instructions do not become high impedance.

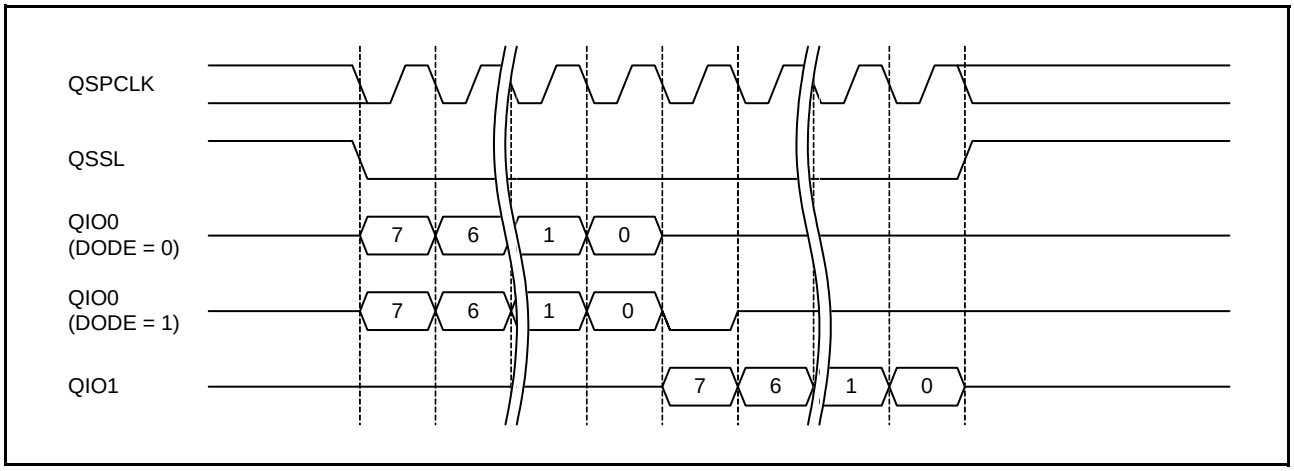


Figure 42.11 Data Output Drive Time Adjustment Using the DODE Bit (1) (Read)

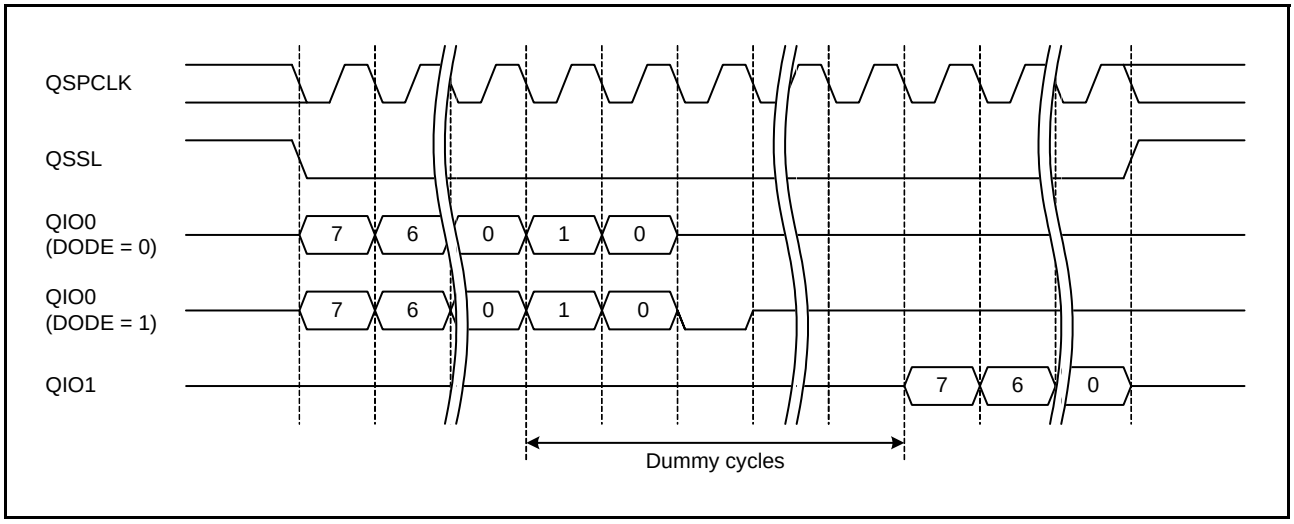


Figure 42.12 Data Output Drive Time Adjustment Using the DODE Bit (2) (Fast Read)

42.6 SPI Instruction Set Used for ROM Access

42.6.1 SPI Instructions That Are Automatically Generated

When QSPIX accesses the serial ROM, an SPI bus cycle using one of the instructions listed in Table 42.4 to Table 42.8 is automatically generated based on the settings in the SPAMR and SPMR0 registers. If the SPMR0.SRIS bit is 1, the setting value of the SPRIR register is used as the instruction code.

Table 42.4 SPI Instructions Automatically Generated When SPAMR.SIZE[1:0] = 00b

SPI Instruction	Instruction code	Address bytes	Dummy cycles	Data bytes	Remarks
Read	03h	1	—	1 to ∞	RISEL[2:0] = 000b, A8 = 0
	0Bh	1	—	1 to ∞	RISEL[2:0] = 000b, A8 = 1

Table 42.5 SPI Instructions Automatically Generated When SPAMR.SIZE[1:0] = 01b

SPI Instruction	Instruction code	Address bytes	Dummy cycles	Data bytes	Remarks
Read	03h	2	—	1 to ∞	RISEL[2:0] = 000b

Table 42.6 SPI Instructions Automatically Generated When SPAMR.SIZE[1:0] = 10b

SPI Instruction	Instruction code	Address bytes	Dummy cycles	Data bytes	Remarks
Read	03h	3	—	1 to ∞	RISEL[2:0] = 000b
Fast Read	0Bh	3	8*1	1 to ∞	RISEL[2:0] = 001b
Fast Read Dual Output	3Bh	3	8*1	1 to ∞	RISEL[2:0] = 010b
Fast Read Dual I/O	BBh	3	4*1	1 to ∞	RISEL[2:0] = 011b
Fast Read Quad Output	6Bh	3	8*1	1 to ∞	RISEL[2:0] = 100b
Fast Read Quad I/O	EBh	3	6*1	1 to ∞	RISEL[2:0] = 101b

Note 1. The number of dummy cycles is configurable in SPDCR.DCYC[3:0].

Table 42.7 SPI Instructions Automatically Generated When SPAMR.SIZE[1:0] = 11b and I4AE = 0

SPI Instruction	Instruction code	Address bytes	Dummy cycles	Data bytes	Remarks
Read	03h	4	—	1 to ∞	RISEL[2:0] = 000b
Fast Read	0Bh	4	8*1	1 to ∞	RISEL[2:0] = 001b
Fast Read Dual Output	3Bh	4	8*1	1 to ∞	RISEL[2:0] = 010b
Fast Read Dual I/O	BBh	4	4*1	1 to ∞	RISEL[2:0] = 011b
Fast Read Quad Output	6Bh	4	8*1	1 to ∞	RISEL[2:0] = 100b
Fast Read Quad I/O	EBh	4	6*1	1 to ∞	RISEL[2:0] = 101b

Note 1. The number of dummy cycles is configurable in SPDCR.DCYC[3:0].

Table 42.8 SPI Instructions Automatically Generated When SPAMR.SIZE[1:0] = 11b and I4AE = 1

SPI Instruction	Instruction code	Address bytes	Dummy cycles	Data bytes	Remarks
Read	13h	4	—	1 to ∞	RISEL[2:0] = 000b
Fast Read	0Ch	4	8*1	1 to ∞	RISEL[2:0] = 001b
Fast Read Dual Output	3Ch	4	8*1	1 to ∞	RISEL[2:0] = 010b
Fast Read Dual I/O	BCh	4	4*1	1 to ∞	RISEL[2:0] = 011b
Fast Read Quad Output	6Ch	4	8*1	1 to ∞	RISEL[2:0] = 100b
Fast Read Quad I/O	ECh	4	6*1	1 to ∞	RISEL[2:0] = 101b

Note 1. The number of dummy cycles is configurable in SPDCR.DCYC[3:0].

42.6.2 Read Instruction

Read instruction is a common read method supported by most serial ROMs.
When the SPI bus cycle is started, the QSSL signal is asserted and the instruction code (03h/13h)*1 and the one- to four-byte address specified by the SPAMR.SIZE[1:0] bits are transmitted from the QIO0 pin. After that, data is received from the QIO1 pin.

Note 1. Most 4K-bit serial ROMs have a 1-byte address field and assign the most significant bit (A8) of the address to bit 3 of the Read instruction code to minimize the overhead. To support these devices, when the SPAMR.SIZE[1:0] bits are 00b (8-bit address), QSPIX outputs A8 (address bit 8) to bit 3 of the Read instruction code. Therefore 0Bh may be output instead of 03h as the code of the Read instruction. This code (0Bh) overlaps with the code of the Fast Read instruction, but most serial ROMs of 2K-bit or less are designated to ignore bit 3 of the instruction code, so 0Bh is correctly recognized as the Read instruction. In rare cases, there are serial ROMs that allow bit 3 to be decoded, but when such a serial ROM is connected, avoid access that causes A8 to be 1.

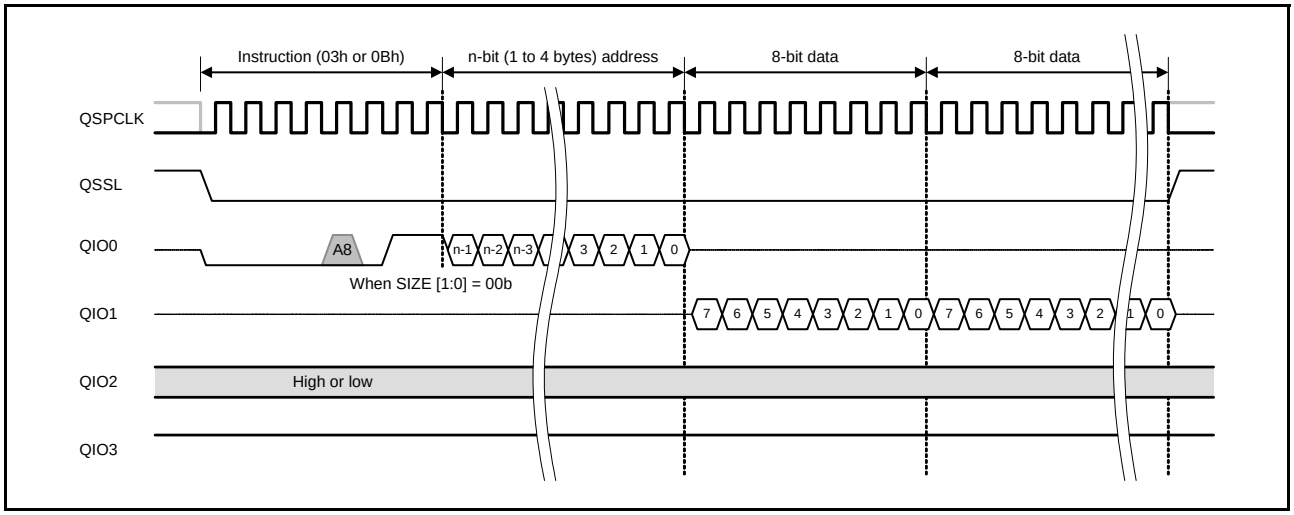


Figure 42.13 Read Instruction Bus Cycle

42.6.3 Fast Read Instruction

Fast Read instruction is a read method that supports a communication clock that is faster than the Read instruction.
When the SPI bus cycle is started, the QSSL signal is asserted, the instruction code (0Bh/0Ch) and the one- to four-byte address specified by the SPAMR.SIZE[1:0] bits are transmitted from the QIO0 pin, and the dummy cycles specified in the SPDCR.DCYC[3:0] bits are inserted. After that, data is received from the QIO1 pin.
The Dual SPI protocol uses the QIO0 and QIO1 pins, and the Quad SPI protocol uses the QIO0 to QIO3 pins to transmit and receive instruction codes, addresses, and data.
The first two dummy cycles are used to select the XIP mode. When the XIP mode is selected, the same instruction used in this SPI bus cycle is applied to the next SPI bus cycle, so the instruction code is omitted in the next SPI bus cycle. For details on the XIP mode, refer to section 42.8, XIP Mode.
Switching to the Fast Read instruction is controlled by the SPMR0 register.

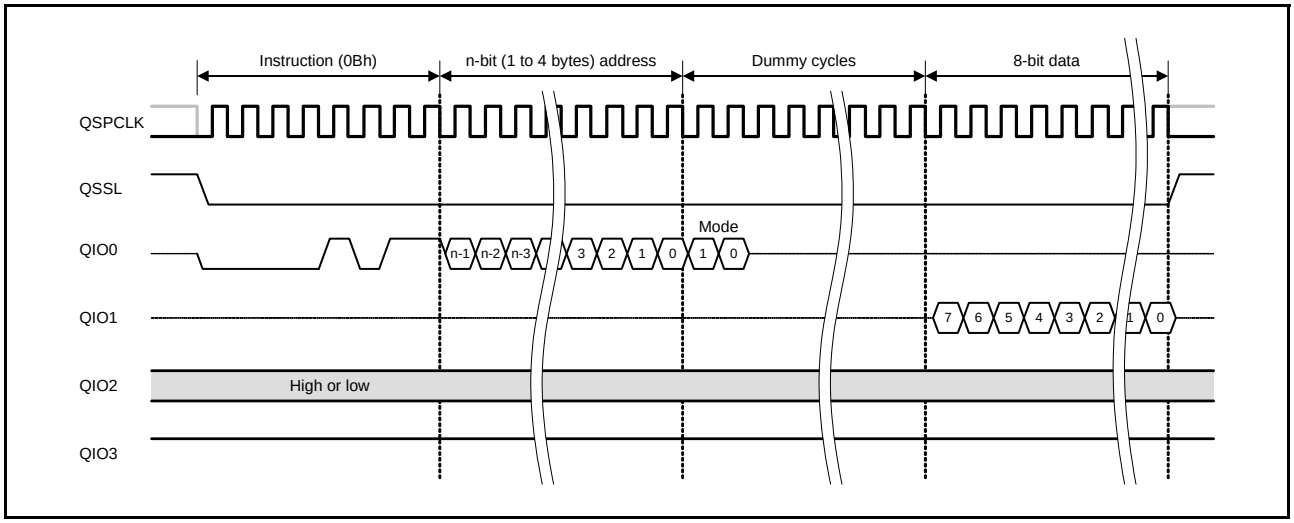


Figure 42.14 Fast Read Instruction Bus Cycle (Extended SPI Protocol)

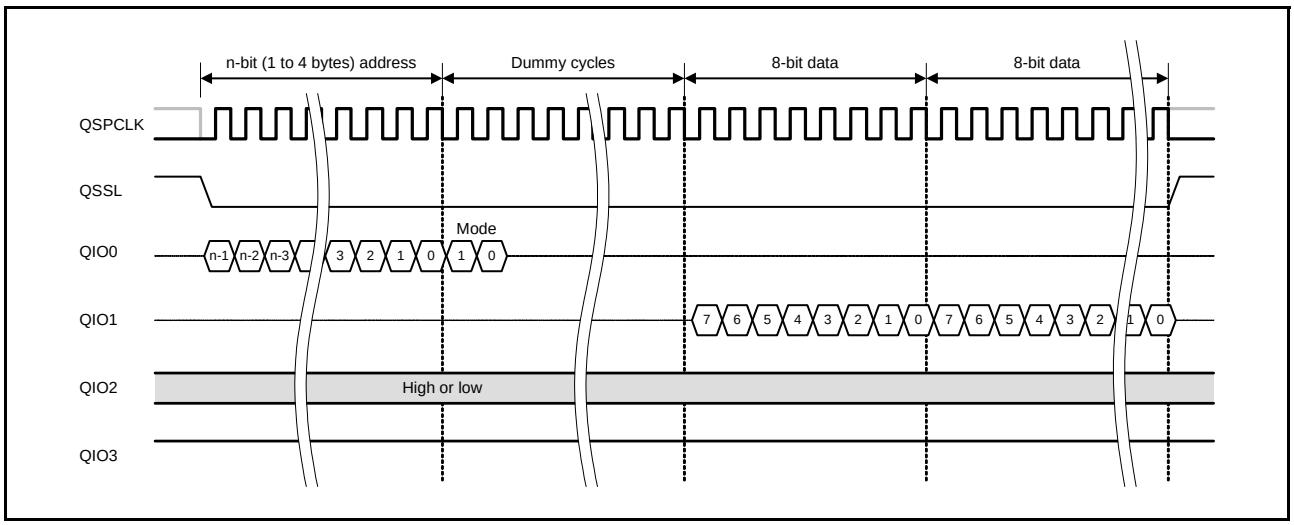


Figure 42.15 Fast Read Instruction Bus Cycle in XIP Mode (Extended SPI Protocol)

Note: To use the Fast Read instruction, a serial ROM that supports Fast Read transfers is required.

42.6.4 Fast Read Dual Output Instruction

The Fast Read Dual Output instruction is a read method that uses two signal lines to receive data.
When the SPI bus cycle is started, the QSSL signal is asserted, the instruction code (3Bh/3Ch) and the one- to four-byte address specified by the SPAMR.SIZE[1:0] bits are transmitted from the QIO0 pin (the QIO0 and QIO1 pins in Dual SPI protocol), and the dummy cycles specified in the SPDCR.DCYC[3:0] bits are inserted. After that, data is received form the QIO0 and QIO1 pins. Even bits of data are received from the QIO0 pin and odd bits are received from the QIO1 pin.
The first two dummy cycles are used to select the XIP mode. When the XIP mode is selected, the same instruction used in this SPI bus cycle is applied to the next SPI bus cycle, so the instruction code is omitted in the next SPI bus cycle. For details on the XIP mode, refer to section 42.8, XIP Mode.
Switching to Fast Read Dual Output instruction is controlled by the SPMR0 register.

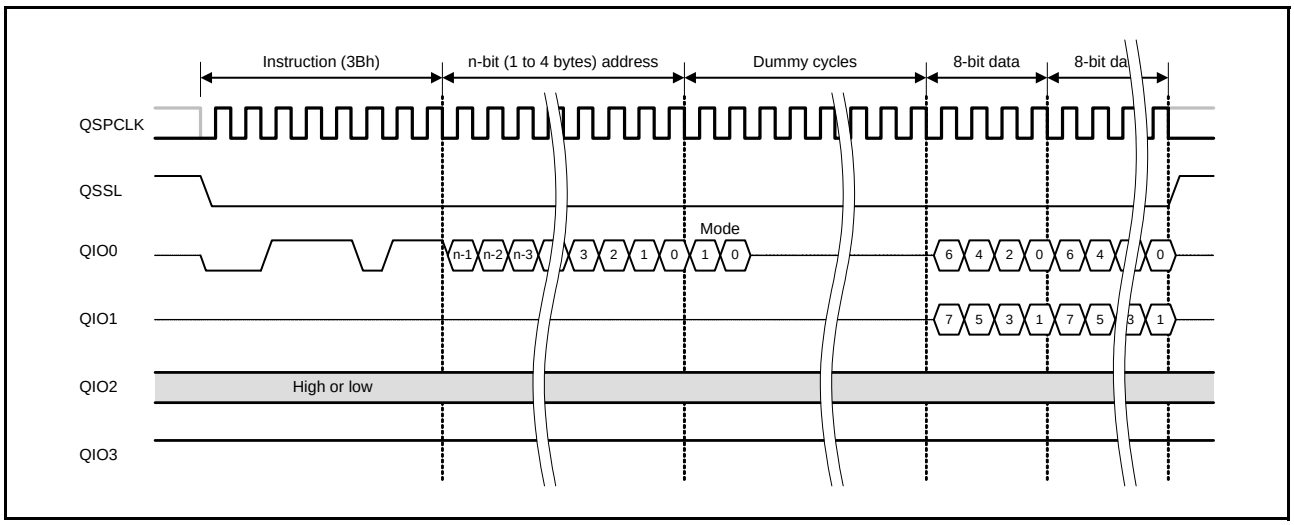


Figure 42.16 Fast Read Dual Output Instruction Bus Cycle (Extended SPI Protocol)

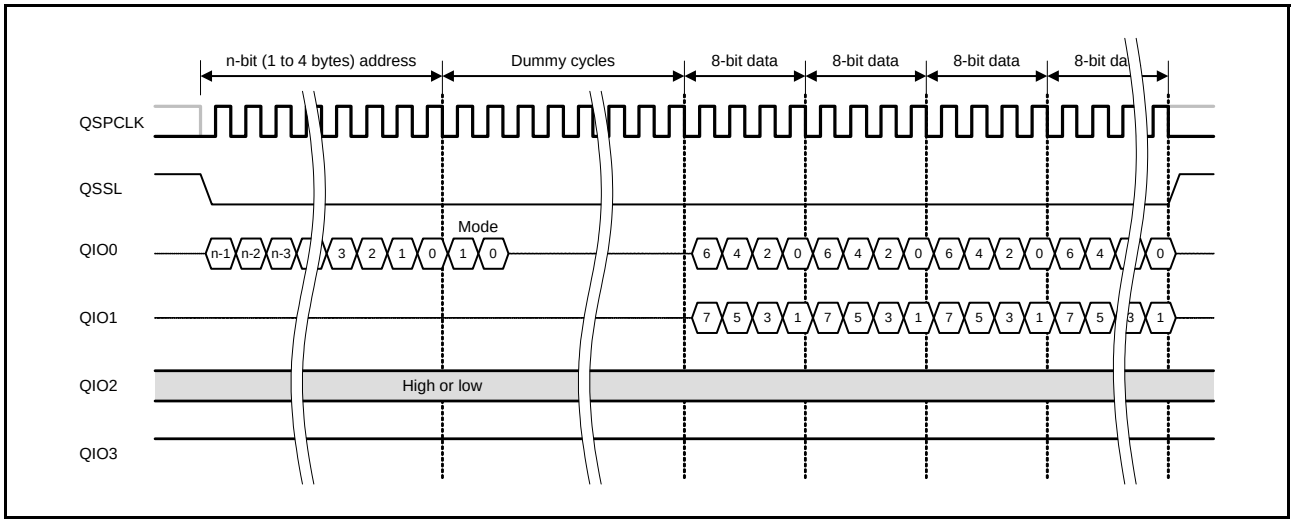


Figure 42.17 Fast Read Dual Output Instruction Bus Cycle in XIP Mode (Extended SPI Protocol)

Note: To use the Fast Read Dual Output instruction, a serial ROM that supports Fast Read Dual Output transfers is required.

42.6.5 Fast Read Dual I/O Instruction

The Fast Read Dual I/O instruction is a read method that uses two signal lines for address transmission and data reception.

When the SPI bus cycle is started, the QSSL signal is asserted, the instruction code (BBh/BCh) is transmitted from the QIO0 pin (the QIO0 and QIO1 pins in Dual SPI protocol), the one- to four-byte address specified by the SPAMR.SIZE[1:0] bits is transmitted from the QIO0 and QIO1 pins, and the dummy cycles specified in the SPDCR.DCYC[3:0] bits is inserted. After that, data is received from the QIO0 and QIO1 pins. The QIO0 pin is used for even bits of address, dummy byte, and data, and the QIO1 pin is used for odd bits.

The first two dummy cycles are used to select the XIP mode. When the XIP mode is selected, the same instruction used in this SPI bus cycle is applied to the next SPI bus cycle, so the instruction code is omitted in the next SPI bus cycle. For details on the XIP mode, refer to section 42.8, XIP Mode.

Switching to Fast Read Dual I/O instruction is controlled by the SPMR0 register.

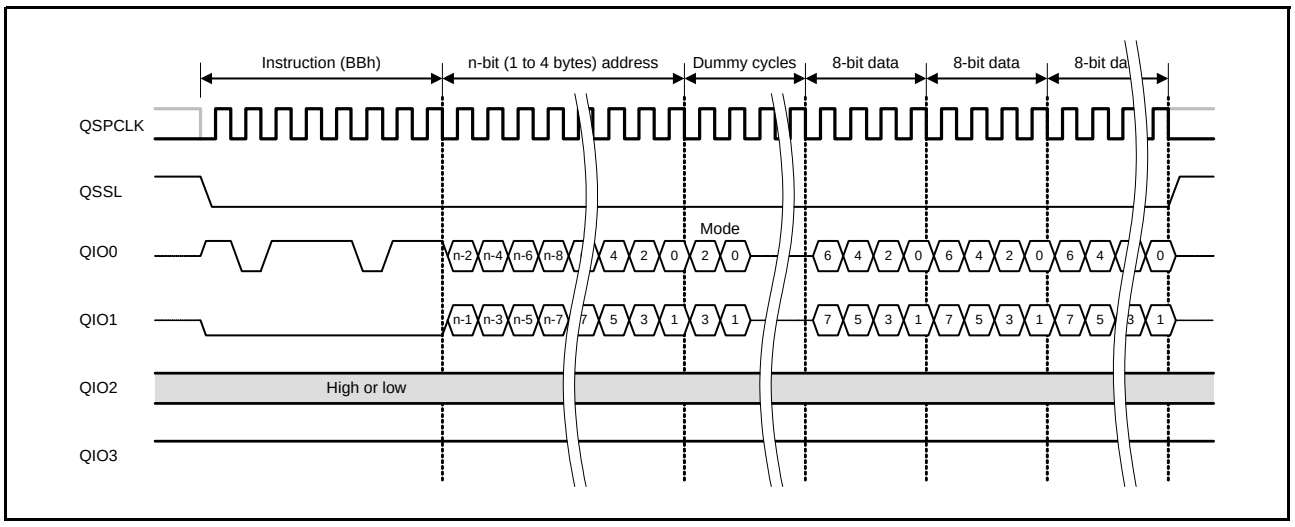


Figure 42.18 Fast Read Dual I/O Instruction Bus Cycle (Extended SPI Protocol)

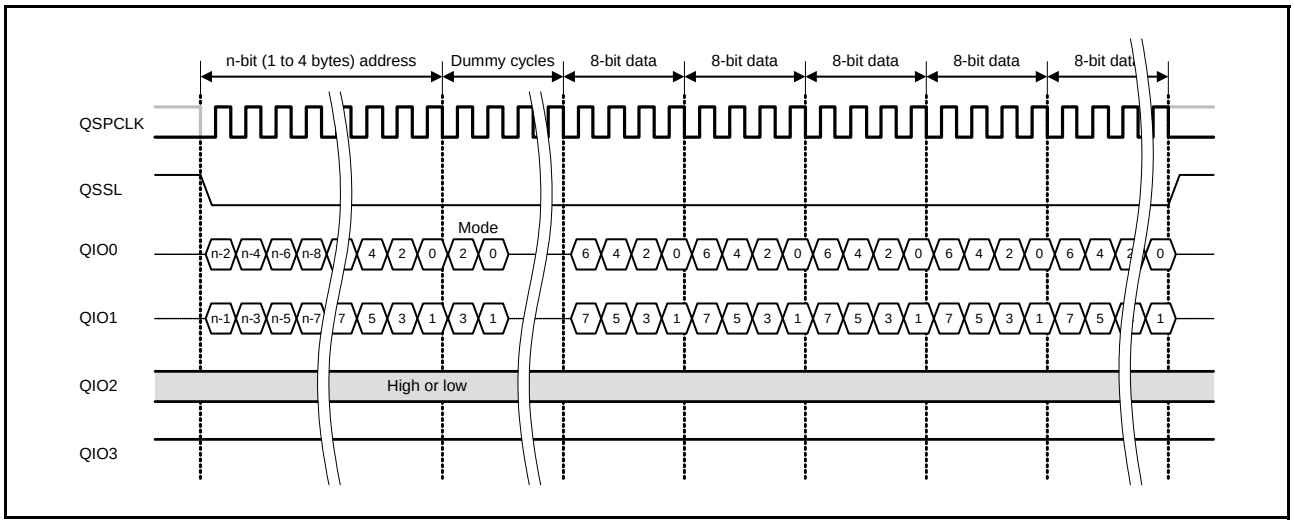


Figure 42.19 Fast Read Dual I/O Instruction Bus Cycle in XIP Mode

Note: To use the Fast Read Dual I/O instruction, a serial ROM that supports Fast Read Dual I/O transfers is required.

42.6.6 Fast Read Quad Output Instruction

The Fast Read Quad Output instruction is a read method that uses four signal lines to receive data.
When the SPI bus cycle is started, the QSSL signal is asserted, the instruction code (6Bh/6Ch) and the one- to four-byte address specified by the SPAMR.SIZE[1:0] bits are transmitted from the QIO0 pin (the QIO0 to QIO3 pins in Quad SPI protocol), and the dummy cycles specified in the SPDCR.DCYC[3:0] bits are inserted. After that, data is received form the QIO0 to QIO3 pins.
The first two dummy cycles are used to select the XIP mode. When the XIP mode is selected, the same instruction used in this SPI bus cycle is applied to the next SPI bus cycle, so the instruction code is omitted in the next SPI bus cycle. For details on the XIP mode, refer to section 42.8, XIP Mode.
Switching to Fast Read Quad Output instruction is controlled by the SPMR0 register.

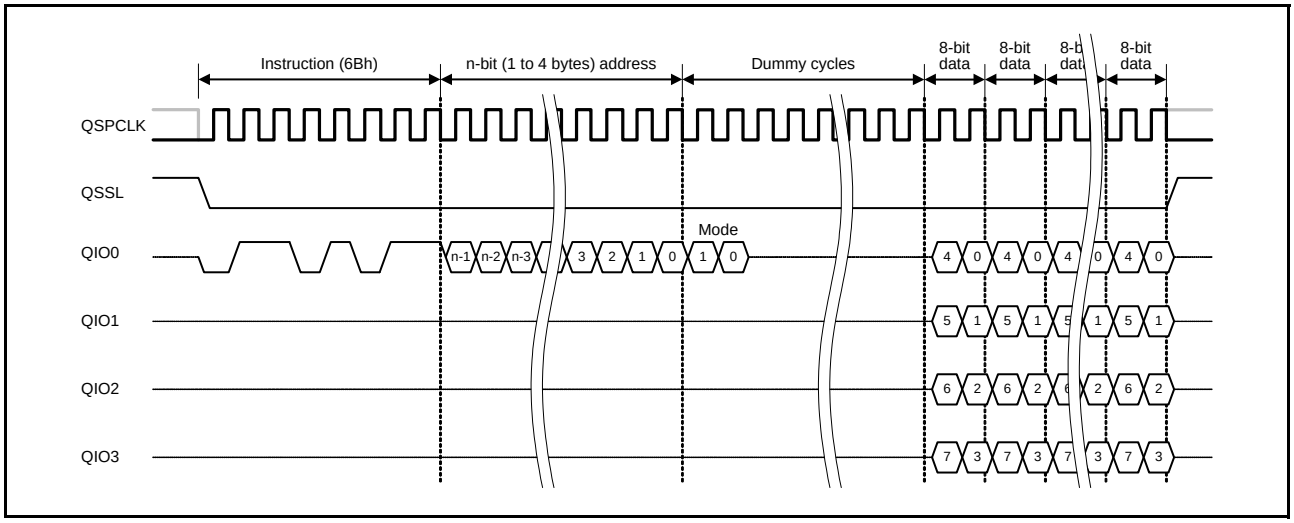


Figure 42.20 Fast Read Quad Output Instruction Bus Cycle (Extended SPI Protocol)

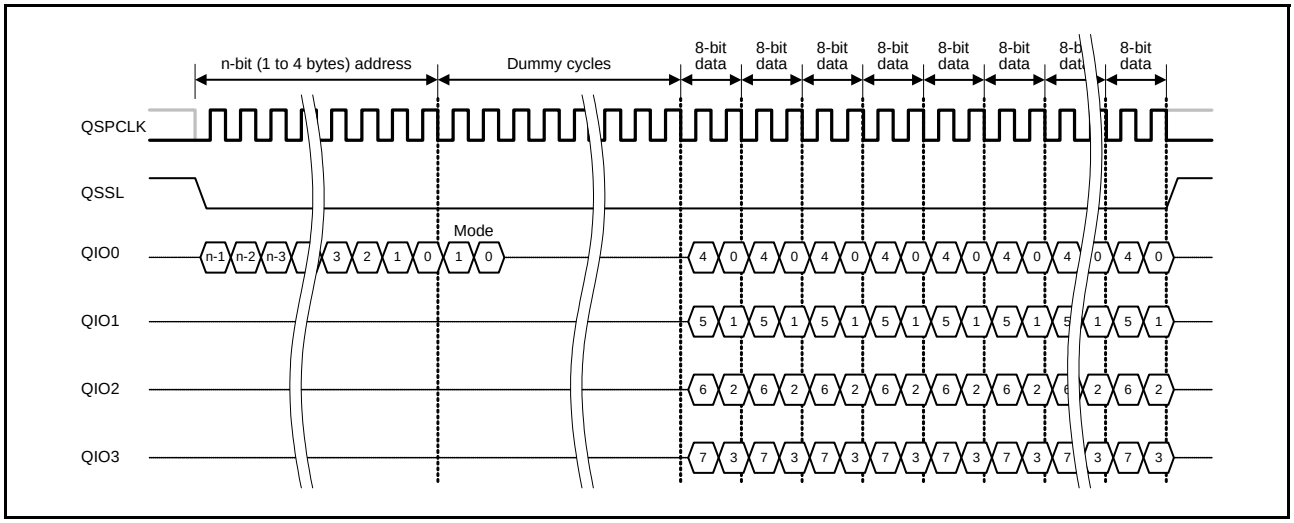


Figure 42.21 Fast Read Quad Output Instruction Bus Cycle in XIP Mode (Extended SPI Protocol)

Note: To use Fast Read Quad Output, a serial ROM that supports Fast Read Quad Output transfer is required.

42.6.7 Fast Read Quad I/O Instruction

The Fast Read Quad I/O instruction is a read method that uses four signal lines for address transmission and data reception.

When the SPI bus cycle is started, the QSSL signal is asserted, the instruction code (EBh/ECh) is transmitted from the QIO0 pin (the QIO0 to QIO3 pins in Quad SPI protocol), the one- to four-byte address specified by the SPAMR.SIZE[1:0] bits is transmitted from the QIO0 to QIO3 pins, and the dummy cycles specified in the SPDCR.DCYC[3:0] bits is inserted. After that, data is received from the QIO0 to QIO3 pins.

The first two dummy cycles are used to select the XIP mode. When the XIP mode is selected, the same instruction used in this SPI bus cycle is applied to the next SPI bus cycle, so the instruction code is omitted in the next SPI bus cycle. For details on the XIP mode, refer to section 42.8, XIP Mode.

Switching to Fast Read Quad I/O instruction is controlled by the SPMR0 register.

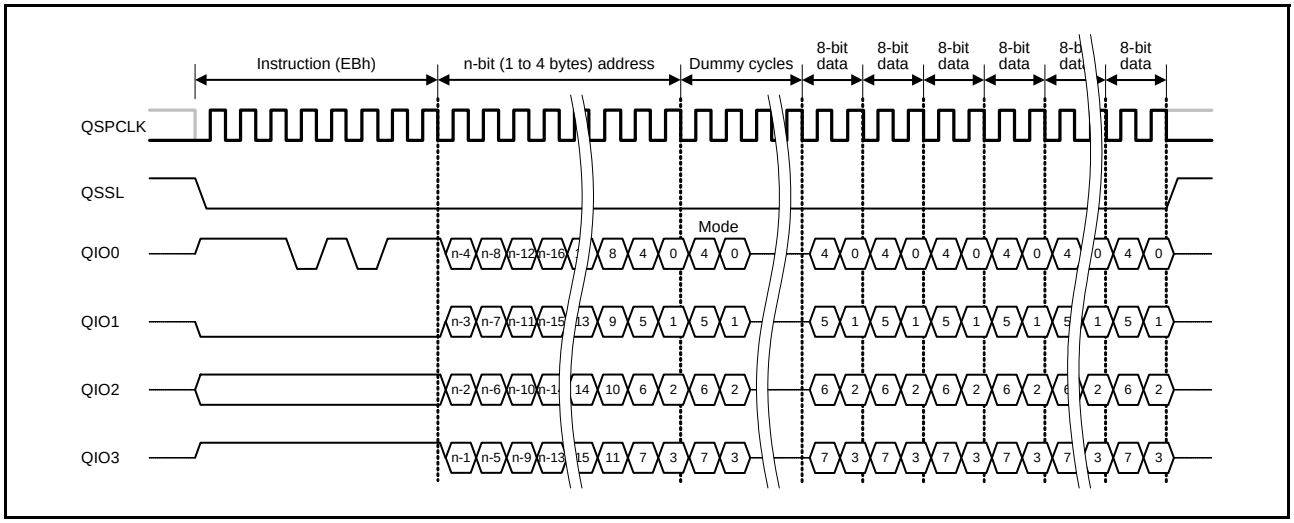


Figure 42.22 Fast Read Quad I/O Instruction Bus Cycle (Extended SPI Protocol)

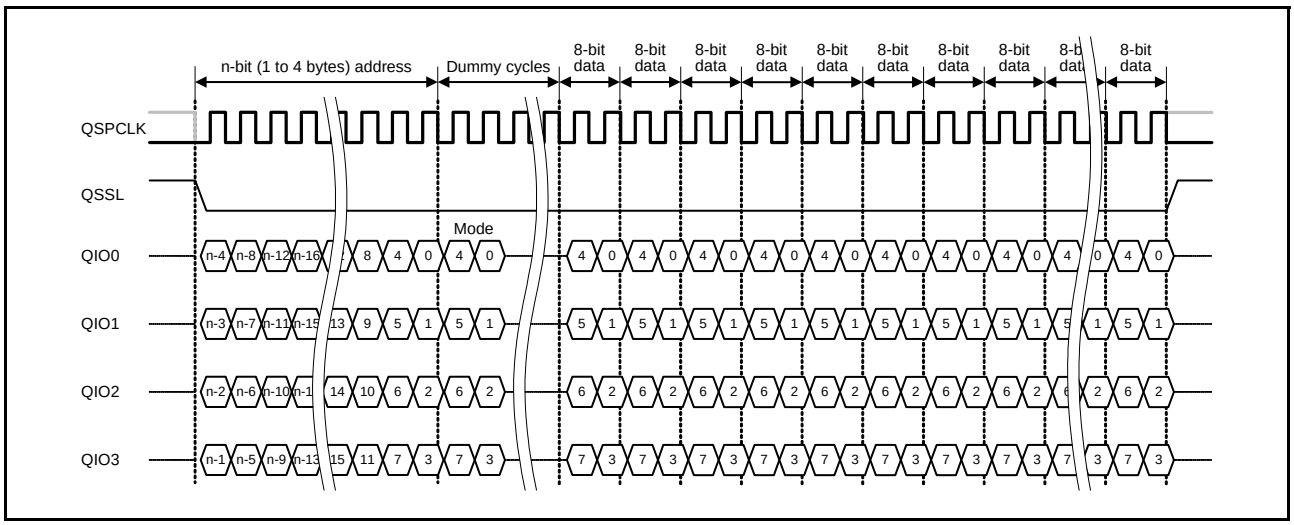


Figure 42.23 Fast Read Quad I/O Instruction Bus Cycle in XIP Mode

Note: To use the Fast Read Quad I/O instruction, a serial ROM that supports Fast Read Quad I/O transfers is required.

42.7 SPI Bus Cycle Generation

42.7.1 Standard ROM Read

Read access in memory-mapped mode is converted into single SPI bus cycle for each access.
When QSPIX detects a read access to QSPI memory space (address 7000 0000h to 73FF FFFFh), it asserts the QSSL signal and initiates the SPI bus cycle. When it receives the required number of data from the serial ROM, it negates the QSSL signal and terminates the SPI bus cycle.
When it detects the next read access, it secures the minimum high width of the QSSL signal and then start a new SPI bus cycle.

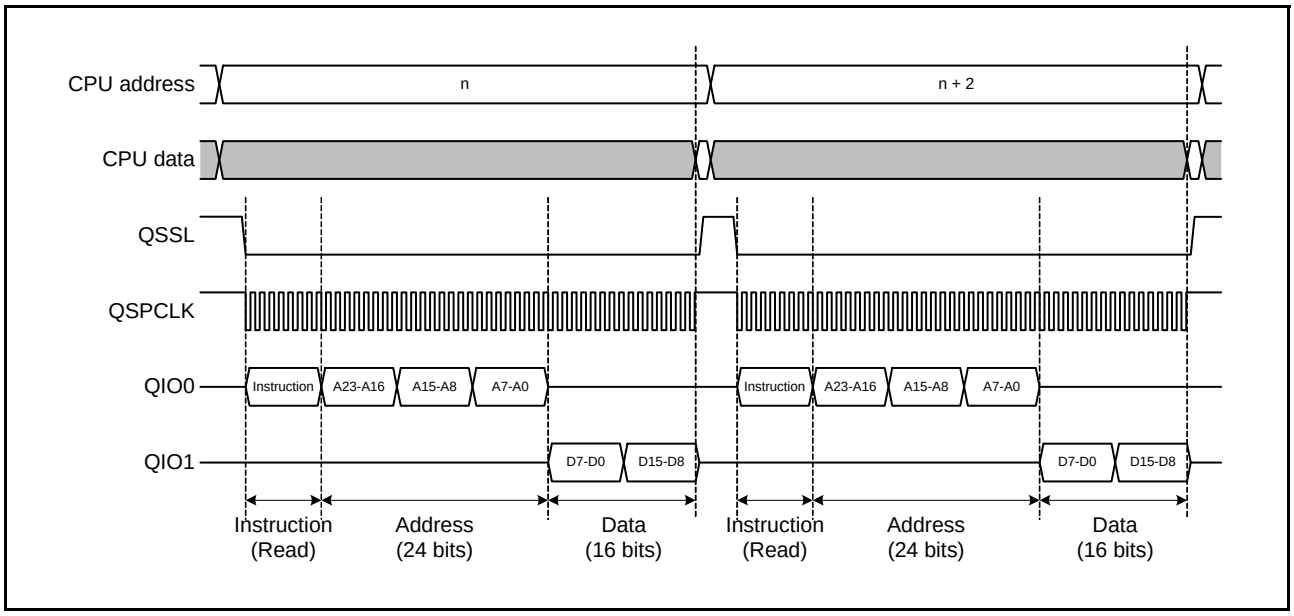


Figure 42.24 Successive Data Read Operations

42.7.2 ROM Read Using Prefetch Function

In many cases, such as CPU instruction execution and block data transfer, ROM data is read from consecutive addresses in ascending order.
The memory read command of the serial ROM can read data infinitely in a single SPI bus cycle. However, if the bus cycle issued by the CPU are individually converted to the SPI bus cycle, the SPI bus cycle is divided and this feature of the serial ROM cannot be utilized.
QSPIX has a prefetch function to take advantage of this feature.
The prefetch function is enabled when the SPMR0.PFE bit is set to 1. When the prefetch function is enabled, QSPIX continuously receives and buffers data without waiting for the next ROM read request. When the CPU reads the ROM, it compares the addresses and returns the data in the buffer to the CPU if the addresses match. If the addresses do not match, the data in the buffer is discarded and a new SPI bus cycle is generated.
The size of the prefetch buffer is 18 bytes. When this buffer is full, the SPI bus cycle ends. When the data in the buffer is read and the buffer becomes free, a new SPI bus cycle is automatically started to resume prefetching.
The prefetch function realizes efficient data transfer in situations where data is read in ascending order from consecutive addresses such as instruction fetch and block data transfer.

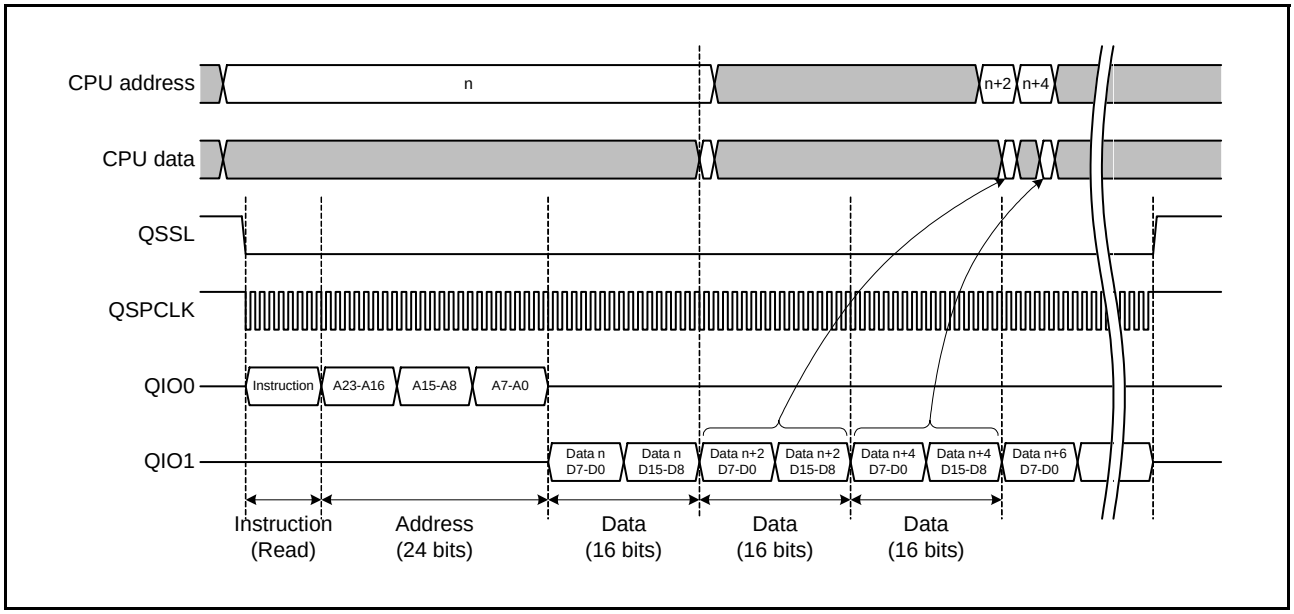


Figure 42.25 Continuous Data Read Using Prefetch Function

42.7.3 Termination of Prefetch

If a ROM read bus cycle to another address occurs while receiving data for prefetching, QSPIX terminates the current SPI bus cycle that is no longer needed and starts a new SPI bus cycle.

42.7.4 Specifying the Prefetch Address

When a write access to the QSPI memory space is detected while the PFE bit is 1, QSPIX obtains the current address as a prefetch address and starts prefetching. Write access to the QSPI memory space is only used to obtain the prefetch address, not to write to the serial ROM.
Combining this function with the prefetch status polling function described below can reduce the load on the internal bus when reading data from a low-speed serial ROM.

Note: When specifying the prefetch address, write to the QSPI memory space in byte size. Writing in word size or long word size causes a bus error.

42.7.5 Prefetch Status Polling

If a low-speed serial ROM is used, the read cycle from the QSPI memory space has to wait until the end of the SPI bus cycle, which increases the load on the internal bus.
The prefetch status polling function is provided to reduce this load.
Read the SPPFSR.PBLVL[4:0] bits to get the number of prefetched data and perform read access to the QSPI memory space if there is the required number of data.

```
//
// copy 1 Kbyte (32 bit x 256 word) data from serial ROM to SDRAM
//
unsigned long *sptr;                        // pointer for the serial ROM
unsigned long *dptr;                        // pointer for the SDRAM
int i;

SPMR0 |= 0x0040;                            // set PFE bit to enable prefetch
*( (volatile unsigned char *) sptr ) = 0;   // make the TAG valid to start prefetch

for ( i = 0 ; i < 256 ; i++ ) {
   while ( ( SPPFSR & 0x00FF ) < 0x04 ) {   // wait for 4-byte data to be received
   };
   *(dptr++) = *(sptr++);
}
```

Note: Place the polling program outside the QSPI memory space. If the polling program is placed on the serial ROM, the prefetch target is frequently switched to the instruction code, so the effect of polling is lost, the prefetch buffer may not be filled, and an infinite loop may occur.

42.7.6 ROM Read Using the Slave Select Extension Function

If the SPMR0.SSE[1:0] bits are set to a value other than 00b, QSPIX waits for the next ROM read without terminating the SPI bus cycle (while stopping the QSPCLK signal and holding the QSSL signal low) even after receiving data from the serial ROM.

If the address of the next ROM read is a continuation of the current address, the QSPCLK signal is resumed and subsequent data reception continues. If the address of the next ROM read is not a continuation of the current address, the QSSL signal is driven high to end the SPI bus cycle, and then a new SPI bus cycle is started.

This function eliminates the time to transmit instruction codes and addresses, when reading data intermittently from consecutive addresses, enabling efficient data transfer.

The extension time of the QSSL signal can be selected with the SPMR0.SSE[1:0] bits. When the specified time elapses, the QSSL signal goes high and the SPI bus cycle ends automatically. If the SSE[1:0] bits are set to 11b, the QSSL signal is extended infinitely, which increases the power consumption of the serial ROM.

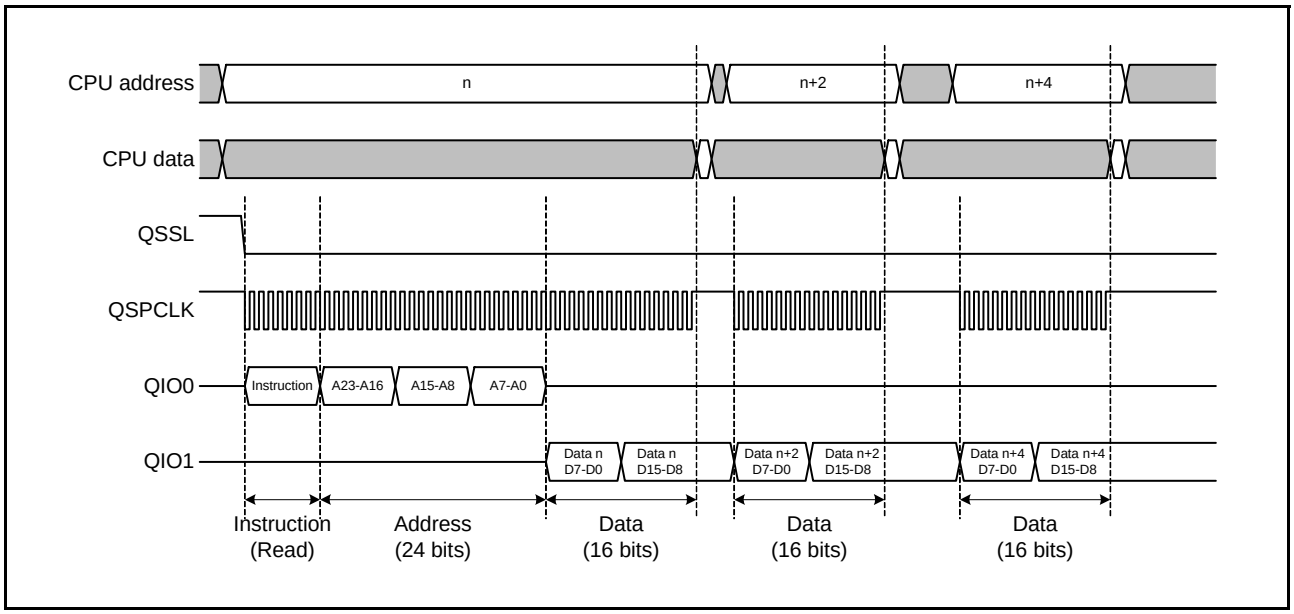


Figure 42.26 Successive Data Read Using the Slave Select Extension Function

42.8 XIP Mode

Some serial ROMs can reduce latency by omitting the instruction code reception for reading the ROM. This function is controlled by the mode data received during the dummy cycles in the previous SPI bus cycle.
In QSPIX, when the SPDCR.XIPE bit is set to 1, the data specified in the SPDCR.MODE[7:0] bits are transmitted during the first two cycles of the dummy cycles as shown in Figure 42.27.
The mode data to enable the XIP mode differs for each serial ROM, so set the MODE[7:0] bits properly according to the serial ROM to be used.

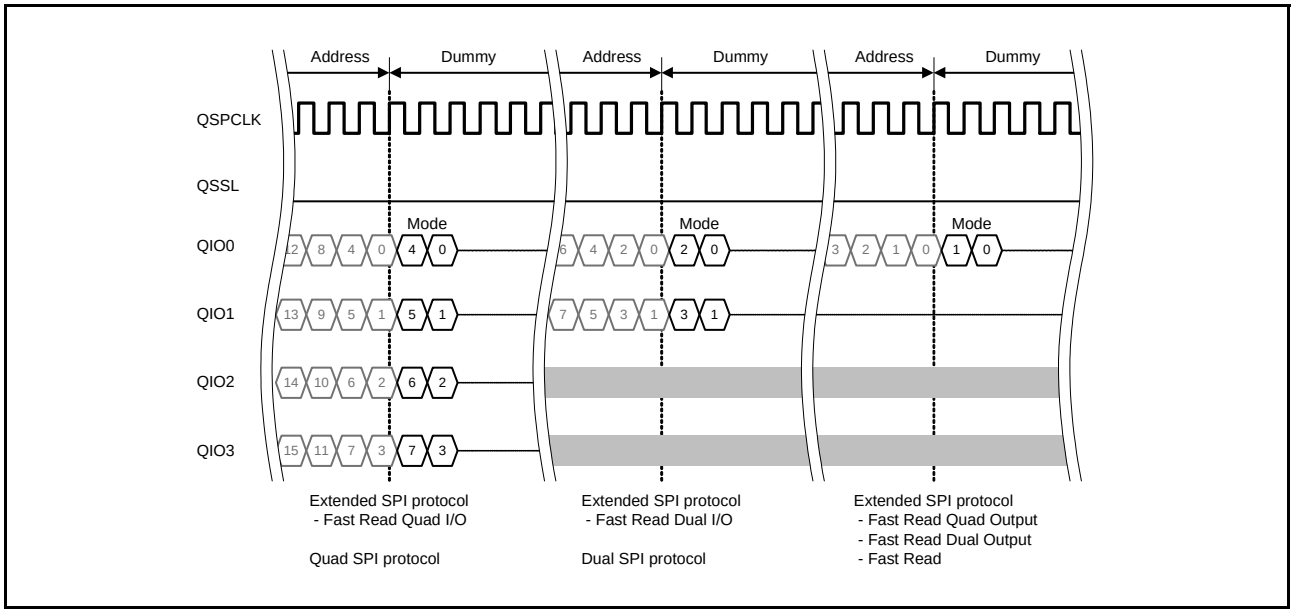


Figure 42.27 XIP Mode Control Data

42.8.1 Activation of the XIP Mode

To activate the XIP mode of the serial ROM, specify the value for activating the XIP mode of the serial ROM to be used in the SPDCR.MODE[7:0] bits, and set the XIPE bit to 1. The value specified in the MODE[7:0] bits is transmitted during the dummy cycles in the next SPI bus cycle. Read the SPDCR.XIPS flag to confirm if the XIP mode has been activated.

42.8.2 Termination of the XIP Mode

To terminate the XIP mode of the serial ROM, specify the value for terminating the XIP mode of the serial ROM to be used in the SPDCR.MODE[7:0] bits, and set the XIPE bit to 0. The value specified in the MODE[7:0] bits is transmitted during the dummy cycles in the next SPI bus cycle. Read the SPDCR.XIPS flag to confirm if the XIP mode has been terminated.

42.9 QIO2 and QIO3 Pin States

The status of the QIO2 and QIO3 pins depend on the instruction selected by the SPMR0.RISEL[2:0] bits.

Table 42.9 QIO2 and QIO3 Pin States

SPMR0.RISEL[2:0] bits	Instruction	QIO2 pin state*1	QIO3 pin state*2
000b	Read	Output the level specified by SPPCR.WP bit (initial output is low)	Output high
001b	Fast Read		
010b	Fast Read Dual Output		
011b	Fast Read Dual I/O		
100b	Fast Read Quad Output	Serial data input or output (Hi-Z during standby)	Serial data input or output (Hi-Z during standby)
101b	Fast Read Quad I/O		
110b	Setting prohibited	—	—
111b			

Note 1. Some serial ROMs may share the QIO2 pin with the WP# pin.
Note 2. Some serial ROMs may share the QIO3 pin with the HOLD# or RESET# pin.

42.10 Indirect Access Mode

In addition to memory read command, the serial ROMs have various commands such as device ID read, erase, program, and status read. To use these commands, QSPIX provides a function to access the serial ROM via I/O register.

42.10.1 Using Indirect Access Mode

To send a commands to the serial ROM via the I/O register, set the SPMR1.AMOD bit to 1 (indirect access mode). Do not perform normal serial ROM reads via the QSPI memory space while the indirect access mode is selected. To return to normal serial ROM reading after using the indirect access mode, set the AMOD bit to 0 (memory mapped mode).

Note: If the XIP mode is activated, terminate the XIP mode before entering indirect access mode.

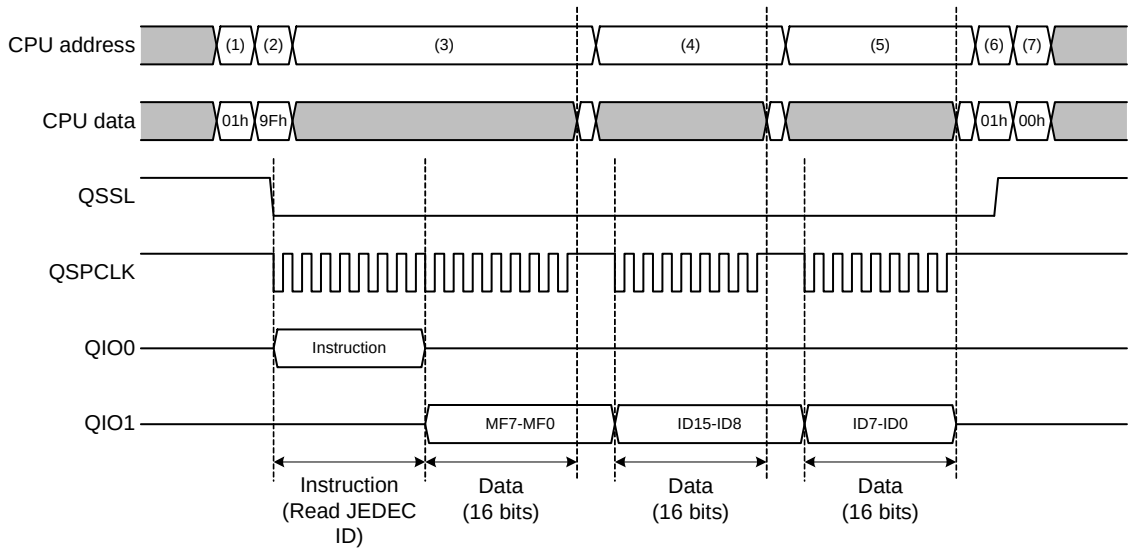
42.10.2 Generating SPI Bus Cycle during Indirect Access Mode

In indirect access mode, the first access to the SPDR register starts the SPI bus cycle and write to the SPMR1 register ends the SPI bus cycle. The QSSL signal goes low during the SPI bus cycle.
A write to the SPDR register during the SPI bus cycle is converted to one-byte transmission to the SPI bus, and a read from the SPDR register is converted to one-byte reception from the SPI bus.

Note: In indirect access mode, writing to registers other than the SPMR1 and SPDR registers (SPMR0, SPSSCR, SPOCR, SPPFSR, SPSR, SPRIR, SPAMR, SPDCR, SPMR2, SPPCR, and SPUAR) is prohibited.

The following is an example program for indirect access mode.

```
//### CAUTION! ### This code must be outside the serial ROM that is to be controlled.

// Define specific instruction codes of the target serial ROM.
#define Instruction_FREAD 0x0B          // Fast Read
#define Instruction_RDSR 0x05           // Read Status register
#define Instruction_RDID 0x9F           // Read Identification
#define Instruction_WREN 0x06           // Write Enable
#define Instruction_CERA 0xC7           // Chip Erase

unsigned char mfid, mtype, mcap, data, temp;

SPMR1 = 0x01;                           // Select indirect access mode

// Get the device identification assigned by JEDEC.
SPDR = Instruction_RDID;                // put “Read Identification” instruction (start SPI bus cycle)
mfid = (unsigned char) SPDR;            // get “Manufacturer Identification”
mtype = (unsigned char) SPDR;           // get “Memory Type”
mcap = (unsigned char) SPDR;            // get “Memory Capacity”
SPMR1 = 0x01;                           // end SPI bus cycle

// Get one byte from the address 0x012345.
SPDR = Instruction_FREAD;               // put “Fast Read” instruction (start SPI bus cycle)
SPDR = 0x01;                            // put upper byte of the address 0x012345
SPDR = 0x23;                            // put middle byte of the target address 0x012345
SPDR = 0x45;                            // put lower byte of the target address 0x012345
temp = (unsigned char) SPDR;            // get one byte dummy code for FAST READ transaction
data = (unsigned char) SPDR;            // get the data
SPMR1 = 0x01;                           // end SPI bus cycle

// Erase All contents.
SPDR = Instruction_WREN;                // put “Write Enable” instruction (start SPI bus cycle)
SPMR1 = 0x01;                           // end SPI bus cycle
SPDR = Instruction_CERA;                // put “Chip Erase” instruction (start SPI bus cycle)
SPMR1 = 0x01;                           // end SPI bus cycle
SPDR = Instruction_RDSR;                // put “Read Status Register” instruction (start SPI bus cycle)
while (SPDR & 0x01){};                  // polling “Write Progress Bit” until completion
SPMR1 = 0x01;                           // end SPI bus cycle

SPMR1 = 0x00;                           // Select memory mapped mode
```



(1)	SPMR1 = 01h;	// Select indirect access mode // Get the device identification assigned by JEDEC
(2)	SPDR = Instruction_RDID;	// Put “Read Identification” instruction (open the transaction)
(3)	mfid = (unsigned char) SPDR;	// Get ID byte-1 “Manufacturer Identification”
(4)	mtype = (unsigned char) SPDR;	// Get ID byte-2 “Memory Type”
(5)	mcap = (unsigned char) SPDR;	// Get ID byte-3 “Memory Capacity”
(6)	SPMR1 = 01h;	// Close the transaction
(7)	SPMR1 = 00h;	// Select memory mapped mode

Figure 42.28 Example of Timing in Indirect Access Mode (Read JEDEC ID)

Note: When Single/Extended SPI protocol is used in indirect access mode, the standard Read or Fast Read instruction must be used to reference the contents of the serial ROM. The QSPIX does not support Fast Read Dual Output, Fast Read Dual I/O, Fast Read Quad Output, or Fast Read Quad I/O instructions in this configuration. When these instructions are required, use memory mapped mode.

42.11 Interrupts

If read access is performed to the QSPI memory space during the indirect access mode, the ROMAE flag becomes 1 and a ROM access error interrupt request (ERI) is generated. The interrupt request is retained until the ROMAE flag is cleared. For details, refer to section 15., Interrupt Controller (ICUE).

42.12 Usage Notes

42.12.1 Settings of the Module Stop Function

QSPIX operation can be disabled or enabled by module stop control register D (MSTPCRD).
After a reset, the QSPIX is stopped. The registers become accessible when it is released from the module stop state. For details, refer to section 11., Low Power Consumption.

42.12.2 Procedure for Changing the Settings of Multiple Control Registers

The QSPIX registers can be changed during the SPI bus cycle. However, if the settings of multiple control registers are changed in succession, the SPI bus cycle may occur before all the settings are completed. Set the registers so that the SPI bus timing specifications can be satisfied at any stage of register setting.
The following is an example program for changing the QSPCLK frequency.

```
//
// Making QSPCLK faster
//
SPMR0 = 0x0041;             // PFE = 1, SSE = 00b, RISEL = 001b (prefetch enable fast read)
SPSSCR = 0x04;              // SSSU = 0, SSHLD = 0, SSHW = 4 (minimum QSSL high width = 5 cycles)
SPOCR = 0x00;               // DUTY = 0, DIV = 0 (1/2 mode)  ### switch clock speed last ###

//
// Making QSPCLK slower
//
SPOCR = 0x06;               // DUTY = 0, DIV = 6 (1/8 mode)  ### switch clock speed first ###
SPSSCR = 0x01;              // SSSU = 0, SSHLD = 0, SSHW = 1 (minimum QSSL high width = 2 cycles)
SPMR0 = 0x0040;             // PFE = 1, SSE = 00b, RISEL = 000b (prefetch enable, standard read)
```

42.12.3 Bus Timeout

When the bus timeout detection is enabled (BSC.BEREN.TOEN bit = 1), a bus timeout may occur if it takes a long time to acquire data from the serial ROM.
In such cases, disable the bus timeout detection or increase the frequency of the QSPIX operating clock.
For details on bus timeout, refer to section 16.7.1.2, Timeout.

42.12.4 Note on the Endian Setting of CPU

The internal expansion bus to which QSPIX is connected is fixed to little endian.
If the CPU is set to big endian (OFSM.MDE.MDE[2:0] bits are 000b), the data arrangement is reversed, so be careful with the data arrangement. Also, in this case, the CPU cannot execute the program directly on the serial ROM.
When executing the program directly on the serial ROM, set the CPU to little endian (OFSM.MDE.MDE[2:0] bits to 111b).

43. CRC Calculator (CRCA)

The CRC (Cyclic Redundancy Check) calculator generates CRC codes.

43.1 Overview

Table 43.1 lists the specifications of the CRC calculator, and Figure 43.1 shows a block diagram of the CRC calculator.

Table 43.1 CRC Specifications

Item	Description	
Data size	8 bits	32 bits
Data for CRC calculation*1	CRC codes are generated for any desired data in 8n-bit units (where n is a whole number)	CRC codes are generated for any desired data in 32n-bit units (where n is a whole number)
CRC processor unit	8-bit parallel processing	32-bit parallel processing
CRC generating polynomial	One of three generating polynomials is selectable • 8-bit CRC $X^8 + X^2 + X + 1$ • 16-bit CRC $X^{16} + X^{15} + X^2 + 1$ $X^{16} + X^{12} + X^5 + 1$	One of two generating polynomials is selectable • 32-bit CRC $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$ $X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$
CRC calculation switching	The order of the bits produced by CRC calculation can be switched for LSB first or MSB first communication	
Low power consumption	Module stop state can be set.	

Note 1. The circuit does not have a function to divide data for calculation into CRC calculation units. Write data in 8-bit or 32-bit units.

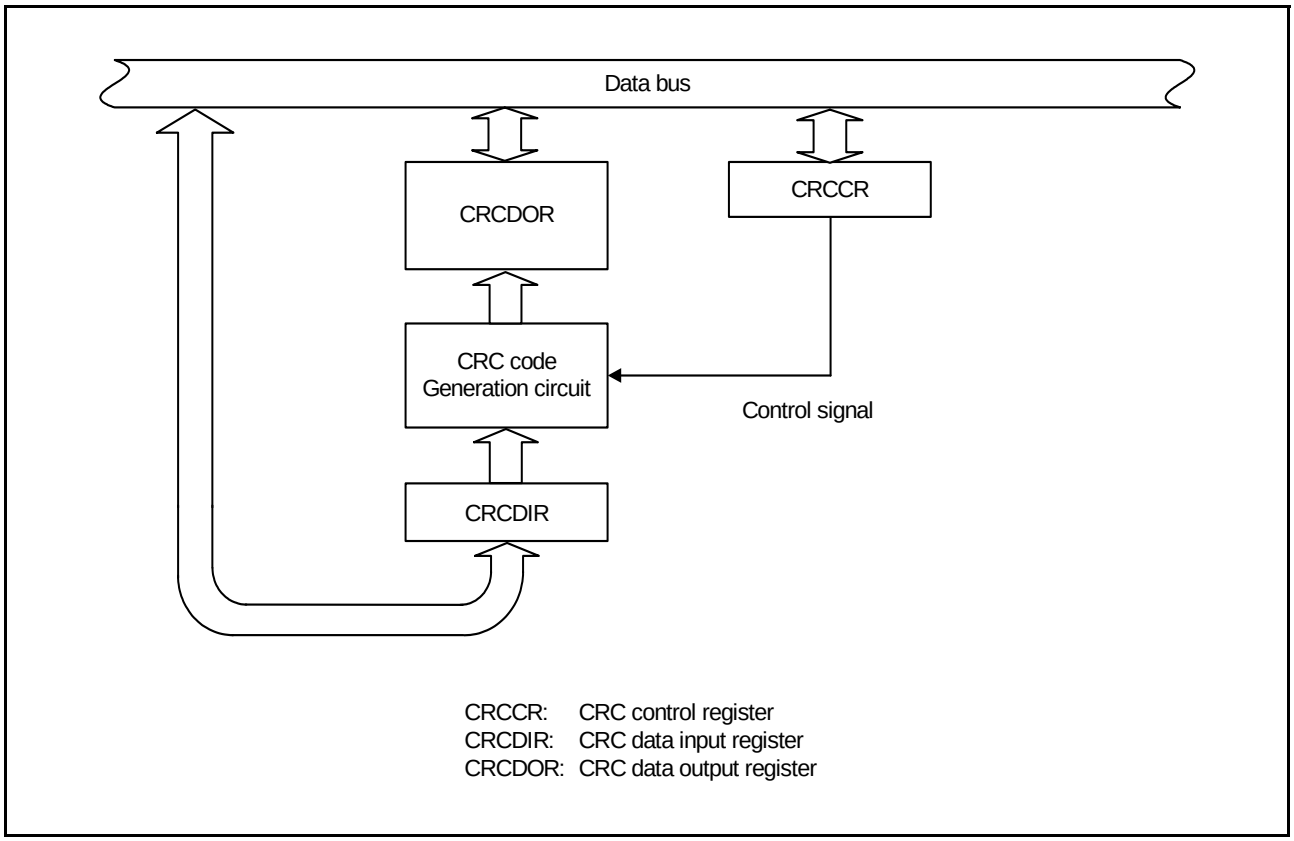


Figure 43.1 CRC Block Diagram

43.2 Register Descriptions

43.2.1 CRC Control Register (CRCCR)

Address(es): CRC.CRCCR 0008 8280h

	b7	b6	b5	b4	b3	b2	b1	b0
	DORCLR	LMS	—	—	—	GPS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	GPS[2:0]	CRC Generating Polynomial Switching	b2 b1 b0 0 0 0: No calculation is executed. 0 0 1: 8-bit CRC ($X^8 + X^2 + X + 1$) 0 1 0: 16-bit CRC ($X^{16} + X^{15} + X^2 + 1$) 0 1 1: 16-bit CRC ($X^{16} + X^{12} + X^5 + 1$) 1 0 0: 32-bit CRC ($X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$) 1 0 1: 32-bit CRC ($X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$) 1 1 0: No calculation is executed. 1 1 1: No calculation is executed.	R/W
b5 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	LMS	CRC Calculation Switching	0: Generates CRC for LSB first communication. 1: Generates CRC for MSB first communication.	R/W
b7	DORCLR	CRCDOR Register Clear	1: Clears the CRCDOR register. This bit is read as 0.	W*1

Note 1. Only 1 can be written.

LMS Bit (CRC Calculation Switching)

The setting this bit selects the order of the bits of generated CRC codes. The bit selects transmission of the lower-order byte of the CRC code first for LSB first communication, or the higher-order byte first for MSB first communication. For details on the transmission and reception of CRC codes, refer to section 43.3, Operation.

DORCLR Bit (CRCDOR Register Clear)

Write 1 to this bit so that the CRCDOR register is set to 0000 0000h.
This bit is read as 0. Writing 0 to this bit has no effect.

43.2.2 CRC Data Input Register (CRCDIR)

- When 32-bit CRC is selected

Address(es): CRC.CRCDIR 0008 8284h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

- When 16-bit or 8-bit CRC is selected

Address(es): CRC.CRCDIR 0008 8284h

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

The CRCDIR register is a readable and writable register. Write data for CRC calculation to this register.
When generating a 32-bit CRC, the CRCDIR register should be accessed in longword units. When generating a 8-bit or 16-bit CRC, the CRCDIR register should be accessed in byte units.

43.2.3 CRC Data Output Register (CRCDOR)

- When 32-bit CRC is selected

Address(es): CRC.CRCDOR 0008 8288h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

- When 16-bit CRC is selected

Address(es): CRC.CRCDOR 0008 8288h

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

- When 8-bit CRC is selected

Address(es): CRC.CRCDOR 0008 8288h

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

The CRCDOR register is a readable and writable register.
The value after a reset is 0000 0000h. When calculating with the initial value set to a value other than 0000 0000h, rewrite the CRCDOR register.
Writing data to the CRCDIR register stores result of calculation in the CRCDOR register. In addition, following communication data, when a CRC code is written to the CRCDIR register and if the calculation result is 0000 0000h, there is no error in the communication data.
When a 32-bit CRC is selected, the CRCDOR register should be accessed in longword units. When a 16-bit CRC is selected, the CRCDOR register should be accessed in word units. When an 8-bit CRC is selected, the CRCDOR register should be accessed in byte units.

43.3 Operation

The CRC calculator generates CRC codes for use in LSB first or MSB first communication.
The following shows examples of generating the CRC code for input data (F0h) using the 16-bit CRC generating polynomial ($X^{16} + X^{12} + X^5 + 1$). In these examples, the value of the CRC data output register (CRCDOR) is cleared by setting the DORCLR bit to 1 to set the initial value for CRC calculation to 0000 0000h.

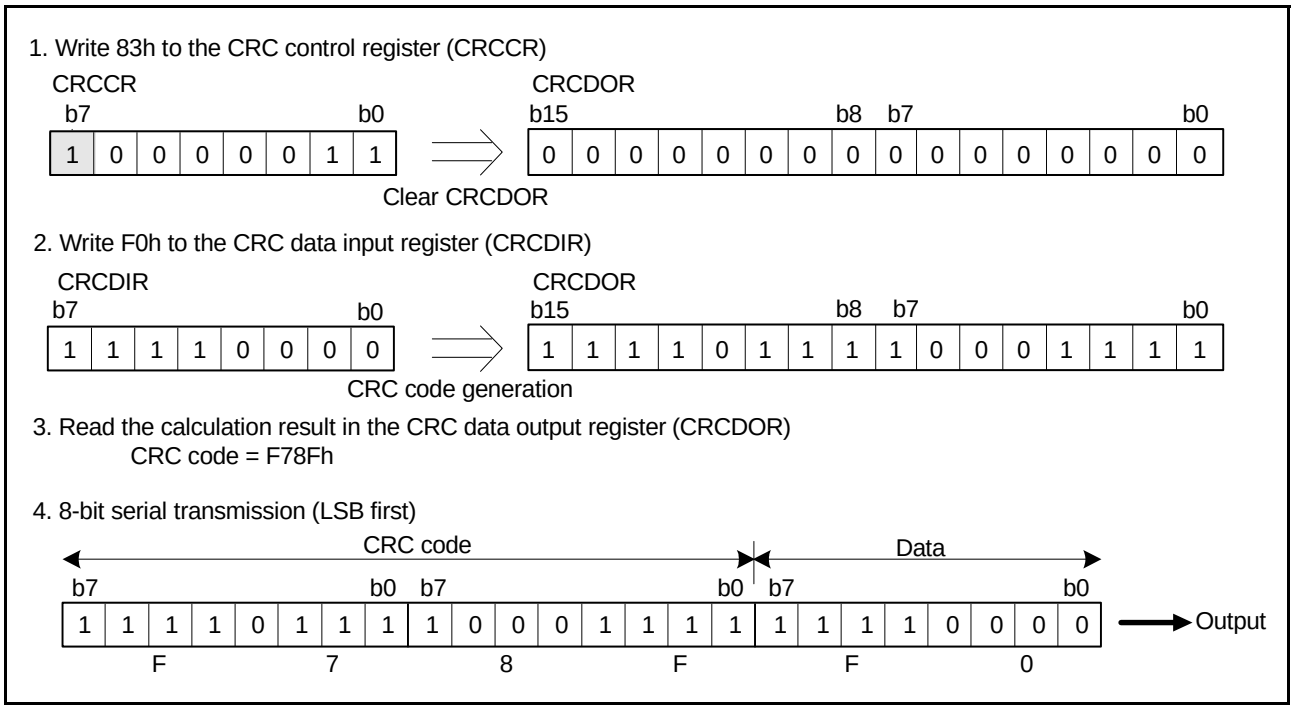


Figure 43.2 LSB First Data Transmission

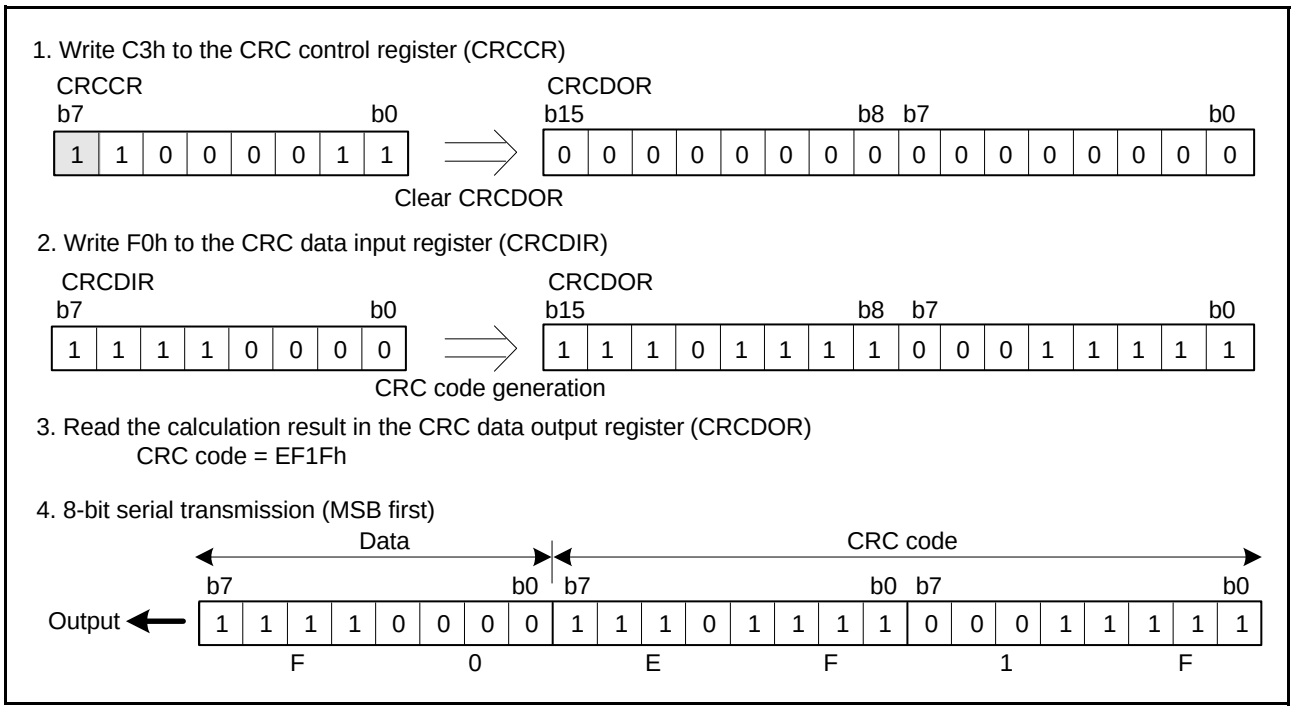


Figure 43.3 MSB First Data Transmission

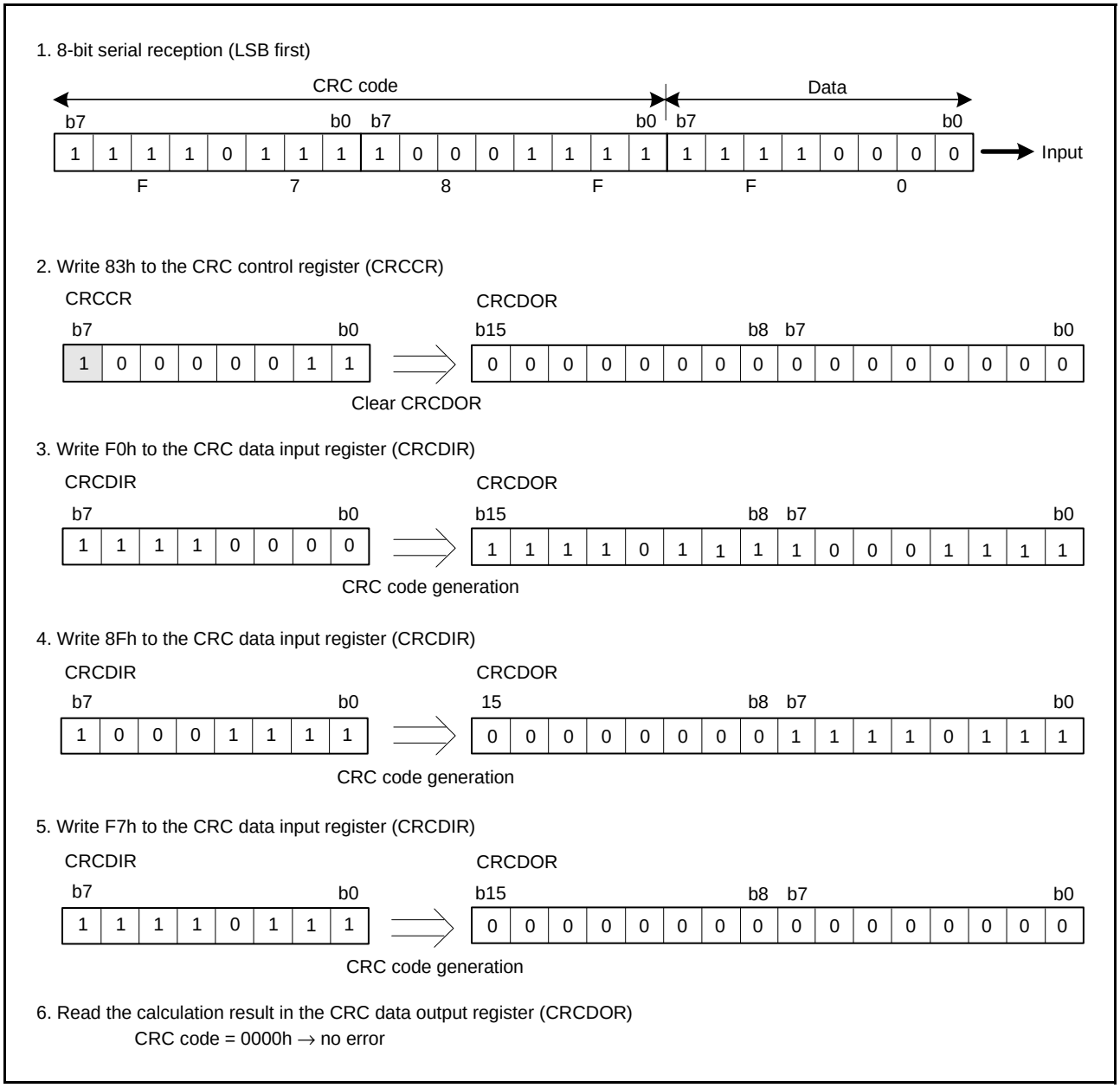


Figure 43.4 LSB First Data Reception

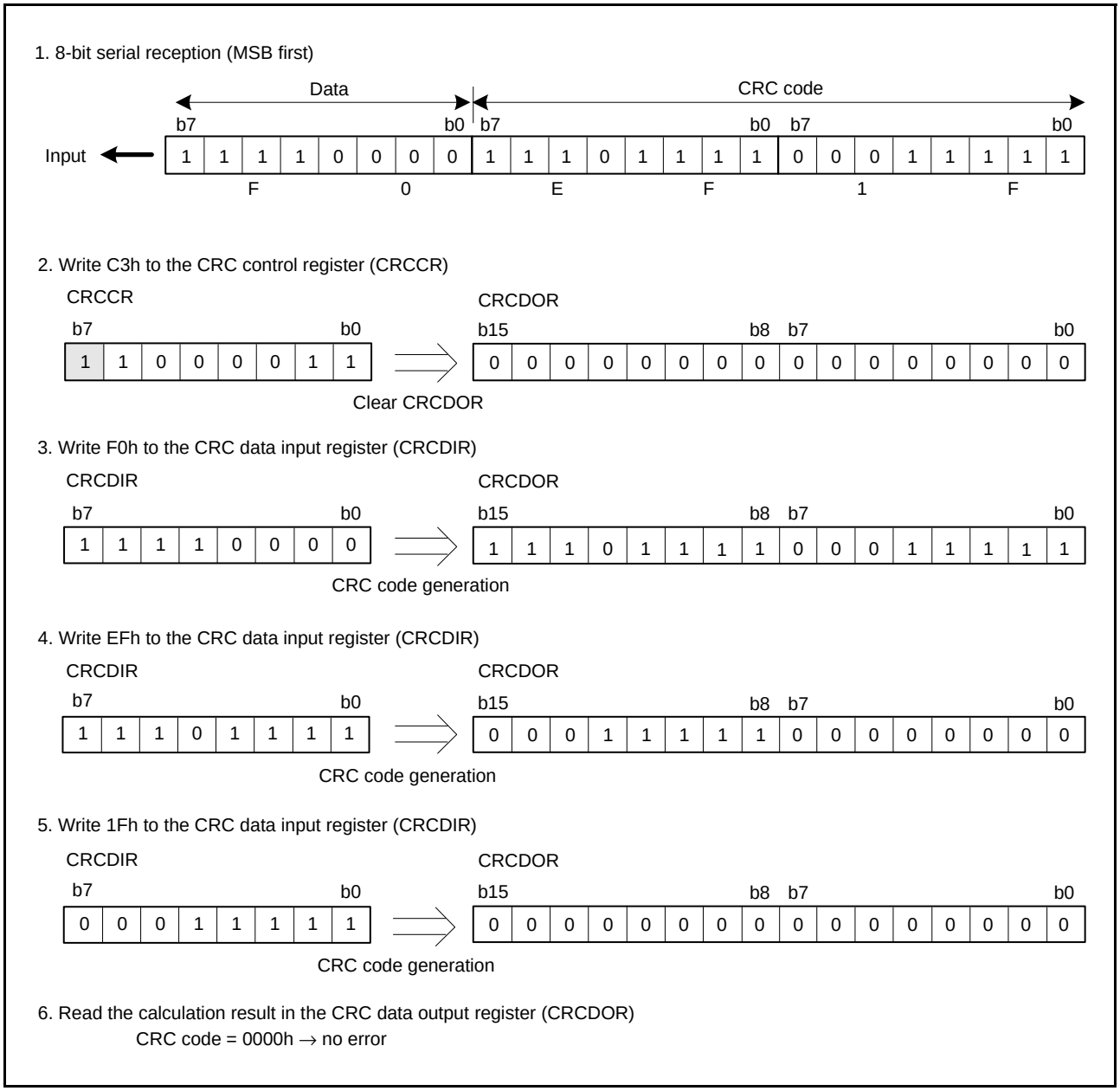


Figure 43.5 MSB First Data Reception

43.4 Usage Notes

43.4.1 Module Stop Function Setting

Operation of the CRC calculator can be disabled or enabled using the module stop control register B (MSTPCRB). After a reset, the CRC is in the module stop state. Register access is enabled by releasing the module stop state.
For details, refer to section 11, Low Power Consumption.

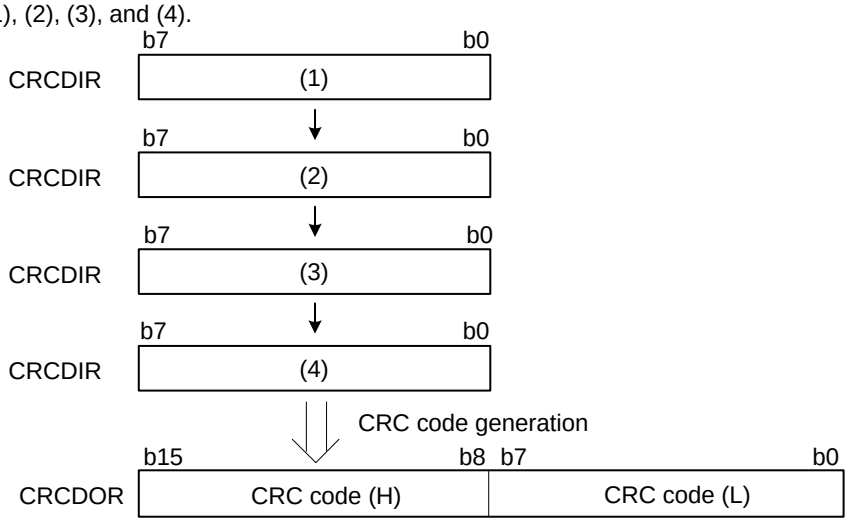
43.4.2 Note on Transmission

Note that the sequence of transmission for the CRC code differs according to whether transmission is LSB first or MSB first.

When transmitting 32-bit data

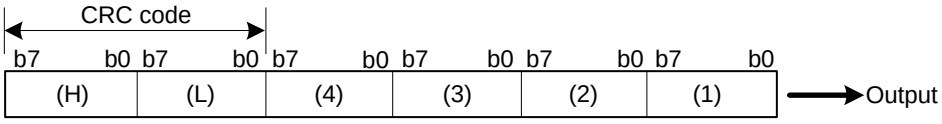
1. CRC code

After specifying the method for generation calculation, write data to the CRCDIR register in the order of (1), (2), (3), and (4).



2. Transmit data

(i) When transmission is LSB first



(ii) When transmission is MSB first

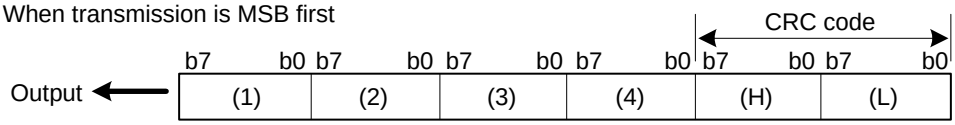


Figure 43.6 LSB First and MSB First Data Transmission

44. SD Host Interface (SDHI)

This MCU incorporates an SD host interface (SDHI) which is compliant with the SD Specifications. When developing host devices that are compliant with the SD Specifications, the user must enter into the SD Host/Ancillary Product License Agreement (SD HALA).

44.1 Overview

Table 44.1 lists the SDHI specifications.

Table 44.1 SDHI Specifications

Item	Description
SD bus interface	• Compatible with SD memory card and SDIO card (**NOT** compatible with the SPI bus interface, embedded SDIO shared bus, 8-bit SD bus, or SDIO suspend/resume functions) • Transfer bus mode selectable from 4-bit wide bus mode or 1-bit default bus mode • Compatible with SD, SDHC, and SDXC formats
Transfer modes	Selectable from high-speed mode or default speed mode
SDHI clock	The SDHI clock is generated by dividing peripheral module clock B (PCLKB) by *n*, where *n* = 1, 2, 4, 8, 16, 32, 64, 128, 256, or 512
Error check functions	• CRC7 (command/response) • CRC16 (transfer data)
Interrupt sources	• Card access interrupt (CACI) • SDIO access interrupt (SDACI) • Card detection interrupt (CDETI) • SD buffer access interrupt (SBFAI)
DMA transfer sources	• DMAC and DTC triggerable by the SBFAI interrupt • SD buffer is read and write accessible using the DMAC and DTC
Other functions	• Card detection • Write protection

Figure 44.1 shows a block diagram of the SDHI.

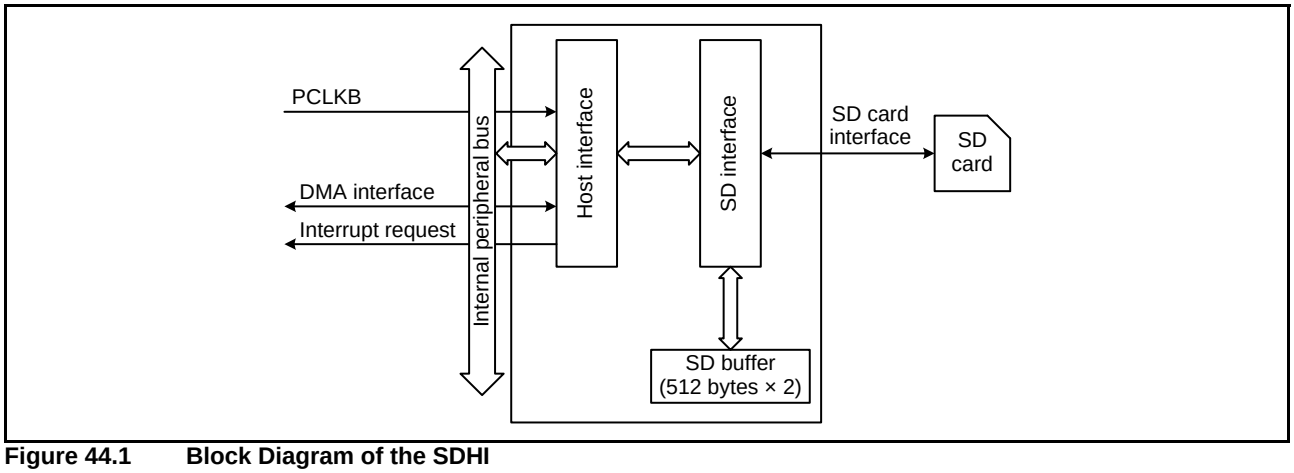


Figure 44.1 Block Diagram of the SDHI

Table 44.2 lists the pin configuration of the SDHI.

Table 44.2 Pin Configuration of the SDHI

Pin Name	I/O	Description
SDHI_CLK	Output	SDHI clock
SDHI_CMD	I/O	Command output, response input
SDHI_D0	I/O	Data 0 (DAT0)
SDHI_D1	I/O	Data 1 (DAT1), SDIO access interrupt
SDHI_D2	I/O	Data 2 (DAT2), read wait
SDHI_D3	I/O	Data 3 (DAT3), SD card detection
SDHI_CD	Input	SD card detection
SDHI_WP	Input	SD card write protection

44.2 Register Details

44.2.1 Command Register (SDCMD)

Address(es): SDHI.SDCMD 0008 AC00h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMD12AT[1:0]		TRSTP	CMDRW	CMDTP	RSPTP[2:0]			ACMD[1:0]		CMDIDX[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	CMDIDX[5:0]	Command Index Field Value Select	These bits configure the command index field value. The examples below include the bit values for the ACMD[1:0] bits. b7 b0 0 0 0 0 0 1 1 0: CMD6 0 0 0 1 0 0 1 0: CMD18 0 1 0 0 1 1 0 1: ACMD13	R/W
b7, b6	ACMD[1:0]	Command Type Select	b7 b6 0 0: CMD 0 1: ACMD Only set the values listed above.	R/W
b10 to b8	RSPTP[2:0]	Response Type Select *1	b10 b8 0 0 0: Normal mode. Depending on the command, the response type and transfer method are selected by setting the ACMD[1:0] bits and CMDIDX[5:0] bits. At this time, the values for b15 to b11 in this register are invalid. 0 1 1: Expansion mode and no response 1 0 0: Expansion mode and R1, R5, R6, or R7 response 1 0 1: Expansion mode and R1b response 1 1 0: Expansion mode and R2 response 1 1 1: Expansion mode and R3 or R4 response Only set the values listed above.	R/W
b11	CMDTP	Data Transfer Select *2	0: Command does not include data transfer (bc, bcr, or ac) 1: Command includes data transfer (adtc)	R/W
b12	CMDRW	Data Transfer Direction Select *3	0: Write data to the SD card 1: Read data from the SD card	R/W
b13	TRSTP	Block Transfer Select *3	0: Single block transferred 1: Multiple blocks transferred	R/W
b15, b14	CMD12AT[1:0]	CMD12 Automatic Issue Select *4	b15 b14 0 0: CMD12 is automatically issued during multi-block transfer 0 1: CMD12 is not automatically issued during multi-block transfer Only set the values listed above.	R/W
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Some commands cannot be used in normal mode. Refer to Table 44.3 and set the RSPTP[2:0] bits.
Note 2. The CMDTP bit is valid only when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b.
Note 3. Bits CMDRW and TRSTP are valid only when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b, and the CMDTP bit is 1.
Note 4. The CMD12AT[1:0] bits are valid only when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b, and the TRSTP bit is 1.

The command type and response type are set in the SDCMD register. The command type and transfer mode must be set when the RSPTP[2:0] bits are 011b, 100b, 101b, 110b, or 111b. The sequence starts when a value is written to this register. Refer to Table 44.3 for setting examples. Do not write to the SDCMD register when the SDSTS2.CBSY flag is 1.

Table 44.3 lists examples of SDCMD register settings.

Table 44.3 Examples of SDCMD Register Settings

Type	Command Symbol	SDCMD Register Setting	Remarks
CMD	CMD0	0000 0000h	
	CMD2	0000 0002h	
	CMD3	0000 0003h	
	CMD4	0000 0004h	
	CMD5	0000 0705h or 0000 0005h	
	CMD6	0000 1C06h or 0000 0006h	
	CMD7	0000 0007h	When the card is deselected, the SD card does not return a response, so the SDSTS2.RSPTO flag becomes 1.
	CMD8	0000 0408h or 0000 0008h	
	CMD9	0000 0009h	
	CMD10	0000 000Ah	
	CMD11	0000 040Bh or 0000 000Bh	
	CMD12	0000 000Ch	
	CMD13	0000 000Dh	
	CMD15	0000 000Fh	
	CMD16	0000 0010h	
	CMD17	0000 0011h	
	CMD18	0000 0012h	
	CMD20	0000 0514h or 0000 0014h	
	CMD24	0000 0018h	
	CMD25	0000 0019h	
	CMD27	0000 001Bh	
	CMD28	0000 001Ch	
	CMD29	0000 001Dh	
	CMD30	0000 001Eh	
	CMD32	0000 0020h	
	CMD33	0000 0021h	
	CMD38	0000 0026h	
	CMD42	0000 002Ah	
	CMD52	0000 0434h or 0000 0034h	
	CMD53	0000 1C35h	Single block read
		0000 0C35h	Single block write
		0000 7C35h	Multi-block read
		0000 6C35h	Multi-block write
		0000 0035h	The setting to the left can be used regardless of the transfer being single block or multi-block. However, the MSB in the SDARG register (RW flag) must be set to 0 when reading and 1 when writing.
	CMD55	0000 0037h	
	CMD56	0000 0038h	
ACMD	ACMD6	0000 0046h	
	ACMD13	0000 004Dh	
	ACMD22	0000 0056h	
	ACMD23	0000 0057h	
	ACMD41	0000 0069h	
	ACMD42	0000 006Ah	
	ACMD51	0000 0073h	

44.2.2 Argument Register (SDARG)

Address(es): SDHI.SDARG 0008 AC08h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The SDARG register is used for setting the argument field value. Set the SDARG register before setting the SDCMD register. The argument field value of the automatically issued CMD12 is 0000 0000h regardless of the SDARG register value.

44.2.3 Data Stop Register (SDSTOP)

Address(es): SDSTOP 0008 AC10h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	—	SDBLKCNTEN
Value after reset:	0	0	0	0	0	0	0	0

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

Bit	Symbol	Bit Name	Description	R/W
b0	STP	Transfer Stop	Data transfer stops when this bit is set to 1.	R/W
b7 to b1	—	Reserved	These bits are 0 when read and cannot be modified.	R
b8	SDBLKCNTEN	Block Count Register Value Select *1	0: SDBLKCNT register value is invalid 1: SDBLKCNT register value is valid	R/W
b31 to b9	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Do not rewrite this bit when the SDSTS2.CBSY flag is 1.

The SDSTOP register stops data transfer. During a multi-block transfer sequence, the SDBLKCNT register value (number of blocks to be transferred) can be set to valid or invalid by setting the SDSTOP register.

STP Bit (Transfer Stop)

When setting the STP bit to 1, set it after the SDSTS1.RSPEND flag becomes 1; when setting the STP bit to 0, set it after the SDSTS1.ACEND flag becomes 1. After a command sequence is complete, the SDHI does not issue CMD12 and the SDSTS1.ACEND flag does not become 1 even if the STP bit is set to 1. When the SDHI is in the busy state after receiving the R1b response, the SDHI does not issue CMD12 even if the STP bit is 1, and after the SDHI is released from

the busy state, the SDSTS1.ACEND flag becomes 1.

- Performing a multi-block transfer
  When the STP bit is set to 1, the SDHI issues CMD12, and the command sequence is stopped. The SD buffer can be accessed even after the STP bit is set to 1, but a buffer access error occurs and the SDSTS2.ILW flag or SDSTS2.ILR flag becomes 1. If the command sequence stops due to a communication error or a timeout, the SDHI does not issue CMD12.
- Performing a single block transfer
  When the STP bit is set to 1 during a single block write access, if there is no data in the SD buffer, the SDSTS1.ACEND flag becomes 1. If there is data in the SD buffer, after the SDHI is released from the busy state, the SDSTS1.ACEND flag becomes 1. When the STP bit is set to 1 during a single block read access, the SDSTS1.ACEND flag becomes 1. Also, CMD12 is not issued even if the STP bit is set to 1 during a single block read access or single block write access.

SDBLKCNTEN Bit (Block Count Register Value Select)

If the SDBLKCNTEN bit is 1 during a multi-block transfer sequence, the SDHI automatically issues CMD12. When the SDCMD.RSPTP[2:0] bits are set to 000b and CMD18 or CMD25 is issued, or if the SDCMD.RSPTP[2:0] bits are set to 011b, 100b, 101b, 110b, or 111b and the SDCMD.TRSTP bit is 1 (multiple blocks transferred), if the SDCMD.CMD12AT[1:0] bits are 00b (CMD12 is automatically issued during multi-block transfer), and the number of transfer blocks reaches the value set in the SDBLKCNT register, the SDHI automatically issues CMD12.
If the command sequence is stopped by a communication error or a timeout, CMD12 is not automatically issued.

44.2.4 Block Count Register (SDBLKCNT)

Address(es): SDHI.SDBLKCNT 0008 AC14h

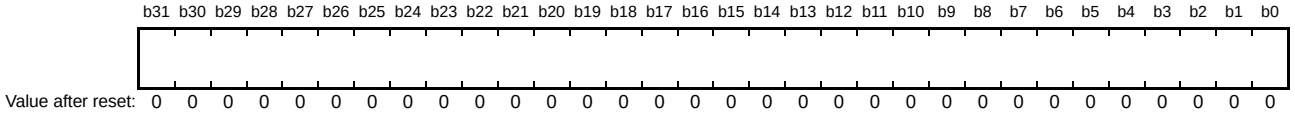
	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

When performing a multi-block transfer, SDBLKCNT is a readable/writable register used to set the number of blocks to be transferred. For example, when the register value is 0000 0001h, 1 block is transferred; when the register value is 0000 FFFFh, 65,535 blocks are transferred; and when the register value is FFFF FFFFh, 4,294,967,295 blocks are transferred. Do not set this register to 0000 0000h. Do not rewrite the SDBLKCNT register when the SDSTS2.CBSY flag is 1.

44.2.5 Response Register 10 (SDRSP10), Response Register 32 (SDRSP32), Response Register 54 (SDRSP54), Response Register 76 (SDRSP76)

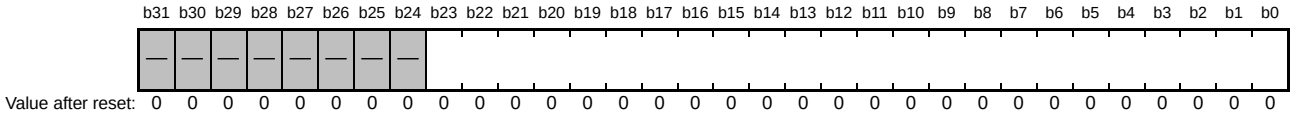
- SDRSP10, SDRSP32, SDRSP54

Address(es): SDHI.SDRSP10 0008 AC18h, SDHI.SDRSP32 0008 AC20h, SDHI.SDRSP54 0008 AC28h



- SDRSP76

Address(es): SDHI.SDRSP76 0008 AC30h



Bit	Symbol	Bit Name	Description	R/W
b23 to b0	—	—	This register stores the response from the SD card.	R
b31 to b24	—	Reserved	These bits are 0 when read.	R

Registers SDRSP10, SDRSP32, SDRSP54, and SDRSP76 are read-only registers that store the response from the SD card. Depending on the type of response from the SD card, the SDHI divides and stores the response among the four registers.

Table 44.4 lists the correspondence between the response type and its storage destination.

Table 44.4 Correspondence Between the Response Type and Its Storage Destination

Response Type	SDRSP10 Register	SDRSP32 Register	SDRSP54 Register	SDRSP76 Register
R1	[39:8]	—	[39:8] *1	—
R1b	[39:8]	—	[39:8] *1	—
R2	[39:8]	[71:40]	[103:72]	[127:104]
R3	[39:8]	—	—	—
R4	[39:8]	—	—	—
R5	[39:8]	—	—	—
R6	[39:8]	—	—	—
R7	[39:8]	—	—	—

Note 1. The response for CMD18 and CMD25 is stored in registers SDRSP10 and SDRSP54. Therefore, even if the SDRSP10 register is overwritten with the response for the automatically transmitted CMD12, the response for CMD18 or CMD25 can be confirmed by reading the SDRSP54 register.

44.2.6 SD Status Register 1 (SDSTS1)

Address(es): SDHI.SDSTS1 0008 AC38h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	SDD3MON	SDD3IN	SDD3RM
Value after reset:	0	0	0	0	0	x	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	SDWPMON	—	SDCDMON	SDCDIN	SDCDRM	ACEND	—	RSPEND
Value after reset:	x	0	x	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RSPEND	Response End Detection Flag	0: Response end is not detected 1: Response end is detected	R/(W) *1
b1	—	Reserved	This bit is 0 when read and cannot be modified.	R
b2	ACEND	Access End Detection Flag	0: Access end is not detected 1: Access end is detected	R/(W) *1
b3	SDCDRM	SDHI_CD Removal Flag	0: SD card removal not detected by the SDHI_CD pin 1: SD card removal detected by the SDHI_CD pin	R/(W) *1
b4	SDCDIN	SDHI_CD Insertion Flag	0: SD card insertion not detected by the SDHI_CD pin 1: SD card insertion detected by the SDHI_CD pin	R/(W) *1
b5	SDCDMON	SDHI_CD Pin Monitor Flag	0: SDHI_CD pin level is high *2 1: SDHI_CD pin level is low *2	R
b6	—	Reserved	This bit is 0 when read and cannot be modified.	R
b7	SDWPMON	SDHI_WP Pin Monitor Flag	0: SDHI_WP pin level is high 1: SDHI_WP pin level is low	R
b8	SDD3RM	SDHI_D3 Removal Flag	0: SD card removal not detected by the SDHI_D3 pin 1: SD card removal detected by the SDHI_D3 pin	R/(W) *1
b9	SDD3IN	SDHI_D3 Insertion Flag	0: SD card insertion not detected by the SDHI_D3 pin 1: SD card insertion detected by the SDHI_D3 pin	R/(W) *1
b10	SDD3MON	SDHI_D3 Pin Monitor Flag	0: SDHI_D3 pin level is low 1: SDHI_D3 pin level is high	R
b31 to b11	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. The flag does not change even if set to 1. Writing 0 changes the flag value to 0.
Note 2. The flag changes when the pin level continues for the period set in the SDOPT.CTOP[3:0] bits or longer.

The SDSTS1 register indicates the detection of a response end or access end for a command sequence. The SDSTS1 register also indicates the detection SD card insertion/removal, and indicates the write protection status.
During a multi-block transfer sequence, if CMD12 or CMD52 (SDIO abort) is issued, the ACEND flag becomes 1, but the RSPEND flag remains set to 0.
If the command sequence is stopped due to a communication error or timeout, the ACEND flag or RSPEND flag becomes 1.
After a reset is released, the SDD3MON flag, SDD3IN flag, and SDD3RM flag values are changed according to the status of the SDHI_D3 pin, and their values are changed when data is being transferred in wide bus mode.
Flags to be cleared should be set to 0; flags not being cleared should be set to 1.

RSPEND Flag (Response End Detection Flag)

— This flag becomes 1 under any of the following conditions:

- A response is received.
- A command that does not have a response is issued.
- After the R1b response is received, the SDHI is released from the busy state.
- During a multi-block transmission, after the SDIOMD.C52PUB bit is set to 1, the CMD52 response is received.
- A communication error or timeout causes the command sequence to abort.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

Note: When a command is issued that is absent of data transfer, the RSPEND flag becomes 1 after the command sequence ends.

ACEND Flag (Access End Detection Flag)

— This flag becomes 1 under any of the following conditions:

- During a single block read sequence, the SD buffer read access is completed.
- During a multi-block read sequence, the last block is read from the SD buffer.
- During a multi-block read sequence, if CMD12 is automatically issued, data is read from the SD buffer, and the response for CMD12 is received.
- During a single block write sequence, after a CRC status token is received, the SDHI is released from the busy state.
- During a multi-block write sequence, after a CRC status token is received for the last block, the SDHI is released from the busy state.
- During a multi-block write sequence, when CMD12 is automatically issued, a response busy of the automatically issued CMD12 is received.
- During a multi-block read sequence, when CMD12 is automatically issued, after setting the SDSTOP.STP bit to 1, a response of the automatically issued CMD12 is received.
- During a multi-block write sequence, when CMD12 is automatically issued, after setting the SDSTOP.STP bit to 1, a response busy of the automatically issued CMD12 is received.
- During a multi-block read sequence, after the SDIOMD.IOABT bit is set to 1, the response for CMD52 is received.
- During a multi-block write sequence, after the SDIOMD.IOABT bit is set to 1, the response for CMD52 is received.
- A communication error or timeout causes the command sequence to abort.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

Note: The ACEND flag becomes 1 after the command sequence ends.

SDCDRM Flag (SDHI_CD Removal Flag)

— This flag becomes 1 under the following condition:

- The SDHI_CD pin changes from low to high, and the high period is the period set in the SDOPT.CTOP[3:0] bits or longer.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

SDCDIN Flag (SDHI_CD Insertion Flag)

— This flag becomes 1 under the following condition:

- The SDHI_CD pin changes from high to low, and the low period is the period specified in the SDOPT.CTOP[3:0] bits or longer.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

SDD3RM Flag (SDHI_D3 Removal Flag)

— This flag becomes 1 under the following condition:

- The SDHI_D3 pin changes from high to low, and the low period is at least two PCLKB cycles.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

SDD3IN Flag (SDHI_D3 Insertion Flag)

— This flag becomes 1 under the following condition:

- The SDHI_D3 pin changes from low to high, and the high period is at least two PCLKB cycles.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

44.2.7 SD Status Register 2 (SDSTS2)

Address(es): SDHI.SDSTS2 0008 AC3Ch

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	ILA	CBSY	SDCLKCREN	—	—	—	BWE	BRE
Value after reset:	0	0	1	0	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	SDD0MON	RSPTO	ILR	ILW	DTO	ENDE	CRCE	CMDE
Value after reset:	x	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMDE	Command Error Detection Flag	0: Command error not detected 1: Command error detected	R/(W) *1
b1	CRCE	CRC Error Detection Flag	0: CRC error not detected 1: CRC error detected	R/(W) *1
b2	ENDE	End Bit Error Detection Flag	0: End bit error not detected 1: End bit error detected	R/(W) *1
b3	DTO	Data Timeout Detection Flag	0: Data timeout not detected 1: Data timeout detected	R/(W) *1
b4	ILW	SDBUFR Illegal Write Access Detection Flag	0: Illegal write access to the SDBUFR register not detected 1: Illegal write access to the SDBUFR register detected	R/(W) *1
b5	ILR	SDBUFR Illegal Read Access Detection Flag	0: Illegal read access to the SDBUFR register not detected 1: Illegal read access to the SDBUFR register detected	R/(W) *1
b6	RSPTO	Response Timeout Detection Flag	0: Response timeout not detected 1: Response timeout detected	R/(W) *1
b7	SDD0MON	SDHI_D0 Pin Status Flag	0: SDHI_D0 pin is low 1: SDHI_D0 pin is high	R
b8	BRE	SDBUFR Read Enable Flag	0: Read access to the SDBUFR register disabled 1: Read access to the SDBUFR register enabled	R/(W) *1
b9	BWE	SDBUFR Write Enable Flag	0: Write access to the SDBUFR register disabled 1: Write access to the SDBUFR register enabled	R/(W) *1
b10	—	Reserved	This bit is 0 when read and cannot be modified.	R
b11	—	Reserved	This bit is 0 when read. Set it to 1 when writing.	R/W
b12	—	Reserved	This bit is 0 when read and cannot be modified.	R
b13	SDCLKCREN	SDCLKCR Write Enable Flag	0: SD bus (CMD and DAT lines) is busy, so write access to the SDCLKCR.CLKEN bit and CLKSEL[7:0] bits is disabled. 1: SD bus (CMD and DAT lines) is not busy, so write access to the SDCLKCR.CLKEN bit and CLKSEL[7:0] bits is enabled.	R
b14	CBSY	Command Sequence Status Flag	0: Command sequence completed 1: Command sequence in progress (busy)	R
b15	ILA	Illegal Access Error Detection Flag	0: Illegal access error not detected 1: Illegal access error detected	R/(W) *1
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. The flag does not change even if set to 1. Writing 0 changes the flag value to 0.

The SDSTS2 register indicates the status of the SD buffer and the status of the SD card. Flags to be cleared should be set to 0; flags not being cleared should be set to 1.

CMDE Flag (Command Error Detection Flag)

The command sequence is stopped when a command error occurs. When the SDIOMD.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence will not be completed. Perform the error processing shown in section 44.3.6.8 or section 44.3.6.9 and complete the command sequence.

— This flag becomes 1 under any of the following conditions:

- The command index field value for the command transmitted differs from the command index field value for the response received.
- The command index field value for the automatically issued CMD12 or CMD52 (which are the commands to stop transfer) differs from the command index field value for the response received.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

CRCE Flag (CRC Error Detection Flag)

The command sequence is stopped when a CRC error occurs. When the SDIOMD.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence will not be completed. Perform the error processing shown in section 44.3.6.8 or section 44.3.6.9 and complete the command sequence.

— This flag becomes 1 under any of the following conditions:

- The received CRC status token is in error (the value of the CRC status is a value other than 010b).
- The read data contains a CRC error.
- The response contains a CRC error.
- The response for the automatically issued CMD12 or CMD52 (which are the commands to stop transfer) contains a CRC error.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

ENDE Flag (End Bit Error Detection Flag)

The command sequence is stopped when an end bit error occurs. When the SDIOMD.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence will not be completed. Perform the error processing shown in section 44.3.6.8 or section 44.3.6.9 and complete the command sequence.

— This flag becomes 1 under any of the following conditions:

- The response length is in error (the end bit could not be detected).
- The read data length is in error (the end bit of the enabled bit could not be detected).
- The CRC status token length is in error (the end bit could not be detected).
- The response length for the automatically issued CMD12 or CMD52 (which are the commands to stop transfer) contains an error (the end bit could not be detected).

— This flag becomes 0 under the following condition:

- The flag is set to 0.

DTO Flag (Data Timeout Detection Flag)

This flag indicates that the data expected to be received during the period specified (set in the SDOPT.TOP[3:0] bits) was not received. However, response timeouts are excluded. The command sequence stops when a data timeout occurs.

— This flag becomes 1 under any of the following conditions:

- After the R1b response is received, the SDHI is busy for the period specified or longer.
- After the CRC status token is received, the SDHI is busy for the period specified or longer.
- After data is written, the CRC status token is not received even after the period specified elapsed.
- After a read command is issued, the read data is not received even after the period specified elapsed.
- After CMD12 is issued during a command sequence, the SDHI is busy for the period specified or longer.
- After the read data is received, the next read data is not received even after the period specified elapsed.
- After the SDHI exits the read wait state, the next read data is not received even after the period specified elapsed.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

ILW Flag (SDBUFR Illegal Write Access Detection Flag)

— This flag becomes 1 under any of the following conditions:

- A value is written to the SDBUFR register while the SDHI is not in the data read or data write command state.
- A value is written to the SDBUFR register while the SD buffer is full.
- A value is written to the SDBUFR register while the CRC status token or CRC status token length is in error.
- After the CRC status token is received, a value is written to the SDBUFR register if the SDHI is busy for the period set in bits SDOPT.TOP[3:0] or longer.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

ILR Flag (SDBUFR Illegal Read Access Detection Flag)

— This flag becomes 1 under any of the following conditions:

- The SDBUFR register is read while the SD buffer is empty.
- The value read from the SDBUFR register includes a CRC error or end bit error.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

RSPTO Flag (Response Timeout Detection Flag)

The command sequence is stopped when a response timeout occurs. When the SDIOMD.C52PUB bit is set to 1 and CMD52 is automatically issued, if a communication error or response timeout occurs, the command sequence will not be completed. Perform the error processing shown in section 44.3.6.8 or section 44.3.6.9 and complete the command sequence.

— This flag becomes 1 under the following condition:

- A response is not received even after 640 SDHI clock cycles or more have elapsed (including the response for the automatically issued CMD12 or CMD52 (which are the commands to stop transfer)).

— This flag becomes 0 under the following condition:

- The flag is set to 0.

SDD0MON Flag (SDHI_D0 Pin Status Flag)

This flag indicates the status of the SDHI_D0 pin. After an erase command is issued, if the DTO flag is 1 and the RSPTO flag is 0, polling can be used to monitor the SDD0MON flag change from 0 to 1, and check that the erase command sequence is complete. If a communication error or timeout occurs during the write sequence, the SDHI_D0 pin may remain low. When the SDHI clock is stopped, the value before the SDHI clock was stopped is retained.

BRE Flag (SDBUFR Read Enable Flag)

— This flag becomes 1 under any of the following conditions:

- During a single block transfer, the data size set in the SDSIZE.LEN[9:0] bits is stored in the SD buffer.
- During a multi-block transfer, the data size set in the SDSIZE.LEN[9:0] bits is stored in one of the two SD buffers.

— This flag becomes 0 under any of the following conditions:

- The bit is set to 0.
- DMA transfer is used to read 1 block of data from the SD buffer.

If the CPU is used to read data from the SDBUFR register, set the BRE flag to 0 before reading the data size *1 set in the SDSIZE.LEN[9:0] bits. Even if the block of data read contains a CRC error or end bit error, the data is stored in the SD buffer and the BRE flag becomes 1.

Note 1. If the transfer data size set in the SDSIZE.LEN[9:0] bits is an odd number, the odd numbered byte is ignored. Refer to section 44.5.2, SDBUFR Register Illegal Write Error for details.

BWE Flag (SDBUFR Write Enable Flag)

— This flag becomes 1 under any of the following conditions:

- During a single block transfer, the SD buffer is empty.
- During a multi-block transfer, bank 1 or bank 2 of the SD buffer is empty.

— This flag becomes 0 under any of the following conditions:

- The flag is set to 0.
- DMA transfer is used to write 1 block of data to the SD buffer.

If the CPU is used to write data to the SDBUFR register, set the BWE flag to 0 before writing the data size *1 set in the SDSIZE.LEN[9:0] bits.

Note 1. If the transfer data size set in the SDSIZE.LEN[9:0] bits is an odd number, the odd numbered byte is ignored. Refer to section 44.5.2, SDBUFR Register Illegal Write Error for details.

SDCLKCREN Flag (SDCLKCR Write Enable Flag)

When a value is written to the SDCMD register, the SDHI starts the command sequence, the SDSTS2.CBSY flag becomes 1, and the SDSTS2.SDCLKCREN flag becomes 0. When the command sequence is complete, after the SDSTS2.CBSY flag becomes 0, eight cycles of the SDHI clock elapse and then the SDSTS2.SDCLKCREN flag becomes 1.

ILA Flag (Illegal Access Error Detection Flag)

— This flag becomes 1 under any of the following conditions:

- A value is written to the SDCMD register when the SDSTS2.CBSY flag is 1.
- The SDCMD.CMDTP bit is set to 1 (command accompanying data transfer), the SDCMD.ACMD[1:0] bits are set to 00b, and the SDCMD.CMDIDX[5:0] bits are set to 001100b (CMD12).

— This flag becomes 0 under the following condition:

- The flag is set to 0.

44.2.8 SD Interrupt Mask Register 1 (SDIMSK1)

Address(es): SDHI.SDIMSK1 0008 AC40h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	SDD3INM	SDD3RMM
Value after reset:	0	0	0	0	0	0	1	1

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SDCDINM	SDCDRMM	ACENDM	—	RSPENDM
Value after reset:	0	0	0	1	1	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	RSPENDM	Response End Interrupt Request Mask	0: Response end interrupt request is not masked 1: Response end interrupt request is masked	R/W
b1	—	Reserved	This bit is 0 when read and cannot be modified.	R
b2	ACENDM	Access End Interrupt Request Mask	0: Access end interrupt request is not masked 1: Access end interrupt request is masked	R/W
b3	SDCDRMM	SDHI_CD Removal Interrupt Request Mask	0: SD card removal interrupt request by the SDHI_CD pin not masked 1: SD card removal interrupt request by the SDHI_CD pin masked	R/W
b4	SDCDINM	SDHI_CD Insertion Interrupt Request Mask	0: SD card insertion interrupt request by the SDHI_CD pin not masked 1: SD card insertion interrupt request by the SDHI_CD pin masked	R/W
b7 to b5	—	Reserved	These bits are 0 when read and cannot be modified.	R
b8	SDD3RMM	SDHI_D3 Removal Interrupt Request Mask	0: SD card removal interrupt request by the SDHI_D3 pin not masked 1: SD card removal interrupt request by the SDHI_D3 pin masked	R/W
b9	SDD3INM	SDHI_D3 Insertion Interrupt Request Mask	0: SD card insertion interrupt request by the SDHI_D3 pin not masked 1: SD card insertion interrupt request by the SDHI_D3 pin masked	R/W
b31 to b10	—	Reserved	These bits are 0 when read and cannot be modified.	R

The SDIMSK1 register enables and disables the interrupt requests from the status flags in the SDSTS1 register. Refer to Table 44.8, Interrupt Sources for details on the relationship between the status flags and the requested interrupt source.

44.2.9 SD Interrupt Mask Register 2 (SDIMSK2)

Address(es): SDHI.SDIMSK2 0008 AC44h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	ILAM	—	—	—	—	—	BWEM	BREM
Value after reset:	1	0	0	0	1	0	1	1

	b7	b6	b5	b4	b3	b2	b1	b0
	—	RSPTOM	ILRM	ILWM	DTTOM	ENDEM	CRCEM	CMDEM
Value after reset:	0	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	CMDEM	Command Error Interrupt Request Mask	0: Command error interrupt request not masked 1: Command error interrupt request masked	R/W
b1	CRCEM	CRC Error Interrupt Request Mask	0: CRC error interrupt request not masked 1: CRC error interrupt request masked	R/W
b2	ENDEM	End Bit Error Interrupt Request Mask	0: End bit detection error interrupt request not masked 1: End bit detection error interrupt request masked	R/W
b3	DTTOM	Data Timeout Interrupt Request Mask	0: Data timeout interrupt request not masked 1: Data timeout interrupt request masked	R/W
b4	ILWM	SDBUFR Register Illegal Write Interrupt Request Mask	0: Illegal write detection interrupt request for the SDBUFR register not masked 1: Illegal write detection interrupt request for the SDBUFR register masked	R/W
b5	ILRM	SDBUFR Register Illegal Read Interrupt Request Mask	0: Illegal read detection interrupt request for the SDBUFR register not masked 1: Illegal read detection interrupt request for the SDBUFR register masked	R/W
b6	RSPTOM	Response Timeout Interrupt Request Mask	0: Response timeout interrupt request not masked 1: Response timeout interrupt request masked	R/W
b7	—	Reserved	This bit is 0 when read and cannot be modified.	R
b8	BREM	BRE Interrupt Request Mask	0: Read enable interrupt request for the SDBUFR register not masked 1: Read enable interrupt request for the SDBUFR register masked	R/W
b9	BWEM	BWE Interrupt Request Mask	0: Write enable interrupt request for the SDBUFR register not masked 1: Write enable interrupt request for the SDBUFR register masked	R/W
b10	—	Reserved	This bit is 0 when read and cannot be modified.	R
b11	—	Reserved	This bit is 1 when read and cannot be modified.	R
b14 to b12	—	Reserved	These bits are 0 when read and cannot be modified.	R
b15	ILAM	Illegal Access Error Interrupt Request Mask	0: Illegal access error interrupt request not masked 1: Illegal access error interrupt request masked	R/W
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. When the SDIMSK2.BWEM bit is 0 or the SDIMSK2.BREM bit is 0, set the SDDMAEN.DMAEN bit to 0. When the SDDMAEN.DMAEN bit is 1, set the SDIMSK2.BWEM bit to 1 and the SDIMSK2.BREM bit to 1.

The SDIMSK2 register enables and disables the interrupt requests from the status flags in the SDSTS2 register. Refer to Table 44.8 for details on the relationship between the status flags and the requested interrupt source.

44.2.10 SDHI Clock Control Register (SDCLKCR)

Address(es): SDHI.SDCLKCR 0008 AC48h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	CLKCTRLEN	CLKEN
Value after reset:	0	0	0	0	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	CLKSEL[7:0]							
Value after reset:	0	0	1	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CLKSEL[7:0]	SDHI Clock Frequency Select *1	b7 b0 0 0 0 0 0 0 0 0: PCLKB divided by 2 0 0 0 0 0 0 0 1: PCLKB divided by 4 0 0 0 0 0 0 1 0: PCLKB divided by 8 0 0 0 0 0 1 0 0: PCLKB divided by 16 0 0 0 0 1 0 0 0: PCLKB divided by 32 0 0 0 1 0 0 0 0: PCLKB divided by 64 0 0 1 0 0 0 0 0: PCLKB divided by 128 0 1 0 0 0 0 0 0: PCLKB divided by 256 1 0 0 0 0 0 0 0: PCLKB divided by 512 1 1 1 1 1 1 1 1: PCLKB *2 Only set the values listed above.	R/W
b8	CLKEN	SDHI Clock Output Control *1	0: SDHI clock output is disabled (SDHI_CLK signal fixed low) 1: SDHI clock output enabled	R/W
b9	CLKCTRLEN	SDHI Clock Output Automatic Control Select	0: Automatic control of SDHI clock output disabled 1: Automatic control of SDHI clock output enabled	R/W
b31 to b10	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Bits CLKSEL[7:0] and CLKEN cannot be write accessed when the SDSTS2.SDCLKCREN flag is 0.
Note 2. When setting the CLKSEL[7:0] bits to 11111111b or when changing the CLKSEL[7:0] bit values from 11111111b to another value, perform the following steps:
(1) Set the CLKEN bit to 0. Do not change the other bit values.
(2) Change the CLKSEL[7:0] bit values. Do not change the other bit values.
(3) Set the CLKEN bit to 1. Do not change the other bit values.

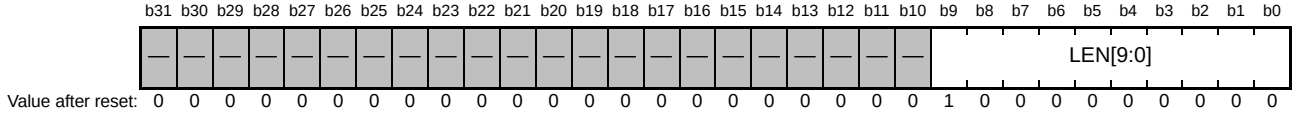
The SDCLKCR register controls the SDHI clock frequency settings and output. Set the CLKEN bit to 1 before writing to the SDCMD register to start a command sequence. Do not write access the SDCLKCR register when the SDSTS2.SDCLKCREN flag is 0.

CLKCTRLEN Bit (SDHI Clock Output Automatic Control Select)

The SDHI clock output automatic control is a function for starting and stopping SDHI clock output only during a command sequence. When this function is enabled, the SDHI starts outputting the SDHI clock after a value is set to the SDCMD register. After the command sequence is complete and eight cycles of the SDHI clock elapse, the SDHI stops outputting the SDHI clock. When the SDCLKCR.CLKEN bit is 0, output from the SDHI_CLK pin becomes low regardless of the CLKCTRLEN bit setting.

44.2.11 Transfer Data Size Register (SDSIZE)

Address(es): SDHI.SDSIZE 0008 AC4Ch



Bit	Symbol	Bit Name	Description	R/W
b9 to b0	LEN[9:0]	Transfer Data Size Setting	Set the transfer data size. *1	R/W
b11 to b10	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R
b31 to b12	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Do not rewrite these bits when the SDSTS2.CBSY flag is 1.

The SDSIZE register is used to set the transfer data size.

LEN[9:0] Bits (Transfer Data Size Setting)

When using single block transfer, the transfer data size can be set from 1 byte to 512 bytes. When CMD12 is automatically issued during a multi-block transfer sequence (CMD18 and CMD25), the transfer data size can only be set to 512 bytes. When CMD12 is not automatically issued during a multi-block transfer sequence, the transfer data size can be set to 32, 64, 128, 256, or 512 bytes. However, a 32-, 64-, 128-, or 256-byte multi-block read transfer can only be performed during an SDIO multi-block transfer (CMD53). Do not set these bits to 0 when using a command that includes data transfer.

44.2.12 Card Access Option Register (SDOPT)

Address SDHI.SDOPT 0008 AC50h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Bit Name	Description		R/W
b3 to b0	CTOP[3:0]	Card Detection Time Counter *1	b3 b0 0 0 0 0: PCLKB × 2^{10} 0 0 0 1: PCLKB × 2^{11} 0 0 1 0: PCLKB × 2^{12} 0 0 1 1: PCLKB × 2^{13} 0 1 0 0: PCLKB × 2^{14} 0 1 0 1: PCLKB × 2^{15} 0 1 1 0: PCLKB × 2^{16} 0 1 1 1: PCLKB × 2^{17}	b3 b0 1 0 0 0: PCLKB × 2^{18} 1 0 0 1: PCLKB × 2^{19} 1 0 1 0: PCLKB × 2^{20} 1 0 1 1: PCLKB × 2^{21} 1 1 0 0: PCLKB × 2^{22} 1 1 0 1: PCLKB × 2^{23} 1 1 1 0: PCLKB × 2^{24} 1 1 1 1: Do not set this value.	R/W
b7 to b4	TOP[3:0]	Timeout Counter *1	b7 b4 0 0 0 0: SDHI clock × 2^{13} 0 0 0 1: SDHI clock × 2^{14} 0 0 1 0: SDHI clock × 2^{15} 0 0 1 1: SDHI clock × 2^{16} 0 1 0 0: SDHI clock × 2^{17} 0 1 0 1: SDHI clock × 2^{18} 0 1 1 0: SDHI clock × 2^{19} 0 1 1 1: SDHI clock × 2^{20}	b7 b4 1 0 0 0: SDHI clock × 2^{21} 1 0 0 1: SDHI clock × 2^{22} 1 0 1 0: SDHI clock × 2^{23} 1 0 1 1: SDHI clock × 2^{24} 1 1 0 0: SDHI clock × 2^{25} 1 1 0 1: SDHI clock × 2^{26} 1 1 1 0: SDHI clock × 2^{27} 1 1 1 1: Do not set this value.	R/W
b8	—	Reserved	This bit is 0 when read. Set it to 0 when writing.		R/W
b12 to b9	—	Reserved	These bits are 0 when read and cannot be modified.		R
b13	—	Reserved	This bit is 0 when read. Set it to 0 when writing.		R/W
b14	—	Reserved	This bit is 1 when read and cannot be modified.		R
b15	WIDTH	SD Bus Width Select *1	0: Wide bus mode (4 bits) 1: Default bus mode (1 bit)		R/W
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.		R

Note 1. Do not rewrite these bits when the SDSTS2.CBSY flag is 1.

The SD bus width and timeout counter are set in the SDOPT register.

44.2.13 SD Error Status Register 1 (SDERSTS1)

Address(es): SDHI.SDERSTS1 0008 AC58h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	—	CRCTK[2:0]			CRCTKE	RDCRCE	RSPCRCE1	RSPCRCE0
Value after reset:	0	0	1	0	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CRCLENE	RDLENE	RSPLENE1	RSPLENE0	CMDE1	CMDE0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMDE0	Command Error Flag 0	0: Command index field value for a command *1 response is error free 1: Command index field value for a command *1 response is in error	R
b1	CMDE1	Command Error Flag 1	0: Command index field value for a command *2 response is error free 1: Command index field value for a command *2 response is in error (by setting the SDCMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the CMDE0 flag)	R
b2	RSPLENE0	Response Length Error Flag 0	0: Command *1 response length is error free 1: Command *1 response length is in error	R
b3	RSPLENE1	Response Length Error Flag 1	0: Command *2 response length is error free 1: Command *2 response length is in error (by setting the SDCMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the RSPLENE0 flag)	R
b4	RDLENE	Read Data Length Error Flag	0: Read data length error did not occur 1: Read data length error occurred	R
b5	CRCLENE	CRC Status Token Length Error Flag	0: CRC status token length error did not occur 1: CRC status token length error occurred	R
b7, b6	—	Reserved	These bits are 0 when read.	R
b8	RSPCRCE0	Response CRC Error Flag 0	0: No CRC error detected in command *1 response 1: CRC error detected in command *1 response	R
b9	RSPCRCE1	Response CRC Error Flag 1	0: No CRC error detected in command *2 response 1: CRC error detected in command *2 response (the error that occurs for CMD12 with the setting of the SDCMD.CMDIDX[5:0] bits is indicated by the RSPCRCE0 flag)	R
b10	RDCRCE	Read Data CRC Error Flag	0: No CRC error detected in read data 1: CRC error detected in read data	R
b11	CRCTKE	CRC Status Token Error Flag	0: No error detected in CRC status token 1: Error detected in CRC status token	R
b14 to b12	CRCTK[2:0]	CRC Status Token	Store the CRC status token value (normal value is 010b)	R
b15	—	Reserved	This bit is 0 when read.	R
b31 to b16	—	Reserved	These bits are undefined when read.	R

Note 1. Command other than CMD12 or CMD52 which are automatically issued to stop data transfer.
Note 2. CMD12 or CMD52 which are automatically issued to stop data transfer.

The SDERSTS1 register indicates the CRC status token, CRC error, end bit error, and command error.

44.2.14 SD Error Status Register 2 (SDERSTS2)

Address(es): SDHI.SDERSTS2 0008 AC5Ch

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b7	b6	b5	b4	b3	b2	b1	b0
	—	CRCBSYTO	CRCTO	RDTO	BSYTO1	BSYTO0	RSPTO1	RSPTO0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RSPTO0	Response Timeout Flag 0	0: After a command *1 was issued, a response was received in less than 640 cycles of the SDHI clock. 1: After a command *1 was issued, a response was not received even after 640 cycles or more of the SDHI clock elapsed.	R
b1	RSPTO1	Response Timeout Flag 1	0: After a command *2 was issued, a response was received in less than 640 cycles of the SDHI clock. 1: After a command *2 was issued, a response was not received even after 640 cycles or more of the SDHI clock elapsed (by setting the SDCMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the RSPTO0 flag).	R
b2	BSYTO0	Busy Timeout Flag 0	0: After the R1b response was received, the SDHI was released from the busy state during the specified period *3. 1: After the R1b response was received, the SDHI was in the busy state even after the specified period *3 elapsed.	R
b3	BSYTO1	Busy Timeout Flag 1	0: After CMD12 was automatically issued, the SDHI was released from the busy state during the specified period *3. 1: After CMD12 was automatically issued, the SDHI was in the busy state even after the specified period *3 elapsed (by setting the SDCMD.CMDIDX[5:0] bits, the error that occurs by issuing CMD12 is indicated by the BSYTO0 flag).	R
b4	RDTO	Read Data Timeout Flag	After a read command is issued, this flag becomes 1 when read data is not received even after the specified period *3 elapses. After read data is received, this flag becomes 1 when the next block of read data is not received even after the specified period *3 elapses. After the SDHI exits the read wait state, this flag becomes 1 when the next block of read data is not received even after the specified period *3 elapses.	R
b5	CRCTO	CRC Status Token Timeout Flag	0: After data was written to the SD card, a CRC status token was received during the specified period *3. 1: After CRC data was written to the SD card, a CRC status token was not received even after the specified period *3 elapsed.	R
b6	CRCBSYTO	CRC Status Token Busy Timeout Flag	0: After a CRC status token was received, the SDHI was released from the busy state during the specified period *3. 1: After a CRC status token was received, the SDHI is in the busy state even after the specified period *3 elapsed.	R
b31 to b7	—	Reserved	These bits are 0 when read.	R

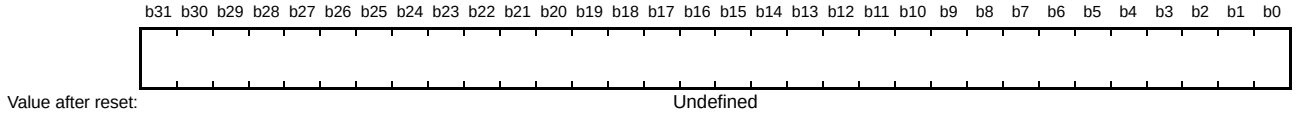
Note 1. Command other than CMD12 or CMD52 which are automatically issued to stop data transfer.
Note 2. CMD12 or CMD52 which are automatically issued to stop transfer.

Note 3. Set the SDOPT.TOP[3:0] bits to select the number of *n* cycles.

The SDERSTS2 register indicates the timeout status.

44.2.15 SD Buffer Register (SDBUFR)

Address(es): SDHI.SDBUFR 0008 AC60h



The SDBUFR register is used when writing data to the SD card and when reading data from the SD card. The SDBUFR register is connected to the SDHI's internal SD buffer. Refer to section 44.3.1, Data Block Format of the SD Card for details on the configuration of the SDBUFR register and the SD buffer.

44.2.16 SDIO Mode Control Register (SDIOMD)

Address(es): SDHI.SDIOMD 0008 AC68h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	C52PUB	IOABT	—	—	—	—	—	RWREQ	—	INTEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	INTEN	SDIO Interrupt Acceptance Enable *1	0: SDIO interrupt accept disabled 1: SDIO interrupt accept enabled	R/W
b1	—	Reserved	This bit is 0 when read and cannot be modified.	R
b2	RWREQ	Read Wait Request	0: SDHI exits read wait state 1: Request for SDHI to enter read wait state	R/W
b7 to b3	—	Reserved	These bits are 0 when read and cannot be modified.	R
b8	IOABT	SDIO Abort	If this bit is set to 1 during multi-block transfer triggered by CMD53, CMD52 is immediately issued, and the command sequence is aborted.	R/W
b9	C52PUB	SDIO None Abort	If this bit is set to 1 during multi-block transfer triggered by CMD53, CMD52 is issued before the transfer process is complete, and the command sequence is completed.	R/W
b31 to b10	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Do not rewrite this bit when the SDSTS2.CBSY flag is 1.

The SDIOMD register controls reception of the SDIO interrupt, controls CMD52 issuance during multi-block transfer, and controls the read wait request. Do not set bits C52PUB and IOABT to 1 at the same time.

RWREQ Bit (Read Wait Request)

If the RWREQ bit is set to 1 during a multi-block read sequence triggered by issuing CMD53, when the current block is done being read, the SDHI enters the read wait state. The method for exiting the read wait state is as follows.

- If the RWREQ bit is set to 0 while the SDHI is in the read wait state, the SDHI exits the read wait state.
- If the IOABT bit is set to 1 while the SDHI is in the read wait state, after CMD52 is issued, the RWREQ bit becomes 0 and the SDHI exits the read wait state.
- If bits C52PUB and RWREQ are simultaneously set to 1 during a multi-block read sequence triggered by issuing CMD53 *1, the SDHI does not automatically exit the read wait state, so after receiving the CMD52 response, set the RWREQ bit to 0.

Note 1. Set bits RWREQ and C52PUB to 1 simultaneously.

If the RWREQ bit is set to 1 while the last block is being transferred during a multi-block read sequence triggered by issuing CMD53, the SDHI will not enter the read wait state, the SDSTS1.ACEND flag becomes 1, and the RWREQ bit becomes 0. Set the RWREQ bit to 1 after the SDSTS1.RSPEND flag becomes 1.

IOABT Bit (SDIO Abort)

- If the IOABT bit is set to 1 during a multi-block transfer sequence triggered by issuing CMD53, the SDHI stops the CMD53 command sequence, and CMD52 is issued. If the command sequence is stopped due to a communication error or timeout, the SDHI does not issue CMD52. The SD buffer can be accessed even after the IOABT bit is set to 1, but the SDSTS2.ILR flag or ILW flag becomes 1, and a buffer access error occurs. Write a value to the SDARG register before setting the IOABT bit to 1.
- During a single block write, if there is no data in the SD buffer when the IOABT bit is set to 1, the SDHI does not issue CMD52, and the SDSTS1.ACEND flag becomes 1. If there is data in the SD buffer when the IOABT bit is set to 1, the SDHI does not issue CMD52, and after the SDHI exits the busy state, the SDSTS1.ACEND flag becomes 1.
- If the IOABT bit is set to 1 during a single block read, the SDHI does not issue CMD52, and the SDSTS1.ACEND flag immediately becomes 1.
- If the SDHI is in the busy state after the R1b response is received and the IOABT bit is set to 1, the SDHI does not issue CMD52, and after the SDHI exits the busy state, the SDSTS1.ACEND flag becomes 1.
- If the IOABT bit is set to 1 after the command sequence is completed, the SDHI does not issue CMD52, and the SDSTS1.ACEND flag does not become 1.
- Set the IOABT bit to 1 after the SDSTS1.RSPEND flag becomes 1.
- Set the IOABT bit to 0 after the SDSTS1.ACEND flag becomes 1.

C52PUB Bit (SDIO None Abort)

- If the C52PUB bit is set to 1 during a multi-block write sequence triggered by issuing CMD53, CMD52 is automatically issued when the SD buffer is empty and the current block write access is complete. The C52PUB bit becomes 0 after the response for CMD52 is received. If the C52PUB bit is 1 while the last block is being transferred, the SDHI does not issue CMD52, and after the SDSTS1.RSPEND flag becomes 1, the C52PUB bit is set to 0.
- If the C52PUB bit and RWREQ bit are set to 1 during a multi-block read sequence triggered by issuing CMD53, the SDHI enters the read wait state after the current block read access is complete, and the SDHI automatically issues CMD52. The C52PUB bit becomes 0 after the response for CMD52 is received. If the C52PUB bit is set to 1 while the last block is being transferred, the SDHI does not issue CMD52, and after the SDSTS1.RSPEND flag becomes 1, the C52PUB bit is set to 0.
- During a multi-block read sequence triggered by issuing CMD53, if the C52PUB bit is set to 1, also set the RWREQ bit to 1.
- Write a value to the SDARG register before setting the C52PUB bit to 1.
- Set the C52PUB bit to 1 after the SDSTS1.RSPEND flag becomes 1.

44.2.17 SDIO Status Register (SDIOSTS)

Address(es): SDHI.SDIOSTS 0008 AC6Ch

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	EXWT	EXPUB52	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

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

Bit	Symbol	Bit Name	Description	R/W
b0	IOIRQ	SDIO Interrupt Status Flag	0: SDIO interrupt not accepted 1: SDIO interrupt accepted	R/(W) *1
b2, b1	—	Reserved	These bits are undefined when read. Set them to 1 when writing.	R/W
b13 to b3	—	Reserved	These bits are 0 when read and cannot be modified.	R
b14	EXPUB52	EXPUB52 Status Flag	Indicates the status of the EXPUB52	R/(W) *1
b15	EXWT	EXWT Status Flag	Indicates the status of the EXWT	R/(W) *1
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. The flag value does not change even when set to 1. If 0 is written to this flag, it becomes 0.

The SDIOSTS register indicates the status of the SDIO card access. When clearing a flag, bits to be cleared should be set to 0; bits not being cleared should be set to 1.

IOIRQ Flag (SDIO Interrupt Status Flag)

— This flag becomes 1 under the following condition:

- The SDIO interrupt from the SDIO card is accepted while the SDIOMD.INTEN bit is 1.

— This flag becomes 0 under the following condition:

- The flag is set to 0. *1

Note 1. Access the SDIO card, negate the SDIO interrupt from the SDIO card, and then set the IOIRQ flag to 0. If the SDIO interrupt from the SDIO card is not negated, the IOIRQ flag might become 1 again.

EXPUB52 Flag (EXPUB52 Status Flag)

— This flag becomes 1 under any of the following conditions:

- When multi-block transfer is triggered by CMD53 being issued, the SDIOMD.C52PUB bit is set to 1 while the last block is being transferred.
- When multi-block write is triggered by CMD53 being issued, the SDIOMD.C52PUB bit remains set to 1 while the last block is being transferred.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

EXWT Flag (EXWT Status Flag)

— This flag becomes 1 under the following condition:

- During a multi-block read sequence triggered by CMD53 being issued, the SDIOMD.RWREQ bit is set to 1 while the last block is being transferred.

— This flag becomes 0 under the following condition:

- The flag is set to 0.

44.2.18 SDIO Interrupt Mask Register (SDIOIMSK)

Address(es): SDHI.SDIOIMSK 0008 AC70h

	b31	b30	b29	b28	b27	b26	b25	b24
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8
	EXWTM	EXPUB52M	—	—	—	—	—	—
Value after reset:	1	1	0	0	0	0	0	0

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

Bit	Symbol	Bit Name	Description	R/W
b0	IOIRQM	IOIRQ Interrupt Mask Control	0: IOIRQ interrupt not masked 1: IOIRQ interrupt masked	R/W
b2, b1	—	Reserved	These bits are 1 when read. Set them to 1 when writing.	R/W
b13 to b3	—	Reserved	These bits are 0 when read and cannot be modified.	R
b14	EXPUB52M	EXPUB52 Interrupt Request Mask Control	0: EXPUB52 interrupt request not masked 1: EXPUB52 interrupt request masked	R/W
b15	EXWTM	EXWT Interrupt Request Mask Control	0: EXWT interrupt request not masked 1: EXWT interrupt request masked	R/W
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

The SDIOIMSK register enables and disables the interrupt requests from the status flags in the SDIOSTS register. Refer to Table 44.8, Interrupt Sources for details on the relationship between the status flags and the requested interrupt source.

44.2.19 DMA Transfer Enable Register (SDDMAEN)

Address(es): SDHI.SDDMAEN 0008 ADB0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b1	DMAEN	DMA Transfer Enable *1 *2	0: Using DMAC and DTC to access the SDBUFR register is disabled 1: Using DMAC and DTC to access the SDBUFR register is enabled	R/W
b3, b2	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R
b4	—	Reserved	This bit is 1 when read. Set it to 1 when writing.	R
b5	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b7, b6	—	Reserved	These bits are 0 when read and cannot be modified.	R
b9, b8	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b11, b10	—	Reserved	These bits are 0 when read and cannot be modified.	R
b12	—	Reserved	This bit is 1 when read. Set it to 1 when writing.	R
b31 to b13	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Do not rewrite this bit when the SDSTS2.CBSY bit is 1.
Note 2. When the SDIMSK2.BWEM bit is 0 or the SDIMSK2.BREM bit is 0, set the SDDMAEN.DMAEN bit to 0. When the SDDMAEN.DMAEN bit is 1, set the SDIMSK2.BWEM bit to 1 and the SDIMSK2.BREM bit to 1.

The SDDMAEN register enables and disables DMA transfer.

DMAEN Bit (DMA Transfer Enable)

When using DMA transfer to access the SD buffer, set the DMAEN bit to 1 before setting the SDCMD register.

44.2.20 SDHI Software Reset Register (SDRST)

Address(es): SDHI.SDRST 0008 ADC0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Bit Name	Description	R/W
b0	SDRST	SDHI Software Reset Control	0: SDHI software reset 1: SDHI software reset released	R/W
b2, b1	—	Reserved	These bits are 1 when read. Set them to 1 when writing.	R
b31 to b3	—	Reserved	These bits are 0 when read and cannot be modified.	R

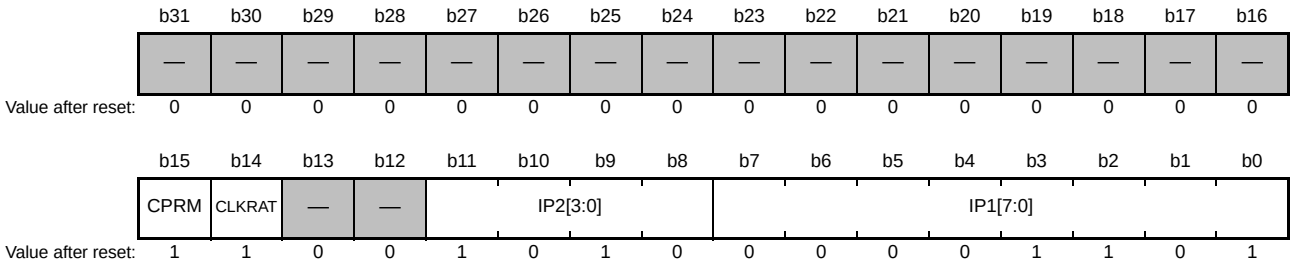
Table 44.5 lists the bits and flags initialized by the SDHI software reset.

Table 44.5 Bits and Flags Initialized by the SDHI Software Reset

Register	Bit/Flag
SDSTOP	SDBLKCNTEN
SDSTS1	RSPEND, ACEND
SDSTS2	CMDE, CRCE, ENDE, DTO, ILW, ILR, RSPTO, SDD0MON, BRE, BWE, SDCLKCREN, ILA
SDCLKCR	CLKEN
SDOPT	CTOP[3:0], TOP[3:0], WIDTH Bits b8 and b13 in the SDOPT register are also initialized by the SDHI software reset.
SDERSTS1	CMDE0, CMDE1, RSPLENE0, RSPLENE1, RDLENE, CRCLENE, RSPCRCE0, RSPCRCE1, RDCRCE, CRCTKE, CRCTK[2:0]
SDERSTS2	RSPTO0, RSPTO1, BSYTO0, BSYTO1, RDTO, CRCTO, CRCBSYTO
SDIOSTS	IOIRQ, EXPUB52, EXWT

44.2.21 Version Register (SDVER)

Address(es): SDHI.SDVER 0008 ADC4h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	IP1[7:0]	IP Version 1	IP version 1	R
b11 to b8	IP2[3:0]	IP Version 2	IP version 2	R
b13, b12	—	Reserved	These bits are 0 when read.	R
b14	CLKRAT	Operating Clock Condition	0: SDHI clock frequency = PCLKB frequency not supported 1: SDHI clock frequency = PCLKB frequency supported	R
b15	CPRM	CPRM Function Select	0: CPRM function is enabled 1: CPRM function is disabled	R
b31 to b16	—	Reserved	These bits are 0 when read.	R

The SDVER register indicates the SDHI version.

44.2.22 Swap Control Register (SDSWAP)

Address(es): SDHI.SDSWAP 0008 ADE0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	BRSWP	BWSWP	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is 0 when read and cannot be modified.	R
b1	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b2	—	Reserved	This bit is 0 when read and cannot be modified.	R
b4, b3	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b5	—	Reserved	This bit is 0 when read and cannot be modified.	R
b6	BWSWP	SDBUFR Swap Write *1	0: Normal write operation 1: Swap the byte endian before writing to the SDBUFR register	R/W
b7	BRSWP	SDBUFR Swap Read *1	0: Normal read operation 1: Swap the byte endian before reading the SDBUFR register	R/W
b10 to b8	—	Reserved	These bits are 0 when read and cannot be modified.	R
b12, b11	—	Reserved	These bits are 0 when read. Set them to 0 when writing.	R/W
b14, b13	—	Reserved	These bits are 0 when read and cannot be modified.	R
b15	—	Reserved	This bit is 0 when read. Set it to 0 when writing.	R/W
b31 to b16	—	Reserved	These bits are 0 when read and cannot be modified.	R

Note 1. Do not rewrite this bit when the SDSTS2.CBSY flag is 1.

The SDSWAP register is used to select whether or not the byte endian is swapped when accessing the SDBUFR register. Refer to section 44.3.1 for details on the differences in accessing the SDBUFR register based on the SDSWAP register value.

44.3 SDHI Operation

44.3.1 Data Block Format of the SD Card

The SDHI has a default bus mode (1-bit width) that uses just the SDHI_D0 pin as a data line and a wide bus mode (4-bit width) that uses pins SDHI_D0 to SDHI_D3. Figure 44.2 shows the transfer format when the SDOPT.WIDTH bit is 1 (default bus mode), and Figure 44.3 shows the transfer format when the SDOPT.WIDTH bit is 0 (wide bus mode).

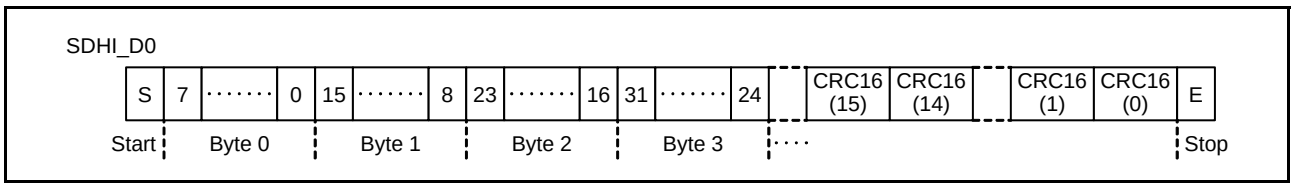


Figure 44.2 Transfer Format in Default Bus Mode

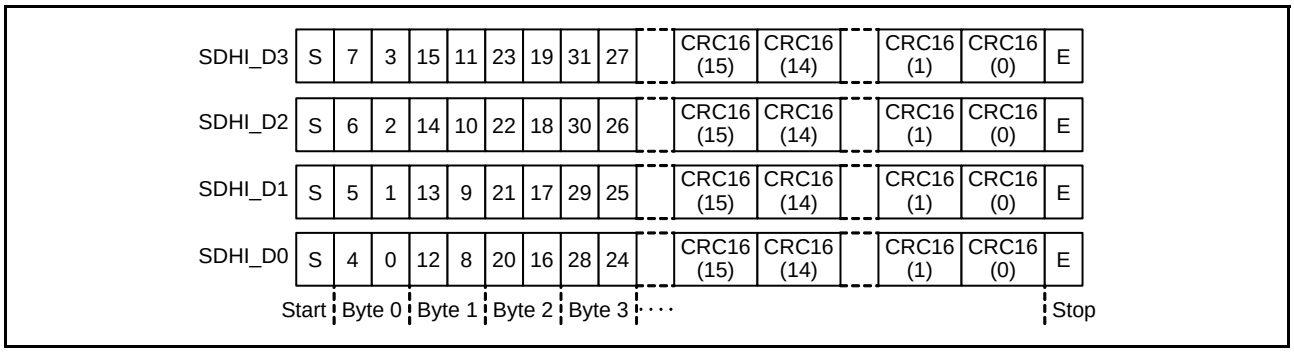


Figure 44.3 Transfer Format in Wide Bus Mode

44.3.2 SD Buffer and the SDBUFR Register

The SDHI transfers data to an SD card via its internal SD buffer. The SD buffer is comprised of a double buffer, and each buffer is 512 bytes. Figure 44.4 shows the data configuration of a single buffer of the SD buffer's double buffer.

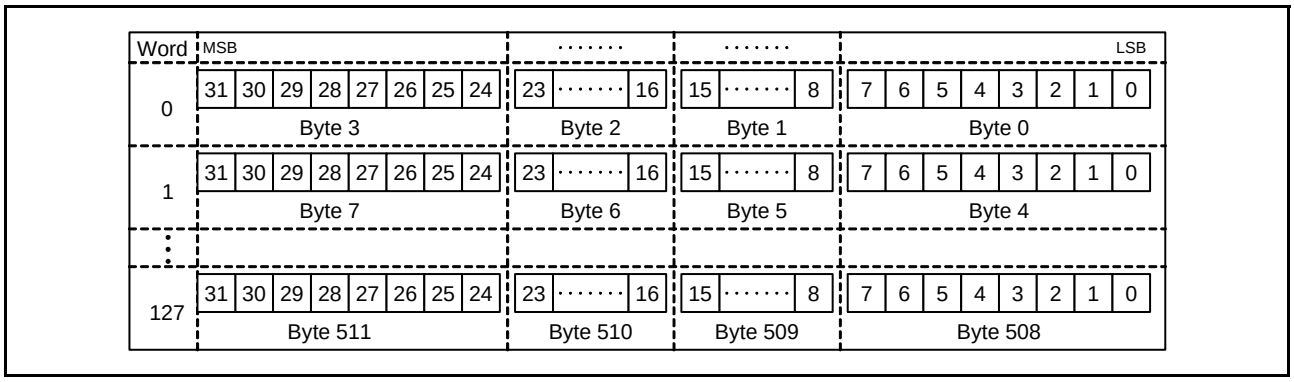


Figure 44.4 Data Configuration of Single Buffer in the SD Buffer

Access to the SD buffer is done via the SDBUFR register. If data is written to the SDBUFR register while the SDSWAP.BWSWP bit is 1, the SDHI swaps the endian for the byte, and stores the data in the SDBUFR register. If data is read from the SDBUFR register while the SDSWAP.BRSWP bit is 1, the data of the endian for the byte that was swapped can be read. Figure 44.5 shows the data alignment when reading the SDBUFR register.

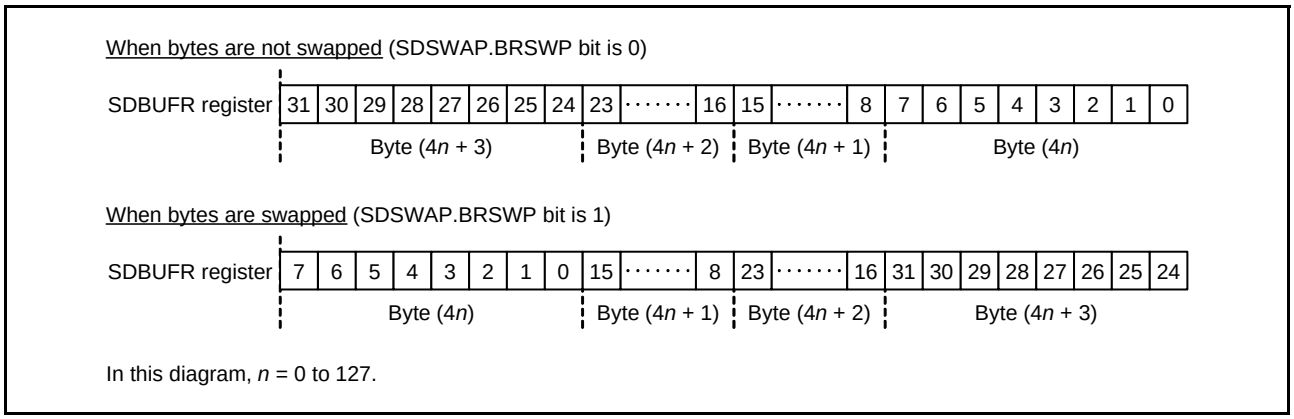


Figure 44.5 Data Alignment When Reading the SDBUFR Register

44.3.3 SD Card Detection

The SDHI can detect an SD card using either the SDHI_CD pin or SDHI_D3 pin.

44.3.3.1 Using the SDHI_CD Pin to Detect an SD Card

Figure 44.6 shows the timing chart for SD card detection using the SDHI_CD pin. The SDHI_CD pin is connected to the card detection switch of the SD card connector, and is pulled-up by the MCU. The pull-up resistance value is determined by the specifications of the host device. Note that there are some SD card sockets whose card detection switches become open when the SD card is inserted.

- Detecting SD card insertion
  The signal from the SDHI_CD pin becomes low when an SD card is inserted. This causes the SDSTS1.SDCDIN flag to become 1 if the SDHI_CD pin is low for the number of cycles set in the SDOPT.CTOP[3:0] bits. The SDSTS1.SDCDIN flag is cleared by setting it to 0.
- Detecting SD card removal
  The signal from the SDHI_CD pin becomes high when the SD card is removed. This causes the SDSTS1.SDCDRM flag to become 1 if the SDHI_CD pin is high for the number of cycles set in the SDOPT.CTOP[3:0] bits. The SDSTS1.SDCDRM flag is cleared by setting it to 0.

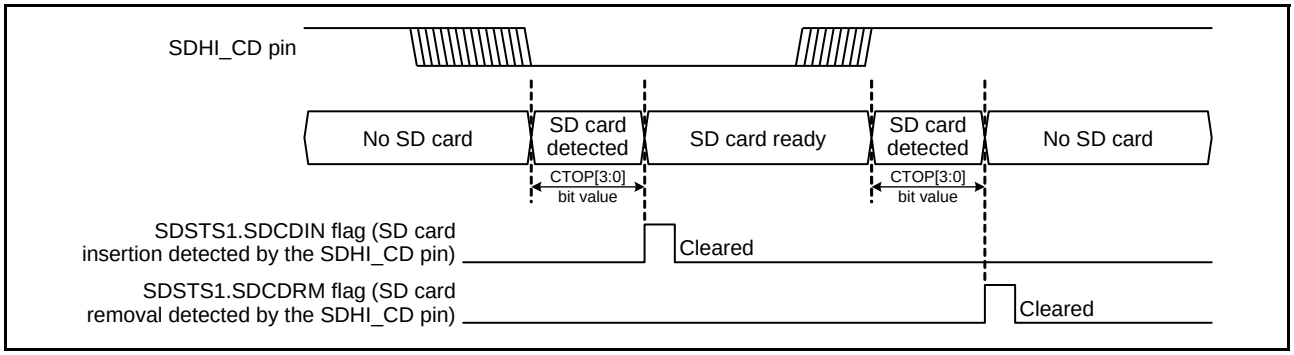


Figure 44.6 SD Card Detection Using the SDHI_CD Pin

44.3.3.2 Using the SDHI_D3 Pin to Detect an SD Card

Figure 44.7 shows the timing chart for SD card detection using the SDHI_D3 pin. The SDHI_D3 pin is pulled-down by the MCU. The pull-down resistance value is determined by the specifications of the host device.

- Detecting SD card insertion
  The signal from the SDHI_D3 pin becomes high when an SD card is inserted. This causes the SDSTS1.SDD3IN flag to become 1. The SDSTS1.SDD3IN flag is cleared by setting it to 0.
- Detecting SD card removal
  The signal from the SDHI_D3 pin becomes low when the SD card is removed. This causes the SDSTS1.SDD3RM flag to become 1. The SDSTS1.SDD3RM flag is cleared by setting it to 0.

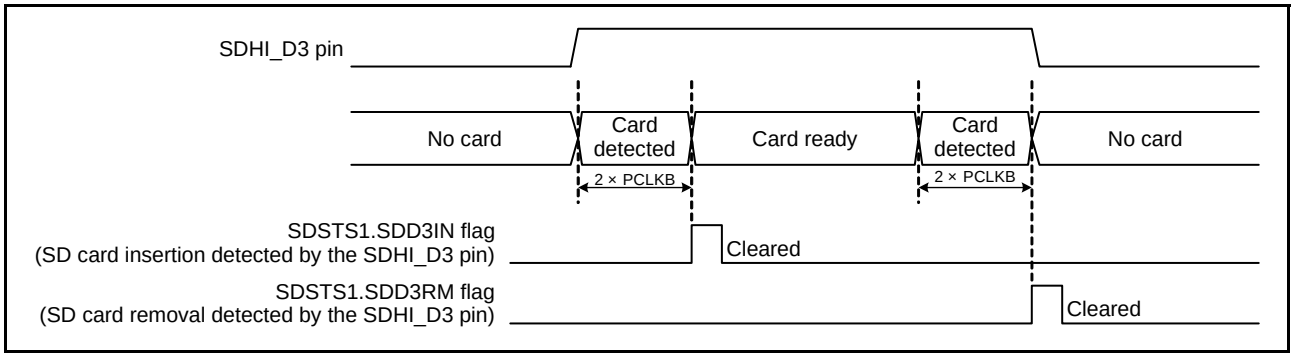


Figure 44.7 SD Card Detection Using the SDHI_D3 Pin

44.3.4 SD Card Write Protection

The SDHI can disable writing to an SD card via the SDHI_WP pin or a command.

44.3.4.1 Using the SDHI_WP Pin to Enable Write Protection

The SDHI_WP pin is connected to the WP detection switch of the SD card connector, and the SDHI_WP pin is pulled-down or pulled-up when an SD card is inserted. The resistance value and whether the SDHI_WP pin is pulled-up or pulled-down are determined by the specifications of the host device. The status of the SDHI_WP pin is reflected in the SDSTS1.SDWPMON flag. After an SD card is inserted, read the SDSTS1.SDWPMON flag to check if write protection is enabled or disabled.

44.3.4.2 Using a Command to Enable Write Protection

The SDHI uses the write protect command or the SD card lock command to disable writing to the SD card.

44.3.5 Communication Errors and Timeouts

When a communication error or timeout error occurs, depending on the type of error, the corresponding status flag in the SDSTS2 register becomes 1. Also, depending on the source of the error, the corresponding flag in the SDERSTS1 or SDERSTS2 register becomes 1.

The status flags in registers SDERSTS1 and SDERSTS2 become 0 by writing to the SDCMD register, or by setting the SDRST.SDRST bit to 0.

Table 44.6 Communication Errors

Communication Error	Interrupt Flag Register		Error Status Register		This Occurs When...
	Register symbol	Bit symbol	Register symbol	Bit symbol	
End bit error	SDSTS2	ENDE	SDERSTS1	CRCLENE	The CRC status token length is in error
				RDLENE	The read data length is in error
				RSPLENE1	The response length is in error *1
				RSPLENE0	The response length is in error *2
CRC error		CRCE		CRCTKE	The CRC status token is in error
				RDCRCE	There is a CRC error in the read data
				RSPCRCE1	There is a CRC error in the response *1
				RSPCRCE0	There is a CRC error in the response *2
Command error		CMDE		CMDE1	The command index field value for the transmitted command and received response do not match *1
				CMDE0	The command index field value for the transmitted command and received response do not match *2

Note 1. CMD12 or CMD52 which are automatically issued to stop transfer.
Note 2. A command other than CMD12 or CMD52 which are automatically issued to stop transfer.

Table 44.7 Timeouts

Timeout	Interrupt Flag Register		Error Status Register		This Occurs When...
	Register symbol	Bit symbol	Register symbol	Bit symbol	
Response timeout	SDSTS2	RSPTO	SDERSTS2	RSPTO1	A response is not received even after a minimum of 640 SDHI clock cycles elapse *1
				RSPTO0	A response is not received even after a minimum of 640 SDHI clock cycles elapse *2
Data timeout (excluding response timeout)		DTO		CRCBSYTO	After the CRC status token is received, the SDHI is busy for at least the period set *3
				CRCTO	After the write data is transmitted, the CRC status token is not received even after at least the period set *3 elapses
				RDTO	After the read command is issued, the read data is not received even after at least the period set *3 elapses
					After the read data is received, the next block read data is not received even after at least the period set *3 elapses
					After the SDHI exits the read wait state, the next block read data is not received even after at least the period set *3 elapses
				BSYTO1	After CMD12 is issued during the command sequence, the SDHI is busy for at least the period set *3
				BSYTO0	After the R1b response is received, the SDHI is busy for at least the period set *3 (a command other than CMD12 is issued during the command sequence)

Note 1. CMD12 or CMD52 which are automatically issued to stop transfer.
Note 2. A command other than CMD12 or CMD52 which are automatically issued to stop transfer.
Note 3. The period is set in the SDOPT.TOP[3:0] bits.

44.3.6 Examples of Issuing a Command

44.3.6.1 Command Absent of Response Reception and Data Transfer

Figure 44.8 shows an example of no response being received and no data being transferred after the SDHI issues a command.

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 44.5.5 for details on setting the SDCLKCR register.
3. After setting the argument field value to the SDARG register, write the command information to be sent to the SDCMD register. The SDHI issues a command when a value is written to the SDCMD register.
4. After a command is issued, the SDSTS1.RSPEND flag becomes 1, and a response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0.

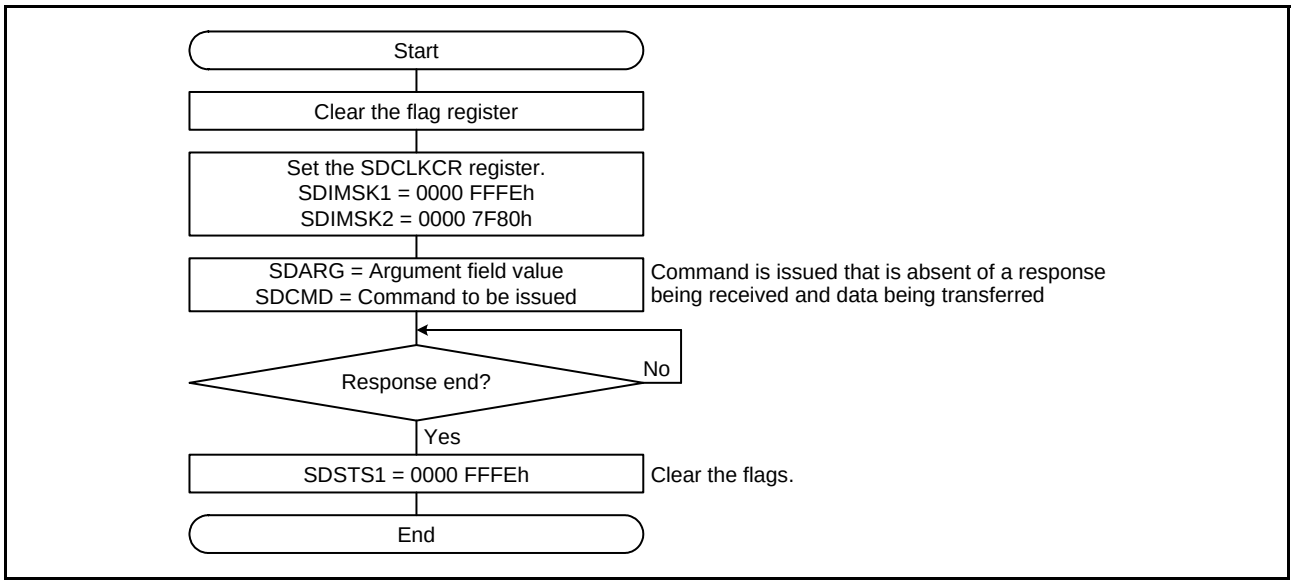


Figure 44.8 Command Issued That Is Absent of Response Reception and Data Transfer

44.3.6.2 Command Absent of Data Transfer

Figure 44.9 shows an example of no data being transferred after the SDHI issues a command.

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 44.5.5 for details on setting the SDCLKCR register.
3. After setting the argument field value in the SDARG register, write the information of the command to be issued to the SDCMD register. The SDHI issues a command when a value is written to the SDCMD register.
4. After a response is received, the SDSTS1.RSPEND flag becomes 1, and a response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0, and read the response stored in the SDRSP10 register.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

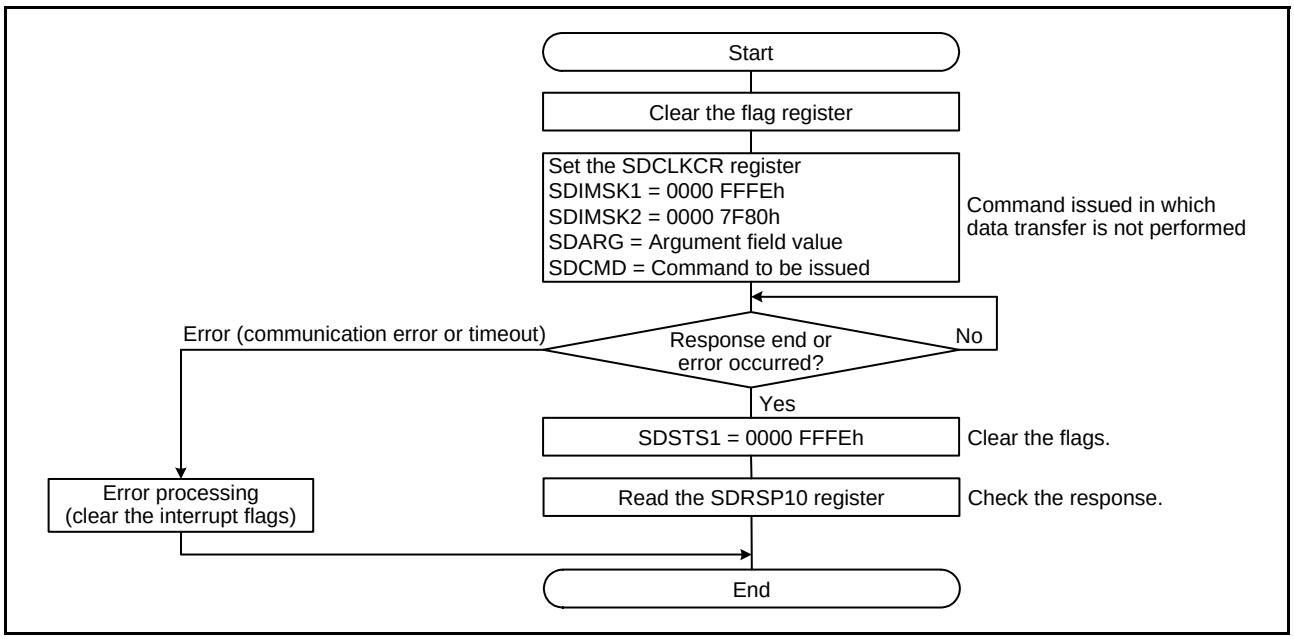


Figure 44.9 Command Issued That Is Absent of Data Transfer

44.3.6.3 Single Block Read Command (CMD17)

Figure 44.10 shows an example of issuing the single block read command (CMD17).

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 44.5.5 for details on setting the SDCLKCR register.
3. After setting the argument field value for CMD17 to the SDARG register, write 0000 0011h to the SDCMD register. The SDHI issues CMD17 when a value is written to the SDCMD register.
4. When the response is received, the SDSTS1.RSPEND flag becomes 1, and a response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0, and read the response stored in the SDRSP10 register. If the response read is in error, set the SDSTOP.STP bit or SDIOMD.IOABT bit to 1, and the command sequence can be stopped. When the command sequence is stopped, the SDSTS1.ACEND flag becomes 1. Note that CMD12 and CMD52 are not automatically issued by stopping this command sequence.
6. After the response is received, set the SDIMSK1.ACENDM bit to 0 and set the SDIMSK2.BREM bit to 0.
7. After the amount of data set in the SDSIZE.LEN[9:0] bits is received from the SD card, the SDSTS2.BRE flag becomes 1, and the BRE interrupt request is generated.
8. Set the SDSTS2.BRE flag to 0, and read the amount of data set in the SDSIZE.LEN[9:0] bits from the SDBUFR register.
9. After data has been read from the SDBUFR register, the SDSTS1.ACEND flag becomes 1, and the access end interrupt request is generated.
10. Set the SDSTS1.ACEND flag to 0.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

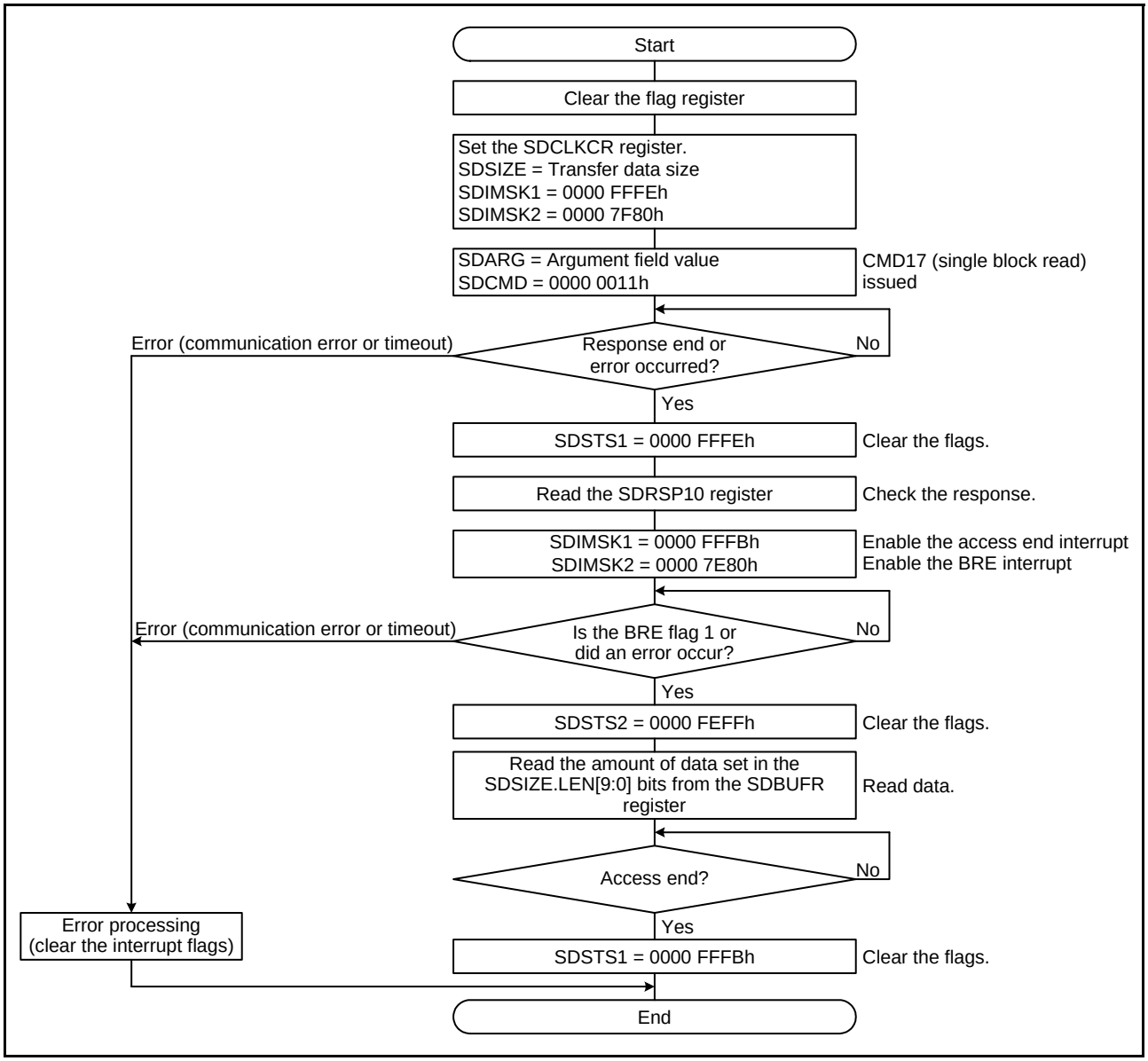


Figure 44.10 Issuing the Single Block Read Command

44.3.6.4 Single Block Write Command (CMD24)

Figure 44.11 shows an example of issuing the single block write command (CMD24).

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 44.5.5 for details on setting the SDCLKCR register.
3. After setting the argument field value for CMD24 to the SDARG register, write 0000 0018h to the SDCMD register. The SDHI issues CMD24 when a value is written to the SDCMD register.
4. When the response is received, the SDSTS1.RSPEND flag becomes 1, and the response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0 and read the response stored in the SDRSP10 register. If the read response is in error, set the SDSTOP.STP bit or SDIOMD.IOABT bit to 1, and the command sequence can be stopped. When the command sequence is stopped, the SDSTS1.ACEND flag becomes 1. Note that when this command sequence is stopped, CMD12 and CMD52 are not automatically issued.
6. After the response is received, set the SDIMSK1.ACENDM bit to 0, and set the SDIMSK2.BWEM bit to 0.
7. When the SDBUFR register becomes write accessible, the SDSTS2.BWE flag becomes 1, and the BWE interrupt request is generated.
8. Set the SDSTS2.BWE flag to 0, and write the amount of data set in the SDSIZE.LEN[9:0] bits to the SDBUFR register. After writing to the SDBUFR register, the SDHI transmits write data to the SD card. Also, after writing to the SDBUFR register, data transmission may cause a communication error or timeout to occur.
9. After all data has been written to the SD card, the SDHI receives the CRC status token, and the SDHI_D0 pin line becomes busy (low). Then, when the SDHI exits the busy state, the SDSTS1.ACEND flag becomes 1, and the access end interrupt request is generated.
10. Set the SDSTS1.ACEND flag to 0.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

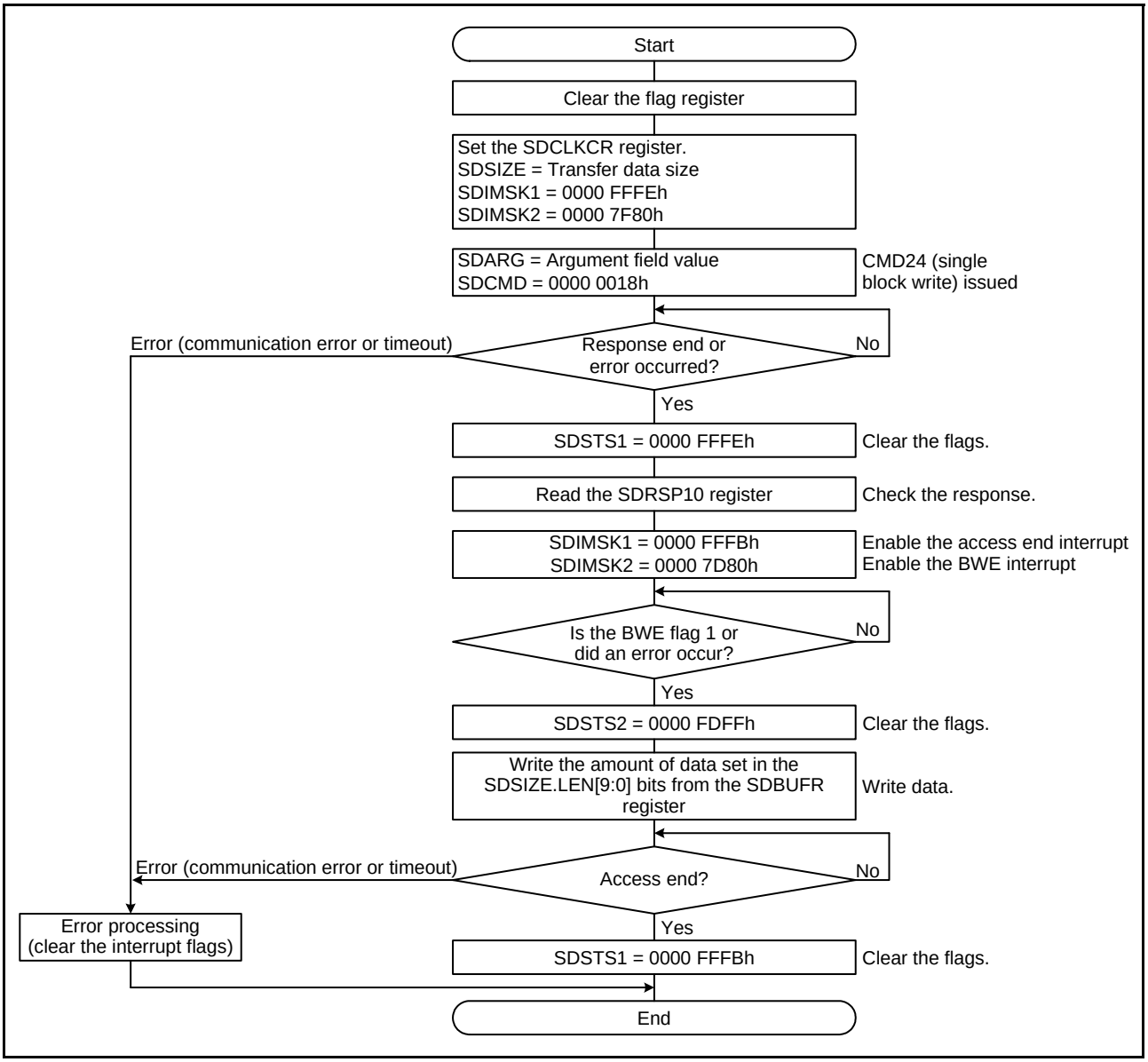


Figure 44.11 Issuing the Single Block Write Command

44.3.6.5 Multi-Block Read Command (CMD18)

Figure 44.12 shows an example of issuing the multi-block read command (CMD18).

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 44.5.5 for details on setting the SDCLKCR register. Set the SDSTOP.SDBLKCNTEN bit to 1, and set the number of transfer blocks in the SDBLKCNT register.
3. After setting the argument field value for CMD18 to the SDARG register, write 0000 0012h to the SDCMD register. The SDHI issues CMD18 when a value is written to the SDCMD register.
4. When the response is received, the SDSTS1.RSPEND flag becomes 1, and the response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0 and read the response stored in the SDRSP54 register. If the read response is in error, set the SDSTOP.STP bit to 1, and the command sequence can be stopped. When the SDSTOP.STP bit is set to 1, the SDHI automatically issues CMD12, and the response is received. At this point, the SDSTS1.ACEND flag becomes 1, and if the access end interrupt request is enabled, the access end interrupt request is generated. Next, set the SDSTS1.ACEND flag to 0 and read the response.
6. After the response is received, set the SDIMSK1.ACENDM bit to 0, and set the SDIMSK2.BREM bit to 0.
7. After receiving one block of data from the SD card, the SDSTS2.BRE bit becomes 1, and the BRE interrupt request is generated.
8. Set the SDSTS2.BRE flag to 0, and read the amount of data set in the SDSIZE.LEN[9:0] bits from the SDBUFR register. The read access to the SDBUFR register repeats for the amount of transfer blocks set in the SDBLKCNT register. Also, while reading the SDBUFR register, data reception may cause a communication error or timeout to occur. After the amount of transfer blocks set in the SDBLKCNT register have been read, the SDHI automatically issues CMD12, and the response is received. At this time, the SDHI automatically writes 0000 0000h to the SDARG register.
9. When all blocks have been received and the CMD12 response is received, the SDSTS1.ACEND flag becomes 1 and the access end interrupt request is generated.
10. Set the SDSTS1.ACEND flag to 0 and read the response.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

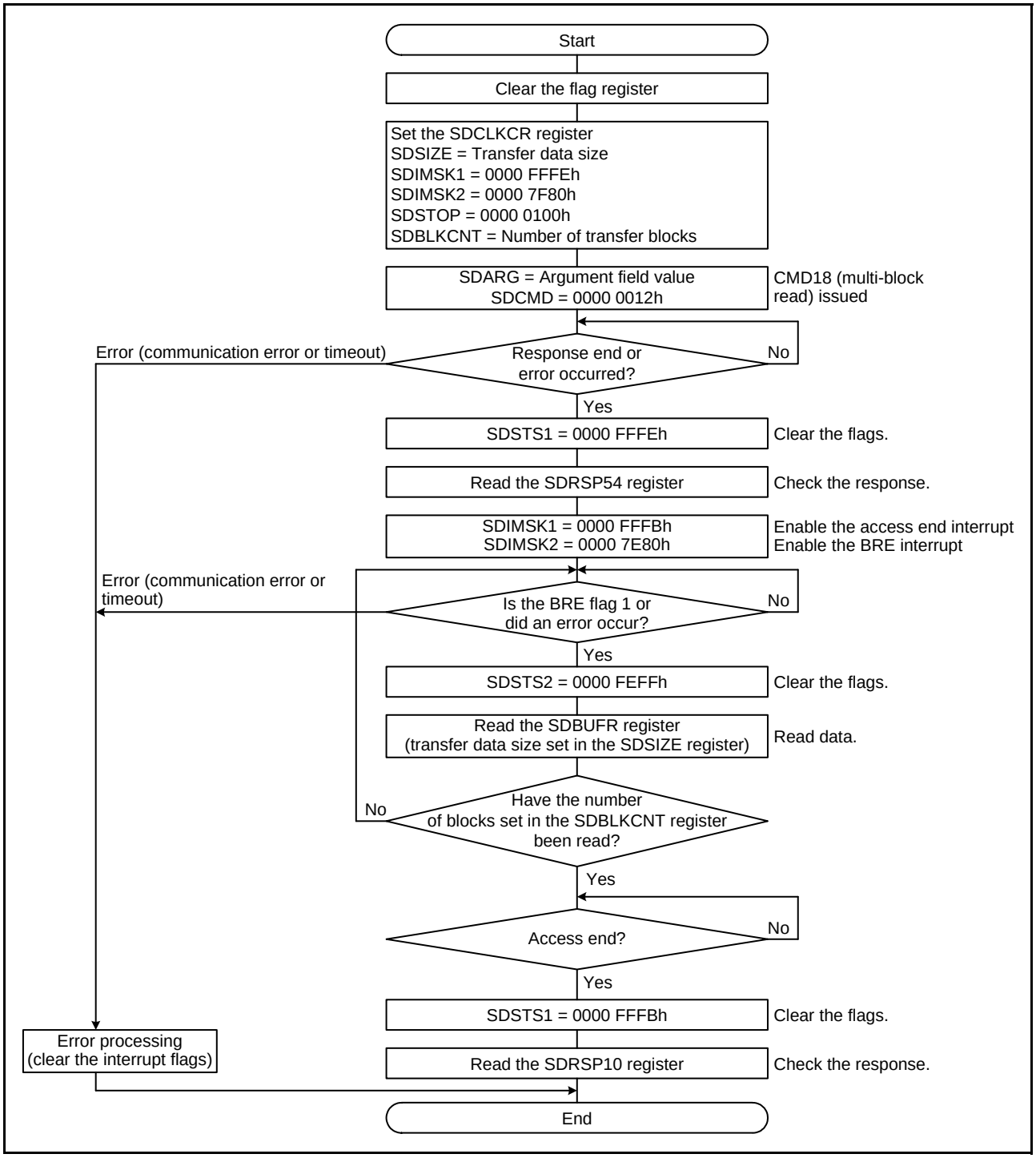


Figure 44.12 Issuing the Multi-Block Read Command

44.3.6.6 Multi-Block Write Command (CMD25)

Figure 44.13 shows an example of issuing the multi-block write command (CMD25).

1. Set the flags in registers SDSTS1 and SDSTS2 to 0.
2. Set the SDHI clock in the SDCLKCR register, and set the interrupt requests to be masked in registers SDIMSK1 and SDIMSK2. Refer to section 44.5.5 for details on setting the SDCLKCR register. Set the SDSTOP.SDBLKCNTEN bit to 1, and set the number of transfer blocks in the SDBLKCNT register.
3. After setting the argument field value for CMD25 to the SDARG register, write 0000 0019h to the SDCMD register. The SDHI issues CMD25 when a value is written to the SDCMD register.
4. When the response is received, the SDSTS1.RSPEND flag becomes 1, and the response end interrupt request is generated.
5. Set the SDSTS1.RSPEND flag to 0 and read the response stored in the SDRSP54 register. If the read response is in error, set the SDSTOP.STP bit to 1, and the command sequence can be stopped. When the SDSTOP.STP bit is set to 1, the SDHI automatically issues CMD12, and the response is received. At this point, the SDSTS1.ACEND flag becomes 1, and if the access end interrupt request is enabled, the access end interrupt request is generated. Next, set the SDSTS1.ACEND flag to 0 and read the response.
6. After the response is received, configure the SDIMSK1 register to enable the access end interrupt request, and configure the SDIMSK2 register to enable the BWE interrupt request.
7. When the SDBUFR register becomes write accessible, the SDSTS2.BWE flag becomes 1, and the BWE interrupt request is generated.
8. Set the SDSTS2.BWE flag to 0, and write the amount of data set in the SDSIZE.LEN[9:0] bits to the SDBUFR register. After writing to the SDBUFR register, and after the SDHI transmits write data to the SD card, the CRC status token is received, and the SDHI_D0 pin line becomes busy (low). The write access to the SDBUFR register and CRC status token reception repeat for the amount of transfer blocks set in the SDBLKCNT register. Also, after writing to the SDBUFR register, data transmission may cause a communication error or timeout to occur. After the amount of transfer blocks set in the SDBLKCNT register have been written, the SDHI automatically issues CMD12, and the response is received. At that time, the SDHI automatically writes 0000 0000h to the SDARG register.
9. When all blocks have been transmitted and the CRC status token is received, the SDHI exits the busy state, the SDSTS1.ACEND flag becomes 1, and the access end interrupt request is generated.
10. Set the SDSTS1.ACEND flag to 0 and read the response.

Perform error processing (clear the interrupt flag) if a communication error or timeout occurs.

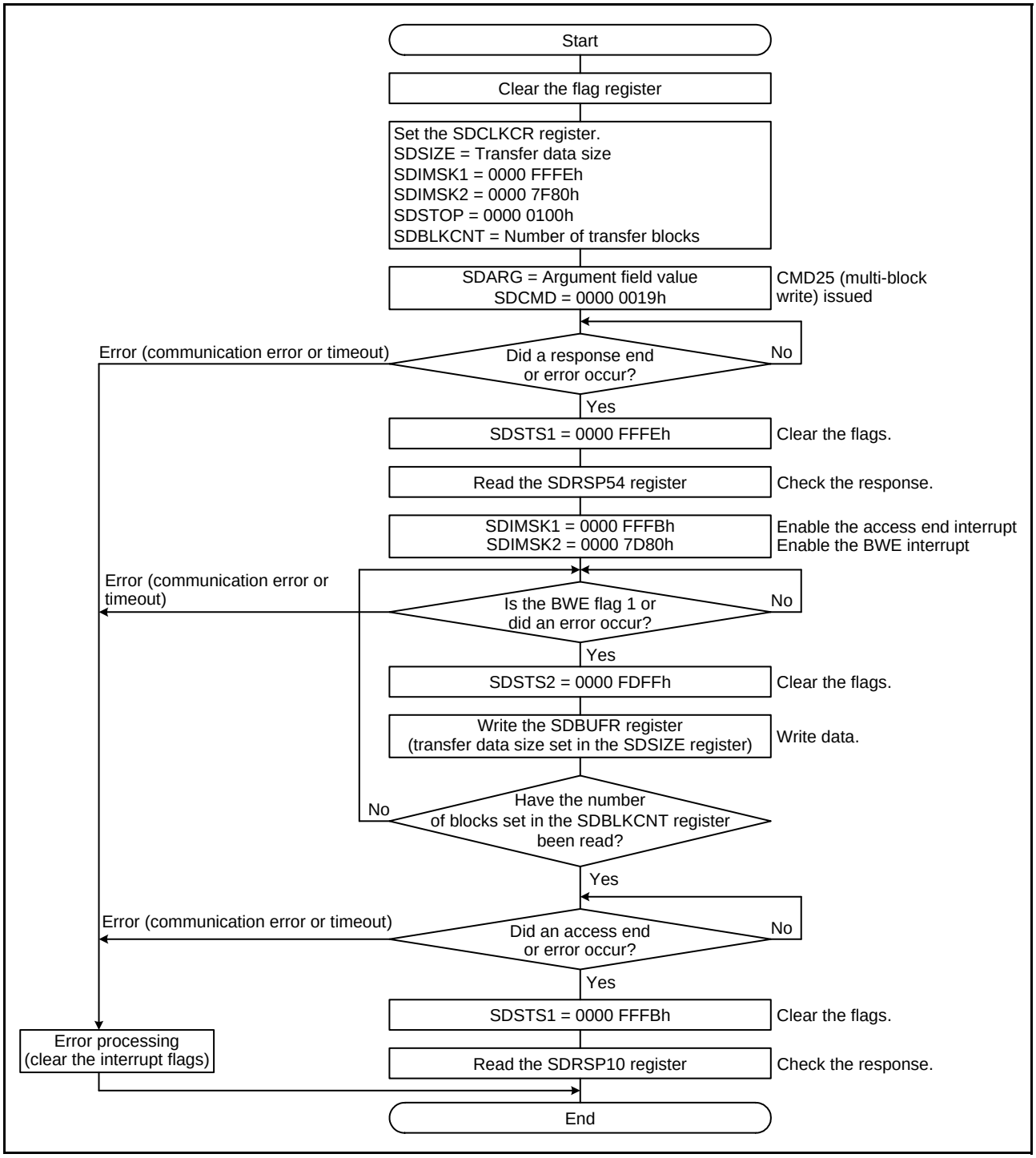


Figure 44.13 Issuing the Multi-Block Write Command

44.3.6.7 IO_RW_DIRECT Command (CMD52)

Figure 44.14 shows an example of issuing the IO_RW_DIRECT command (CMD52).

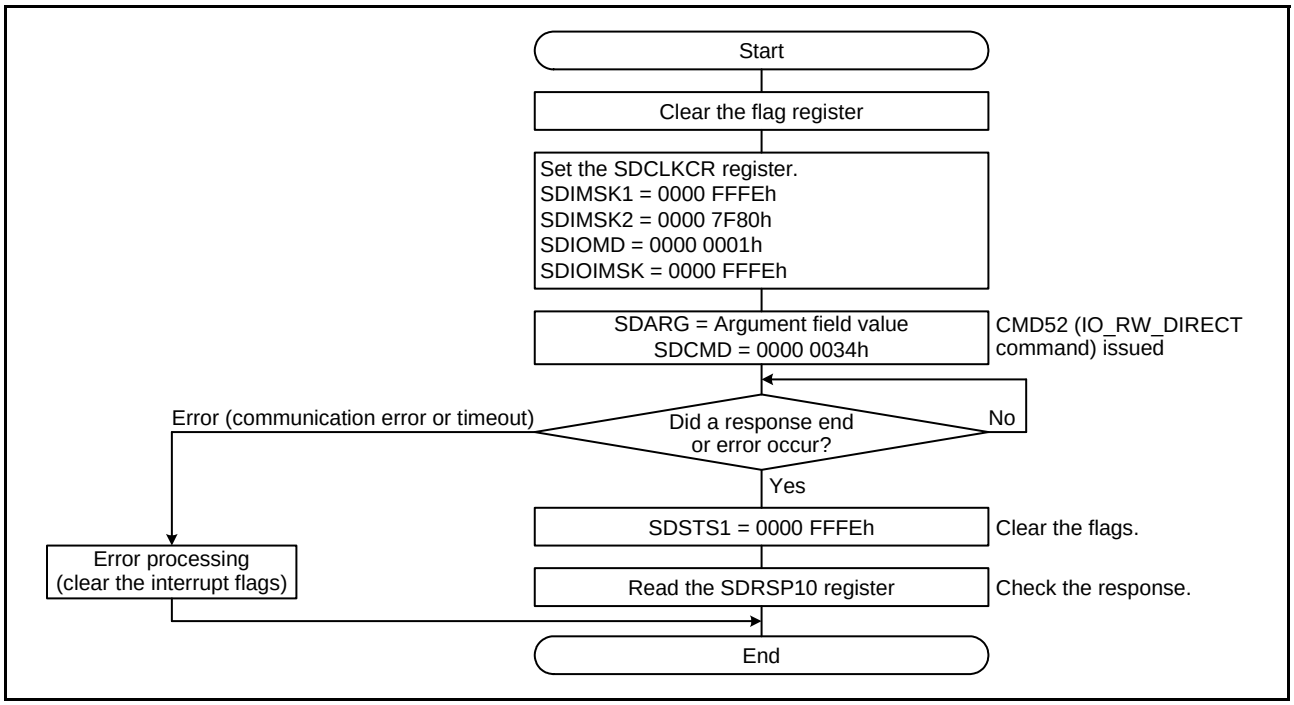


Figure 44.14 Issuing the IO_RW_DIRECT Command

44.3.6.8 IO_RW_EXTENDED Command (CMD53 (Multi-Block Read))

Figure 44.15 shows an example of issuing the IO_RW_EXTENDED command (CMD53/Multi-block read).

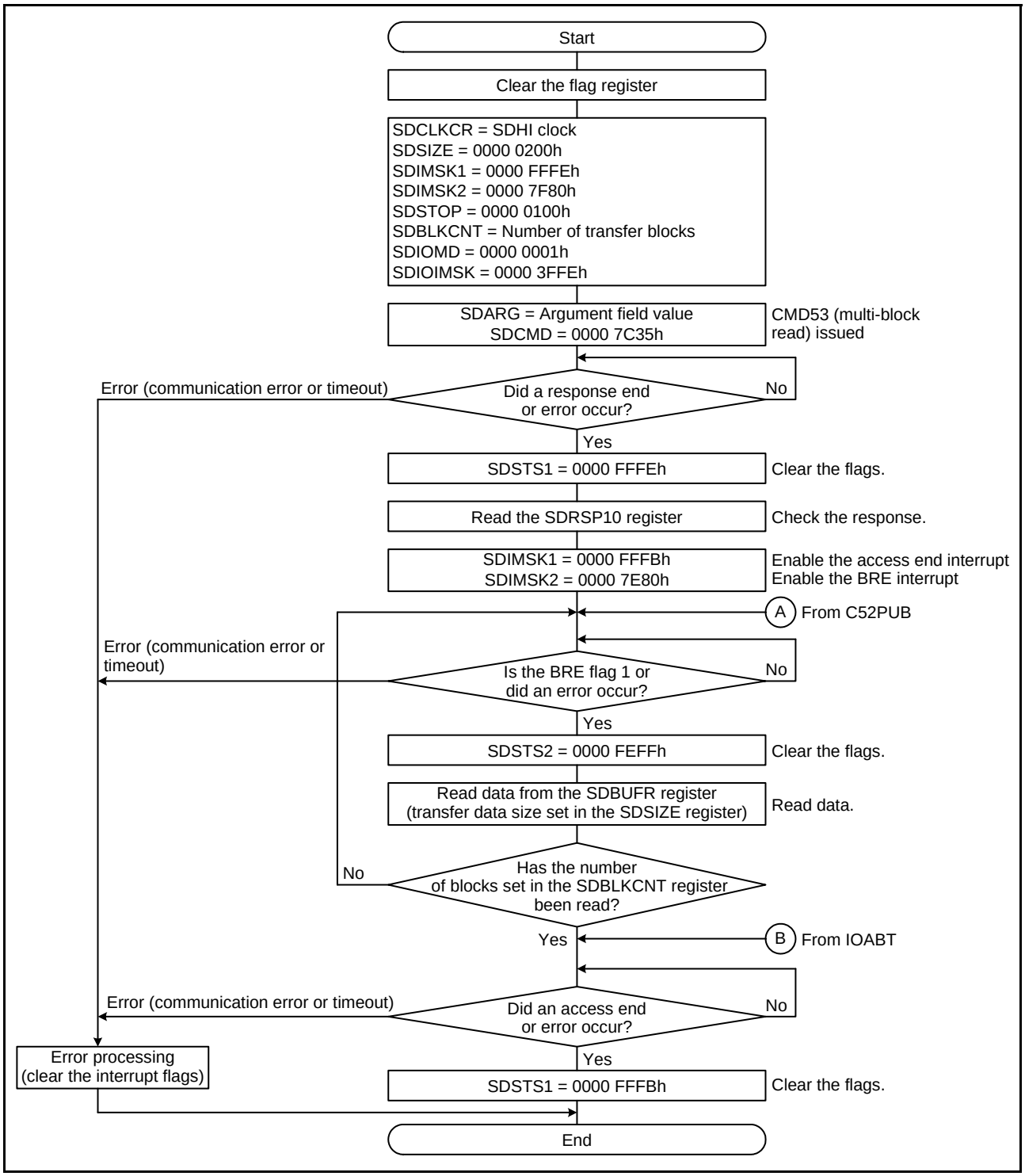


Figure 44.15 Issuing the IO_RW_EXTENDED Command (CMD53/Multi-Block Read)

Figure 44.16 shows an example of entering the read wait state and then issuing the SDIO none abort command (CMD52) during the IO_RW_EXTENDED command (CMD53/Multi-block read).

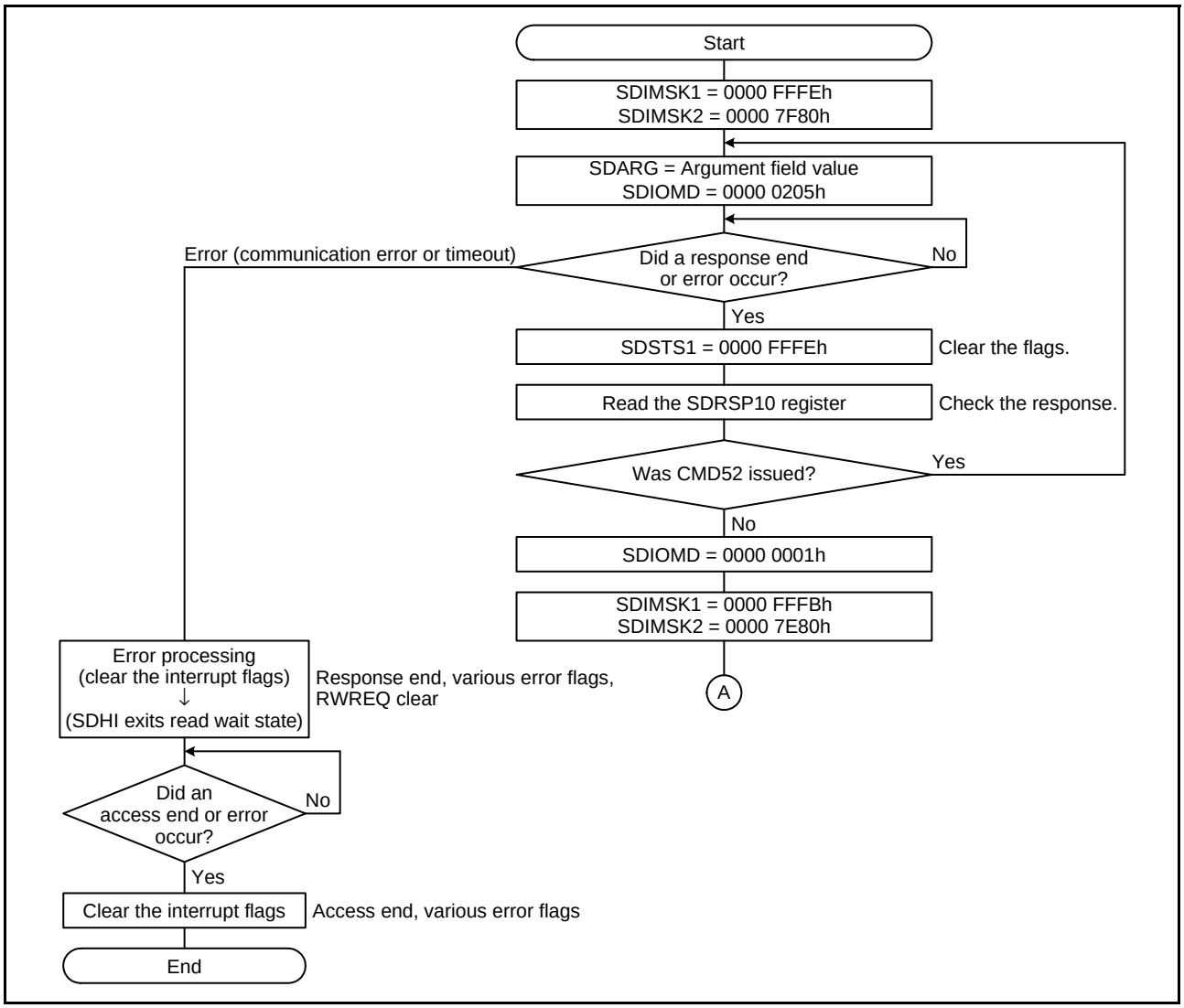


Figure 44.16 Entering the Read Wait State and Then Issuing the SDIO None Abort Command (CMD52) During the IO_RW_EXTENDED Command (CMD53/Multi-Block Read)

Figure 44.17 shows an example of SDIO abort (CMD52) issued during IO_RW_EXTENDED command (CMD53/ Multi-block read) sequence.

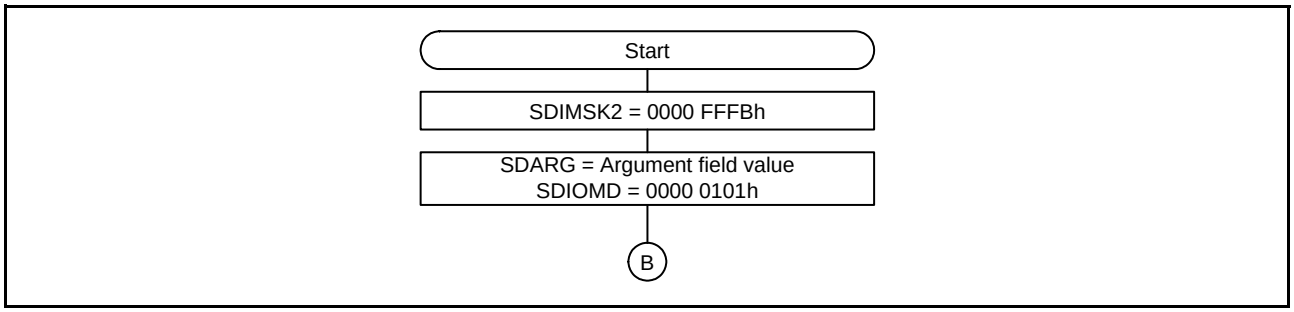


Figure 44.17 SDIO Abort (CMD52) Issued During IO_RW_EXTENDED Command (CMD53/Multi-Block Read) Sequence

44.3.6.9 IO_RW_EXTENDED (CMD53 Multi-Block Write)

Figure 44.18 shows an example of issuing the IO_RW_EXTENDED command (CMD53/Multi-block write).

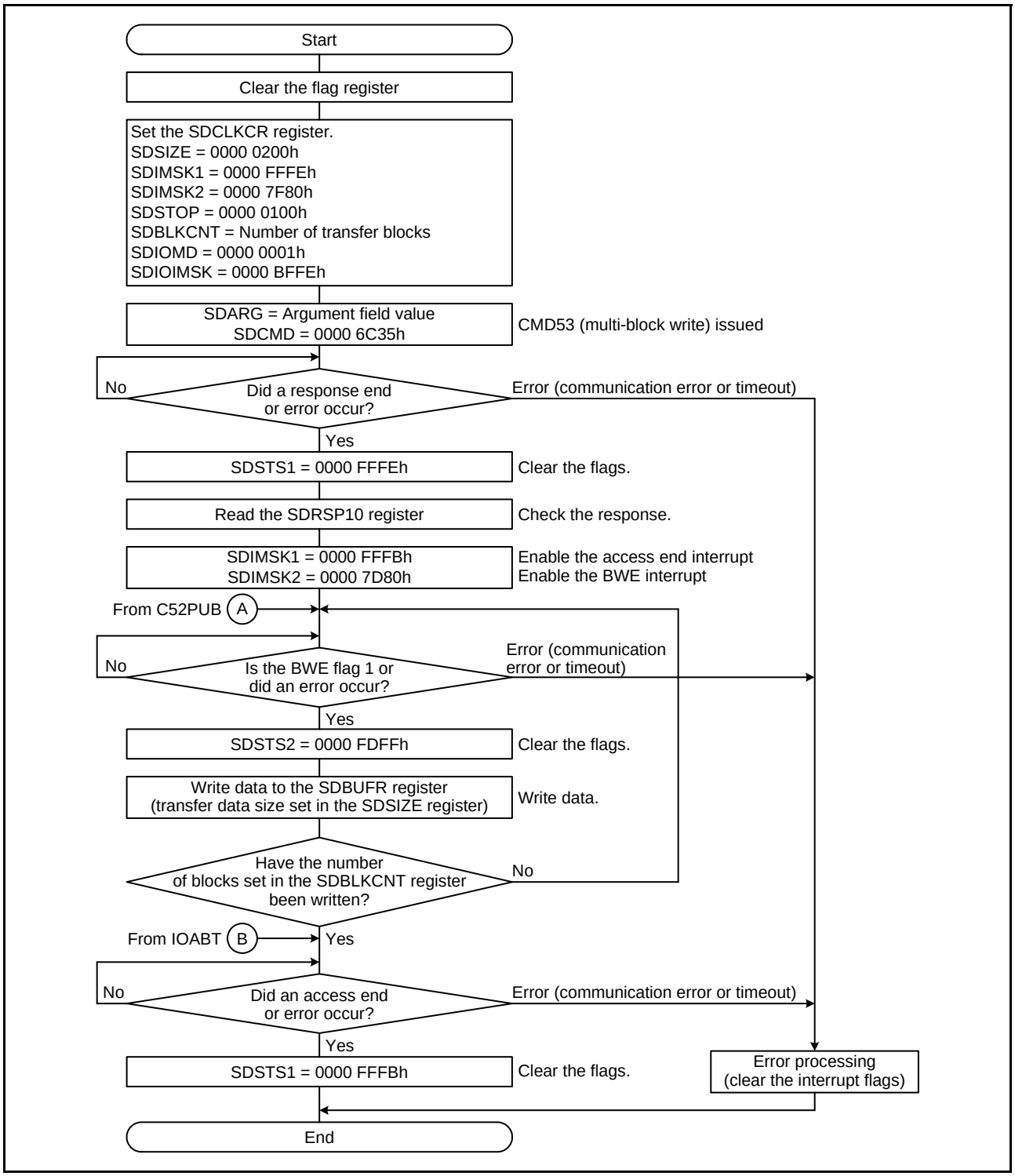


Figure 44.18 Issuing the IO_RW_EXTENDED Command (CMD53/Multi-Block Write)

Figure 44.19 shows SDIO none abort (CMD52) issued during IO_RW_EXTENDED command (CMD53/Multi-block write) sequence.

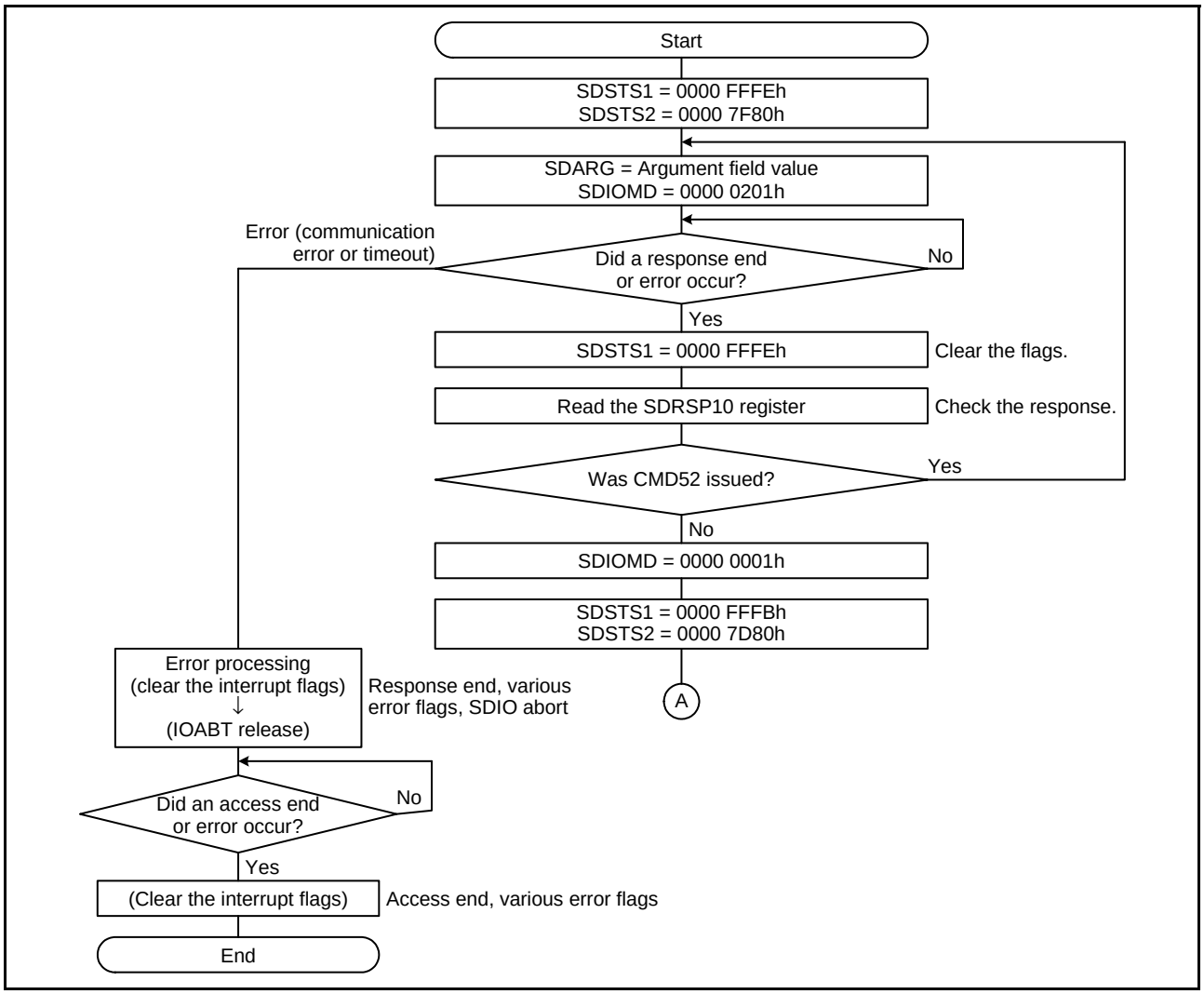


Figure 44.19 SDIO None Abort (CMD52) Issued During IO_RW_EXTENDED Command (CMD53/Multi-Block Write) Sequence

Figure 44.20 shows SDIO abort (CMD52) issued during IO_RW_EXTENDED command (CMD53) sequence.

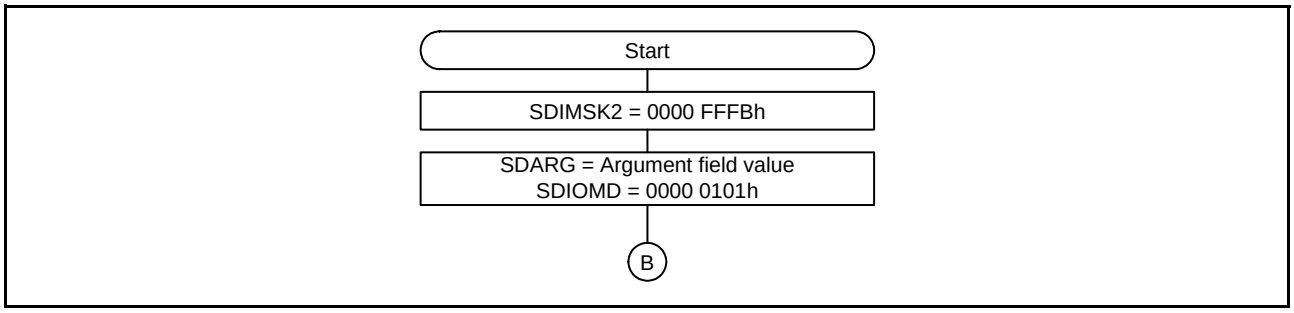


Figure 44.20 SDIO Abort (CMD52) Issued During IO_RW_EXTENDED Command (CMD53) Sequence

44.3.6.10 DMA Transfer

Figure 44.21 shows an example of data being transferred from the SDBUFR register after the CMD18 multi-block read command is issued.

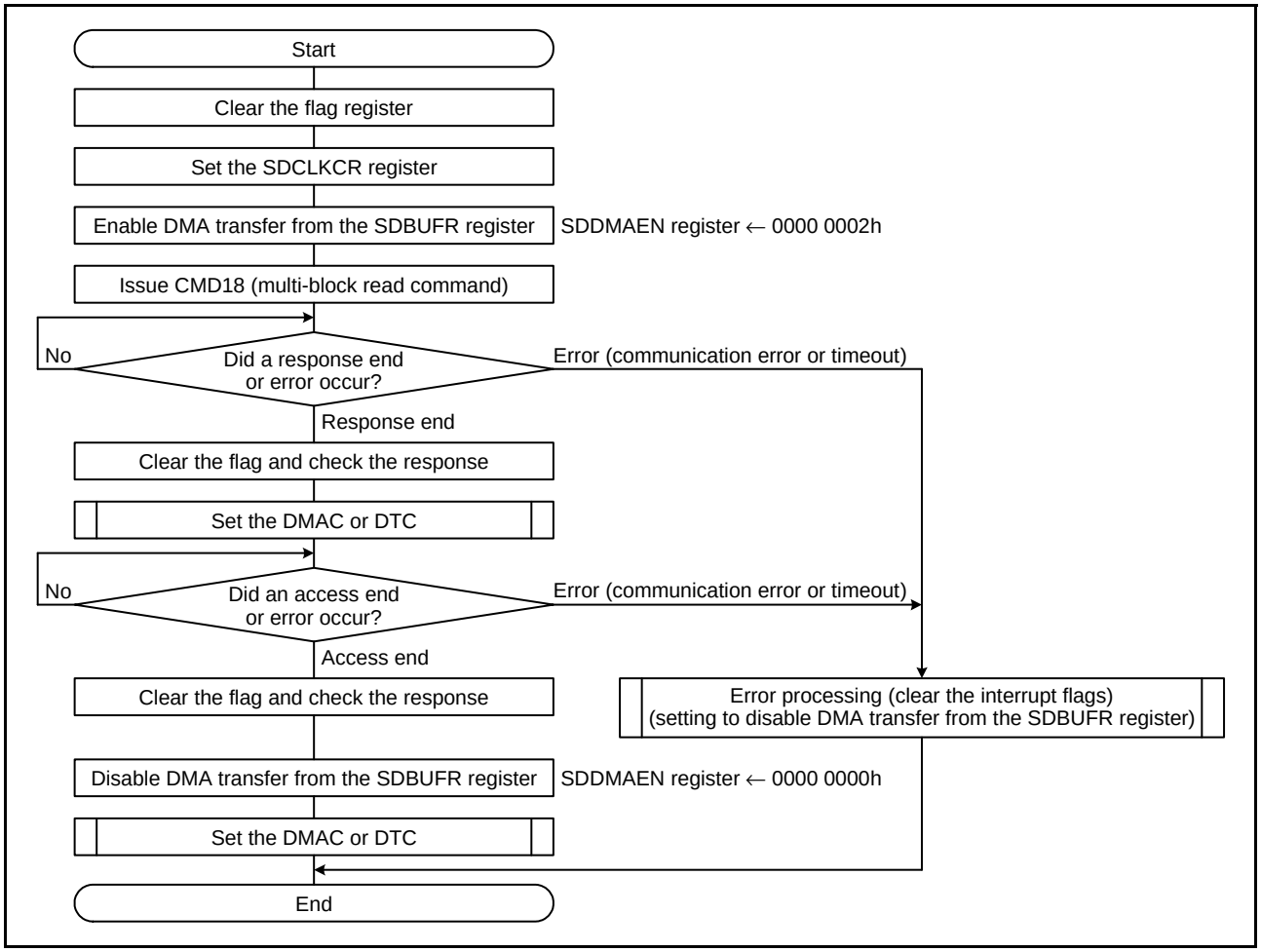


Figure 44.21 DMA Transfer After CMD18 is Issued

Figure 44.22 shows an example of data being DMA transferred to the SDBUFR register after the CMD25 multi-block write command is issued.

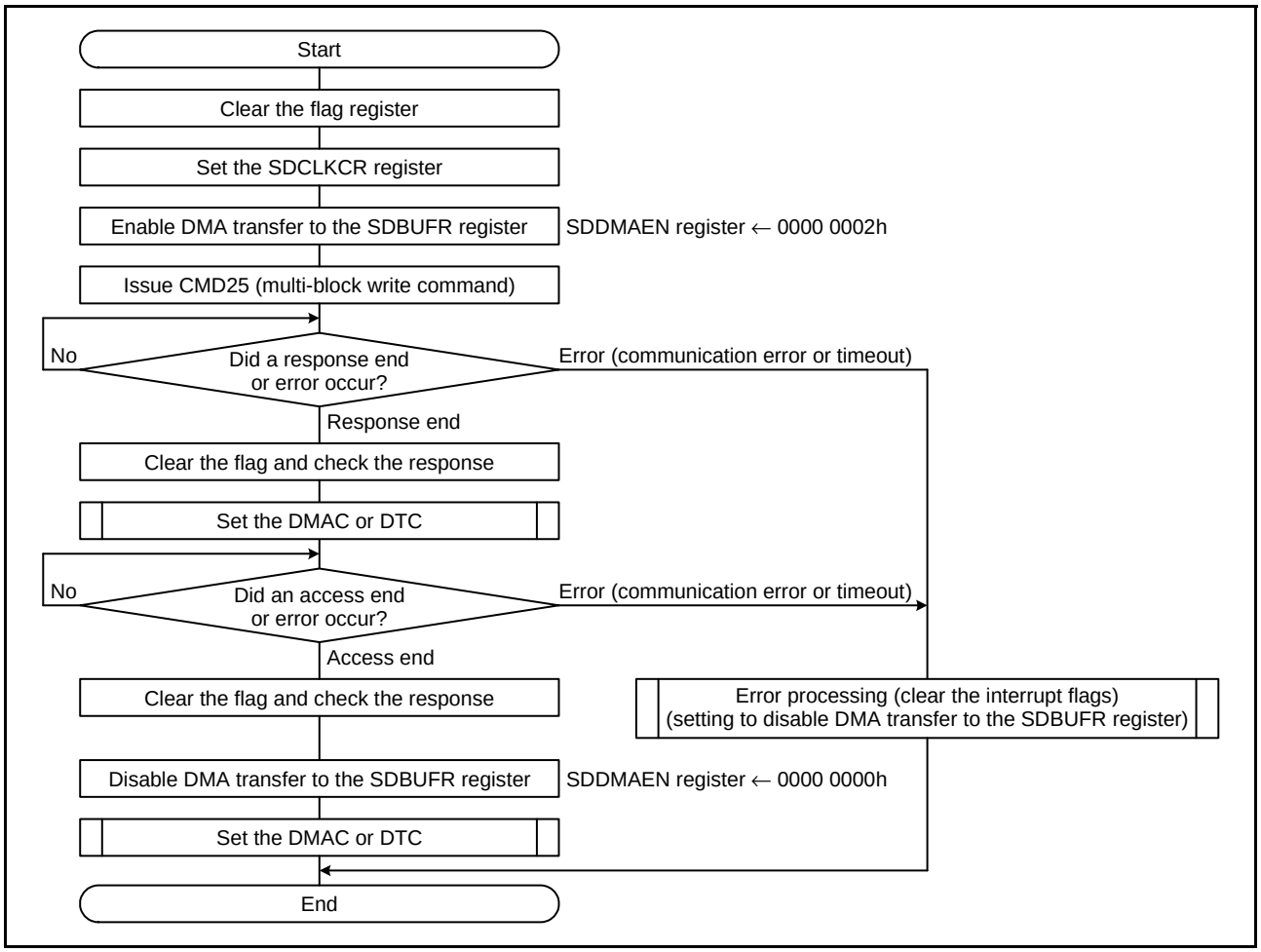


Figure 44.22 DMA Transfer After CMD25 is Issued

44.4 Interrupts

Table 44.8 lists the SDHI interrupt sources. When the status flags in registers SDSTS1, SDSTS2, and SDIOSTS become 1, if the corresponding bits in registers SDIMSK1, SDIMSK2, and SDIOIMSK are 0, the SDHI requests an interrupt. When clearing the status flags in registers SDSTS1, SDSTS2, and SDIOSTS, write 0 to the status flags to be cleared and write 1 to the states flags not being cleared.

Table 44.8 Interrupt Sources

Interrupt Source	Status Flag Register		Interrupt Mask/Enable Register		DMAC/DTC Triggerable
	Register symbol	Bit symbol	Register symbol	Bit symbol	
CACI	SDSTS1	ACEND	SDIMSK1	ACENDM	No
		RSPEND		RSPENDM	
	SDSTS2	ILA	SDIMSK2	ILAM	
		BWE		BWEM	
		BRE		BREM	
		RSPTO		RSPTOM	
		ILR		ILRM	
		ILW		ILWM	
		DTO		DTTOM	
		ENDE		ENDEM	
		CRCE		CRCEM	
		CMDE		CMDEM	
SDACI	SDIOSTS	EXWT	SDIOIMSK	EXWTM	No
		EXPUB52		EXPUB52M	
		IOIRQ		IOIRQM	
CDETI	SDSTS1	SDD3IN	SDIMSK1	SDD3INM	No
		SDD3RM		SDD3RMM	
		SDCDIN		SDCDINM	
		SDCDRM		SDCDRMM	
SBFAI	SDSTS2	BWE	SDDMAEN	DMAEN	Yes
		BRE			

44.4.1 DMA Transfer Triggered by Interrupt Requests

When the SBFAI interrupt is requested, DMA/DTC transfer can be used to write to or read the SDBUFR register. When using the SBFAI interrupt, set the SDDMAEN.DMAEN bit to 1, the SDIMSK2.BWEM bit to 1, and SDIMSK2.BREM bit to 1.

When the SDDMAEN.DMAEN bit is 1, if a write command is issued, the SDSTS2.BWE flag becomes 1; if a read command is issued, the SDSTS2.BRE flag becomes 1. At this point, the SBFAI interrupt request is output. When the last data of a block is transferred (one block is the transfer data size set in the SDSIZE.LEN[9:0] bits), the SBFAI interrupt request is cleared, and the SDSTS2.BWE flag or the SDSTS2.BRE flag becomes 0.

The SBFAI interrupt request is also cleared by following:

- The SDRST.SDRST bit is set to 0 (SDHI software reset).
- The SDSTOP.STP bit is set to 1.
- The SDIOMD.IOABT bit is set to 1.
- The SDDMAEN.DMAEN bit is set to 0.
  However, if the DMAEN bit is set to 1 again before the next command is written to the SDCMD register, the SBFAI interrupt request is output again.

The SBFAI interrupt request will not be cleared when a communication error or timeout occurs during DMA transfer. Perform error processing by software.

The SDSTS2.BWE flag and BRE flag will not become 0 when a communication error or timeout occurs, nor will they become 0 when the SDSTOP.STP bit and SDIOMD.IOABT bit are set to 1. If the SDSTS2.BWE flag or BRE flag remain set to 1, even if a write command or read command is issued, the SBFAI interrupt request will not be output, and the SDSTS2.BWE flag and BRE flag must be set to 0 before issuing the next command.

Table 44.9 lists the DMAC and DTC settings when performing DMA transfer.

Table 44.9 DMAC and DTC Settings When Performing DMA Transfer

Item		Setting Description
Transfer mode		Block transfer mode
Transfer data	1 data	32 bits
	Block size	Size set in the SDSIZE.LEN[9:0] bits divided by 4
Number of block transfers		Number of transfers set in the SDBLKCNT register

44.5 Notes on Using the SDHI

44.5.1 Illegal Read Access During a Multi-Block Read and How To Avoid It

When the multi-block read command (CMD18) is issued to read one or two blocks, if the response to CMD18 stored in the SDRSP10 register is read, the timing of the read access may cause the response to be read incorrectly.
Figure 44.23 shows examples of a normal read and an illegal read when using CMD18 to read two blocks.

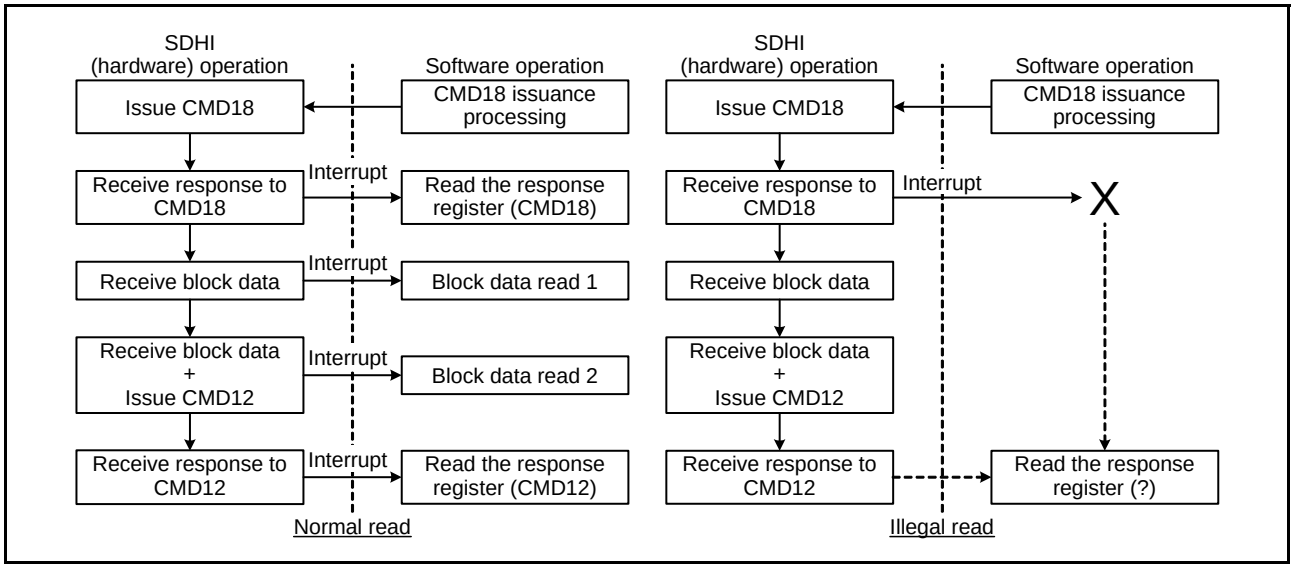


Figure 44.23 Multi-Block Read Processing When Reading Two Blocks

In the example of the illegal read, when the interrupt is generated by receiving the response to CMD18, the timing for reading the SDRSP10 register during the interrupt handling is delayed, and instead of the response for CMD18 being read, the response for CMD12 or the data during the response for CMD12 may be read.
This problem can be avoided by performing either of the following:

- When reading one block or two blocks, use a single block read command instead of a multi-block read command.
- When reading the response for CMD18, read the SDRSP54 register instead of the SDRSP10 register.

When using a multi-block read command to read three or more blocks, CMD12 will not be issued if no block data is read, which will prevent this problem from happening. Also, when using a multi-block write command, the above problem can be avoided by transmitting block data after reading the response to CMD25.

44.5.2 SDBUFR Register Illegal Write Error

When writing to the SDBUFR register after issuing a single block write command or a multi-block write command, write data for the size set by the SDSIZE.LEN[9:0] bits. If the amount of data written to the SDBUFR register is greater than the size set by the SDSIZE.LEN[9:0] bits, an illegal write error occurs in the SDBUFR register, and the SDSTS2.ILW flag becomes 1. However, the padding data included in the data being written to the SDBUFR register is ignored, so this error does not occur. For example, if the data size set by the SDSIZE.LEN[9:0] bits is an odd number, of the data written to the SDBUFR register, there is a remainder of 1 byte or 3 bytes. Although the extra data is written to the register, no error occurs. If the data size set by the SDSIZE.LEN[9:0] bits is an even number, and there is a remainder of 2 bytes, no error occurs even if these 2 bytes are written to the SDBUFR register.
If data written to the SDBUFR register is not transmitted, the SDSTS2.SDCLKCREN flag may remain set to 0. In this case, in order to set the SDSTS2.SDCLKCREN flag to 1, the SDRST.SDRST bit must be set to 0 and then set back to 1.

44.5.3 Automatic Control of the SDHI Clock Output

As per the SD card specifications, after MCU power-on, 74 cycles of the SDHI clock must be output from the host to the SD card before the card initialization command (CMD0) can be issued. Therefore, 74 cycles of the SDHI clock should be output from the SDHI to the SD card before enabling automatic control of the SDHI clock output.
When automatic control of the SDHI clock output is enabled, SDHI clock output stops if the command sequence is ended by a communication error or timeout. Therefore, if it is necessary to change the internal status of the SD card even after the command sequence ends, disable automatic control of the SDHI clock output and output the SDHI clock to the SD card.

44.5.4 Restrictions on Setting the C52PUB Bit During a Multi-Block Write Sequence

During a CMD53 multi-block write sequence, if the SDIOMD.C52PUB bit is set to 1, the SDHI issues CMD52 after the SD buffer becomes empty. To immediately issue CMD52, perform one of the procedures below to suspend writing to the SD buffer, and set the C52PUB bit to 1.

Procedure to suspend writing to the SD buffer when not performing DMA transfer (interrupt used)

1. Set the SDIMSK2.BWEM bit to 1 to disable the interrupt, and suspend writing to the SDBUFR register.
2. Set the SDIOMD.C52PUB bit to 1. Then, when the SD buffer becomes empty, the SDHI issues CMD52.
3. After receiving the response for CMD52, set the SDIMSK2.BWEM bit to 0 to enable the interrupt, and resume writing to the SDBUFR register.

Procedure to suspend writing to the SD buffer when performing DMA transfer

1. Configure settings to perform DMA transfer every [SDSIZE register setting value × *n* blocks], and suspend writing to the SDBUFR register before setting the SDIOMD.C52PUB bit (*n* = 1, 2, ...).
2. Set the SDIOMD.C52PUB bit to 1. Then, when the SD buffer becomes empty, the SDHI issues CMD52.
3. After receiving the response for CMD52, resume DMA transfer to the SDBUFR register.

44.5.5 Note on Setting the SDCLKCR Register

The SDCLKCR register cannot be written when the SDSTS2.SDCLKCREN flag is 0. Set the SDSTS2.SDCLKCREN flag to 1 before writing to the SDCLKCR register.

44.5.6 Writing to the SDSTOP Register During a Multi-Block Read Sequence

When the SDSTOP.SDBLKCNTEN bit is 1 during a multi-block read sequence, if the SDSTOP.STP bit is set to 1 and the command sequence is stopped, the command sequence may not be completed depending on when the SDSTOP.STP bit is set to 1. To avoid this problem, set the SDSTOP.STP bit and the SDSTOP.SDBLKCNTEN bit to 0 simultaneously. Note that at this time, the SDSTOP.SDBLKCNTEN bit should be set to 0 even if the SDSTS2.SDCLKCREN bit is 0.
If the SDSTOP.SDBLKCNTEN bit is not set to 0, the command sequence can be completed by setting the SDRST.SDRST bit to 0.
During a CMD53 multi-block transfer, when stopping data transfer by setting the SDIOMD.IOABT bit to 1, the SDSTOP.SDBLKCNTEN bit should remain set to 1.

44.5.7 Controlling Module Operation

SDHI operation is controlled by setting the MSTPCRD.MSTPD19 bit. Setting the MSTPD19 bit to 0 enables the SDHI; setting the MSTPD19 bit to 1 disables the SDHI. The SDHI is disabled after a reset. Registers in the SDHI can be accessed by setting the MSTPD19 bit to 0. Refer to section 11, Low Power Consumption for details.

45. Serial Sound Interface (SSIE)

45.1 Overview

This MCU has a serial sound interface (SSIE0). SSIE supports monaural and time-division multiplexed (TDM) formats in addition to standard audio data formats such as I^2S, left-justified, and right-justified.

Table 45.1 Features of SSIE

Item		Description
Number of channels		One channel, SSIE0
Transfer mode		• Master/slave • Transmission, reception, or transception
Data format		• I^2S format • Left-justified format • Right-justified format • Monaural format • TDM format
Serial data		• Fixed to MSB first • System word length: Selectable from among 8, 16, 24, 32, 48, 64, 128, or 256 bits • Data word length: Selectable from among 8, 16, 18, 20, 22, 24, or 32 bits • The level of padding bits is programmable. • Mute function
Bit clock (BCK)	In master mode	• Clock source: AUDIO_CLK • Frequency: Selectable from among 1/1, 1/2, 1/4, 1/6, 1/8, 1/12, 1/16, 1/24, 1/32, 1/48, 1/64, 1/96, or 1/128 of the AUDIO_CLK frequency • Continued supply or stopping while data transfer is halted is selectable.
	In master and slave modes	• Polarity is selectable as rising edges or falling edges.
LR clock (LRCK)		• Polarity is selectable as low or high. • Continued supply or stopping while data transfer is halted is selectable.
FIFO	Capacity	• Transmit FIFO: 4 bytes × 32 stages • Receive FIFO: 4 bytes × 32 stages
	Data alignment	• Alignment of data (left-justified or right-justified) in the FIFO is selectable.
Interrupts		• Data transfer error/idle state • Receive data full • Transmit data empty
Low power consumption function		• Module stop function • Master clock (MCK) supply stopping function

Table 45.2 lists and Figure 45.1 shows the terms and the definitions used in this chapter.

Table 45.2 Definitions of Terms

Term	Definition
Start trigger	First edge of the type specified by the SSICR.LRCKP bit of the signal on the SSILRCK pin after the start of data transfer
Frame boundary	Point where SSIE starts transferring the first data of a frame or the point where SSIE ends transferring the last data of the frame
Frame word length	Number of system words per frame (number of channels)
System word length	Number of transferred bits per channel
Data word length	Number of effective bits per channel
Master clock (MCK)	Clock for master mode input from the AUDIO_CLK pin

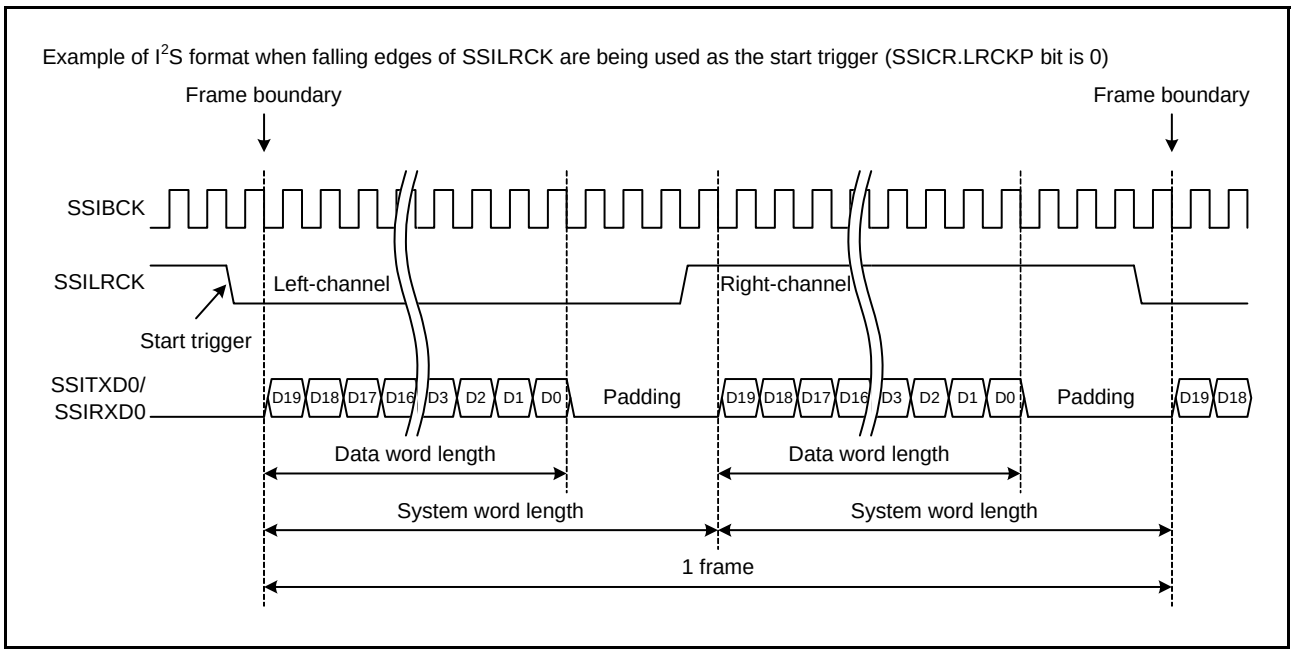


Figure 45.1 Definitions of Terms Related to Data Format

Figure 45.2 shows a block diagram of the SSIE0.

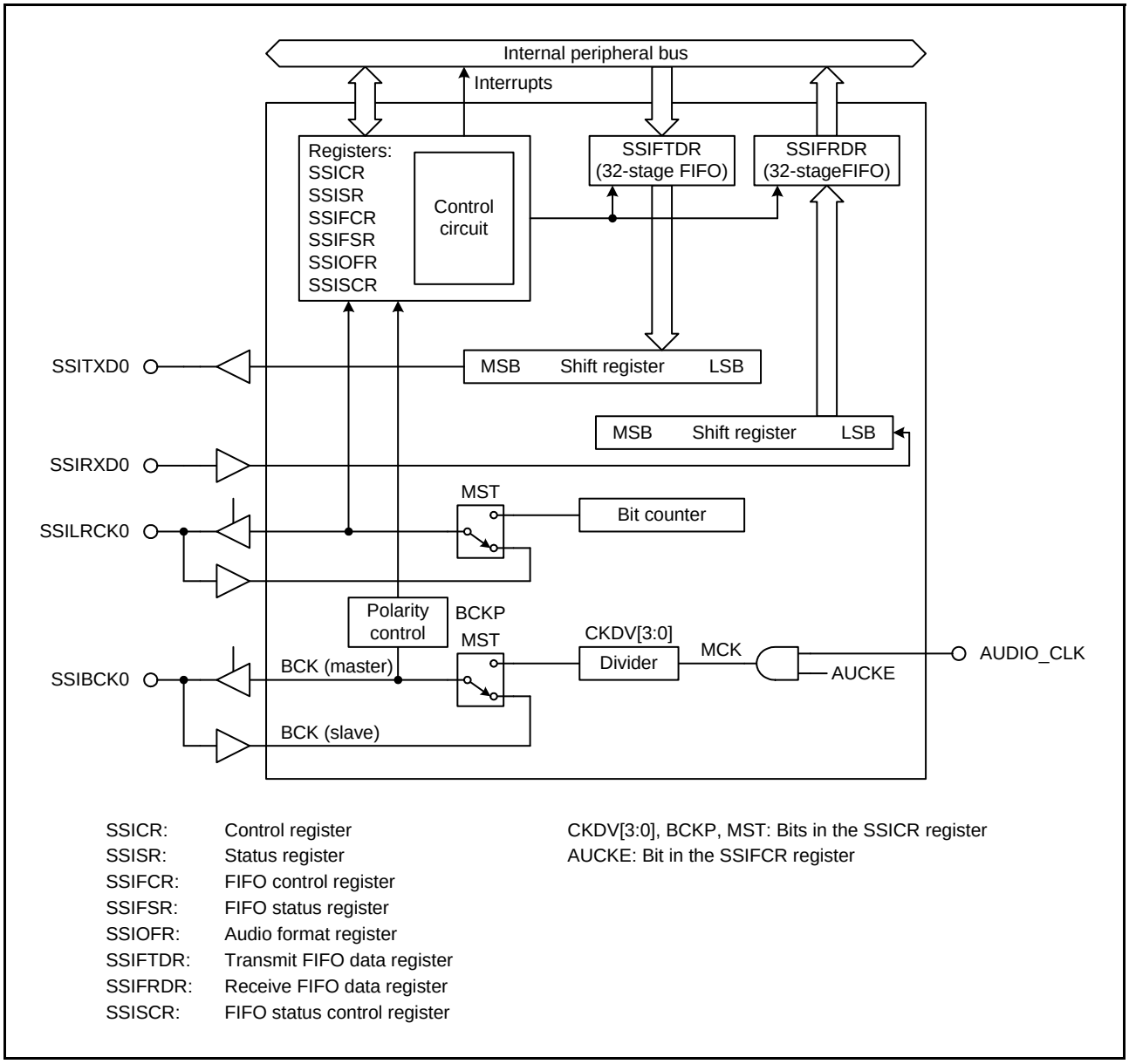


Figure 45.2 SSIE Block Diagram (SSIE0)

Table 45.3 lists the I/O pins of the SSIE.

Table 45.3 I/O pins of SSIE

Pin Name	I/O	Function
AUDIO_CLK	Input	Audio clock input
SSIBCK0	I/O	Channel 0 bit-clock input/output
SSILRCK0	I/O	Channel 0 LR clock input/output
SSITXD0	Output	Channel 0 data output
SSIRXD0	Input	Channel 0 data input

45.2 Register Descriptions

45.2.1 Control Register (SSICR)

Address(es): SSIE0.SSICR 0008 A500h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	—	FRM[1:0]		DWL[2:0]			SWL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	MST	BCKP	LRCKP	SPDP	SDTA	PDTA	DEL	CKDV[3:0]				MUEN	—	TEN	REN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	REN	Reception Enable*1	0: Reception disabled 1: Reception enabled	R/W
b1	TEN	Transmission Enable*1	0: Transmission disabled 1: Transmission enabled	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	MUEN	Mute	0: Unmute 1: Mute	R/W
b7 to b4	CKDV[3:0]	Bit Clock Division Ratio Select *2	Set the division ratio to generate a bit clock from MCK. b7 b4 0 0 0 0: No division 0 0 0 1: Divided by 2 0 0 1 0: Divided by 4 0 0 1 1: Divided by 8 0 1 0 0: Divided by 16 0 1 0 1: Divided by 32 0 1 1 0: Divided by 64 0 1 1 1: Divided by 128 1 0 0 0: Divided by 6 1 0 0 1: Divided by 12 1 0 1 0: Divided by 24 1 0 1 1: Divided by 48 1 1 0 0: Divided by 96 Settings other than above are prohibited.	R/W
b8	DEL	Serial Data Delay Select*2	Set the delay between the edges of SSILRCK and the input and output of data. 0: Delay of 1 cycle of SSIBCK 1: No delay	R/W
b9	PDTA	Data Alignment Select*2	Set the data alignment of the SSIFTDR and SSIFRDR registers. 0: Left-justified 1: Right-justified	R/W
b10	SDTA	Serial Data Alignment Select *2	0: I2S and left-justified format (data bits are transmitted and received before the padding bits) 1: Right-justified format (data bits are transmitted and received after the padding bits)	R/W
b11	SPDP	Padding Data Select*2	0: Padding bits are at the low level. 1: Padding bits are at the high level.	R/W
b12	LRCKP	LR Clock Polarity Select*2	0: SSILRCK is low for left channel. The frame boundary is a falling edge of SSILRCK. 1: SSILRCK is high for left channel. The frame boundary is a rising edge of SSILRCK.	R/W

Bit	Symbol	Bit Name	Description	R/W
b13	BCKP	Bit Clock Polarity Select*2	0: Output signals change at a falling edge of bit clock and input signals are sampled at a rising edge of bit clock. 1: Output signals change at a rising edge of bit clock and input signals are sampled at a falling edge of bit clock.	R/W
b14	MST	Master Mode*2	0: Slave mode 1: Master mode	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b18 to b16	SWL[2:0]	System Word Length Select*2	b18 b16 0 0 0: 8 bits 0 0 1: 16 bits 0 1 0: 24 bits 0 1 1: 32 bits 1 0 0: 48 bits 1 0 1: 64 bits 1 1 0: 128 bits 1 1 1: 256 bits	R/W
b21 to b19	DWL[2:0]	Data Word Length Select*2	b21 b19 0 0 0: 8 bits 0 0 1: 16 bits 0 1 0: 18 bits 0 1 1: 20 bits 1 0 0: 22 bits 1 0 1: 24 bits 1 1 0: 32 bits 1 1 1: Setting prohibited.	R/W
b23, b22	FRM[1:0]	Frame Word Length Select*2	Set a number of system words per a frame. b23 b22 0 0: 2 system words (I^2S, left-justified, or right-justified) or 1 system word (monaural) 0 1: 4 system words (TDM format only) 1 0: 6 system words (TDM format only) 1 1: 8 system words (TDM format only)	R/W
b24	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b25	IIEN	Idle State Interrupt Enable	0: Idle state interrupt disabled 1: Idle state interrupt enabled	R/W
b26	ROIEN	Receive Overflow Interrupt Enable	0: Receive overflow interrupt disabled 1: Receive overflow interrupt enabled	R/W
b27	RUIEN	Receive Underflow Interrupt Enable	0: Receive underflow interrupt disabled 1: Receive underflow interrupt enabled	R/W
b28	TOIEN	Transmit Overflow Interrupt Enable	0: Transmit overflow interrupt disabled 1: Transmit overflow interrupt enabled	R/W
b29	TUIEN	Transmit Underflow Interrupt Enable	0: Transmit underflow interrupt disabled 1: Transmit underflow interrupt enabled	R/W
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. After the TEN or REN bit is rewritten, confirm that the SSISR.IIRQ flag is in the desired status (when transmission or reception is enabled, confirm that the IIRQ flag is 0; when transmission and reception are disabled, confirm that the IIRQ flag is 1).

Note 2. Rewrite these bits while SSIE is in idle state (SSISR.IIRQ flag = 1). If the value of these bits is changed while data transfer is in progress, the operation after the change is not guaranteed.

TEN Bit (Transmission Enable) and REN Bit (Reception Enable)

These bits enable/disable transmission and reception. When one of these bits is set to 1, the given type of data transfer starts in synchronization with a start trigger of the SSILRCK signal. When the bit is set to 0, the data transfer stops at the next frame boundary. To use an SSIE for both transmission and reception, set both the TEN and REN bits to 1 simultaneously. To stop the data transfer, set both the TEN and REN bits to 0 simultaneously.

To switch among transmission, reception, and transception, do so in the idle state.

MUEN Bit (Mute)

This bit is used to mute audio output by driving the SSITXD0 pin low. When this bit is changed during a frame, the setting takes effect at the next frame boundary.
The MUEN bit only controls fixing of the SSITXD0 pin. Status flags are set and interrupt signals are generated regardless of the setting of the MUEN bit.
Changing the value of this bit must be performed after setting the data format to be used.

CKDV[3:0] Bits (Bit Clock Division Ratio Select)

These bits specify the divisor for the generation of a bit clock (BCK) from the master clock (MCK). This bit is disabled in slave mode.
Set these bits while the SSIFCR.AUCKE bit is 0.

DEL Bit (Serial Data Delay Select)

When this bit is 0, a delay of one cycle of the bit clock is inserted between edges of SSILRCK0 and SSITXD0 and SSIRXD0. When 1, no delay is inserted.
Set this bit to 0 for the I^2S format and to 1 for the left-justified or right-justified format. Set this bit in conformance with the other party device for the monaural and TDM formats.When the monaural format is used, setting of this bit changes the high width of SSILRCK.

PDTA Bit (Data Alignment Select)

This bit specifies the alignment of the effective bits in the SSIFTDR and SSIFRDR registers. This bit is valid when the data word length 18 to 24 bits (the DWL[2:0] bits = 010b to 101b). Set this bit to 0, when the other data word length is selected.
Figure 45.3 shows the data alignment of the SSIFTDR and SSIFRDR registers.

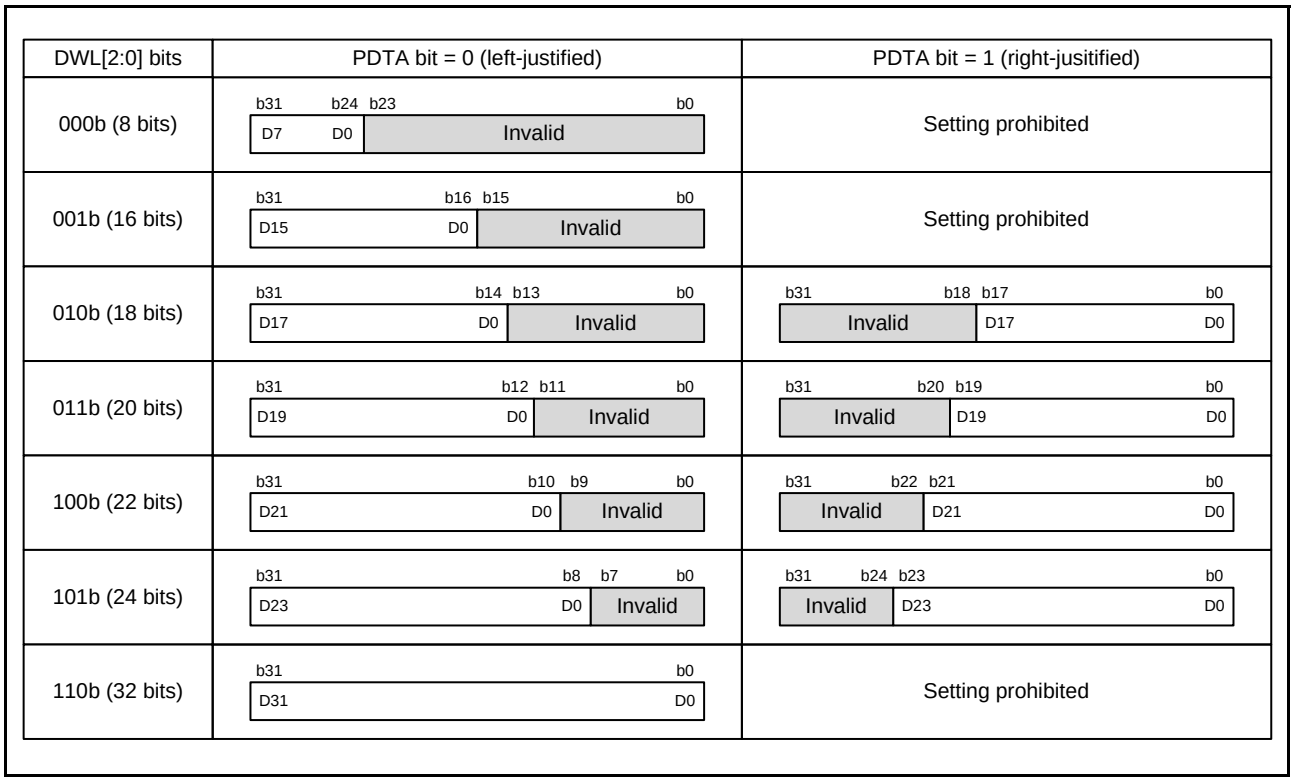


Figure 45.3 Data Alignment in the SSIFTDR and SSIFRDR Registers

SDTA Bit (Serial Data Alignment Select)

This bit specifies the order of data bits and padding bits for transception in system words. SSIE transmits and receives data bits first and then padding bits when this bit is 0 and padding bits first and then data bits when this bit is 1. This bit is invalid for data format without padding bits.

LRCKP Bit (LR Clock Polarity Select)

This bit specifies the initial status and polarity of the SSILRCK pin. Table 45.4 lists the settings of this bit and the corresponding states of the LR clock. SSIE uses the SSILRCK signal as a start trigger in slave mode.
Set this bit to 0 when the I^2S format is selected and 1 when the other data formats are selected.

Table 45.4 The Value of the LRCKP Bit and Responses to the States of the LR clock

LRCKP bit	Initial State	Left Channel	Right Channel	Frame Boundary
0	High	Low	High	Falling edge
1	Low	High	Low	Rising edge

BCKP Bit (Bit Clock Polarity Select)

This bit specifies the polarity of the bit clock. Table 45.5 lists the settings of this bit and the corresponding timings of the respective signals.
Set this bit while the SSIFCR.AUCKE bit is 0.

Table 45.5 The Value of the BCKP bit and the Bit Clock Polarity

BCKP bit	Output of SSILRCK and SSITXD0	Input of SSILRCK and SSIRXD0
0	The signals change on falling edges of SSIBCK.	The signals are sampled on rising edges of SSIBCK.
1	The signals change on rising edges of SSIBCK.	The signals are sampled on falling edges of SSIBCK.

SWL[2:0] Bits (System Word Length Select)

These bits specify the number of bits in a system word. The setting of these bits determines the frequency ratio of the bit clock to the LR clock. Then number of the padding bits is obtained from the difference in the SWL[2:0] and DWL[2:0] bits. Refer to Table 45.8 for details.

DWL[2:0] Bits (Data Word Length Select)

These bits specify the number of bits of valid data in a system word. The data word length must not exceed the system word length. For details, refer to Table 45.8.

FRM[1:0] Bits (Frame Word Length Select)

These bits specify the number of system words in a frame.

Set these bits to 00b when the I^{2}S, left-justified, right-justified, or monaural format is selected. Do not set these bits to 00b when the TDM format is selected.

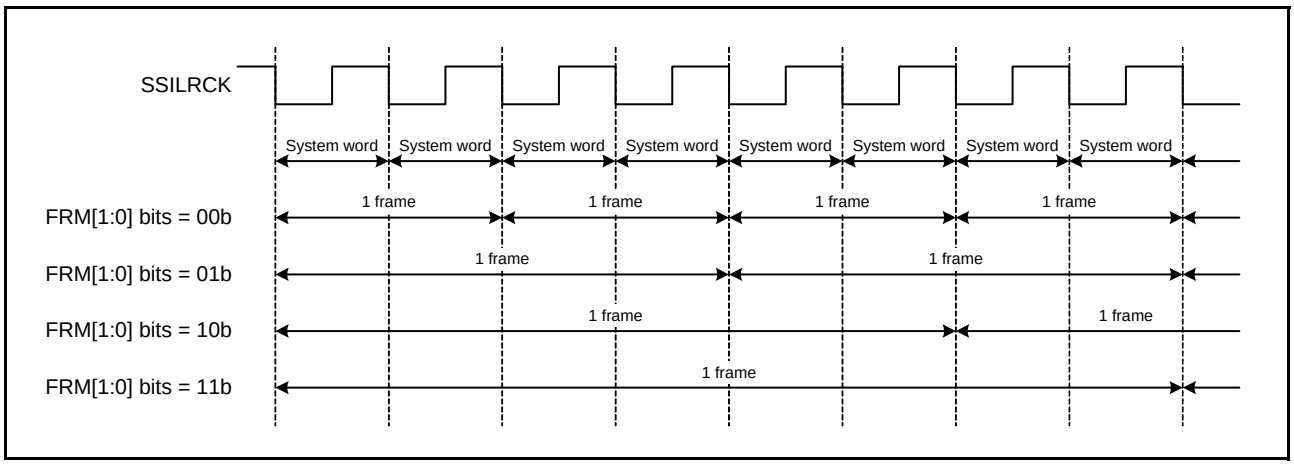


Figure 45.4 Frame Word Length

IIEN Bit (Idle State Interrupt Enable)

This bit enables or disables the idle state interrupts.

While this bit is 1, an interrupt request is output when the SSISR.IIRQ flag becomes 1. An interrupt request is also output when this bit is changed from 0 to 1 while the IIRQ flag is 1.

ROIEN Bit (Receive Overflow Interrupt Enable)

This bit enables or disables interrupts receive overflow interrupts.

While this bit is 1, an interrupt request is output when the SSISR.ROIRQ flag becomes 1. An interrupt request is also output when this bit is changed from 0 to 1 while the ROIRQ flag is 1.

RUIEN Bit (Receive Underflow Interrupt Enable)

This bit enables/disables the receive underflow interrupts.

While this bit is 1, an interrupt request is output when the SSISR.RUIRQ flag becomes 1. An interrupt request is also output when this bit is changed from 0 to 1 while the RUIRQ flag is 1.

TOIEN Bit (Transmit Overflow Interrupt Enable)

This bit enables/disables the transmit overflow interrupts.

While this bit is 1, an interrupt request is output when the SSISR.TOIRQ flag becomes 1. An interrupt request is also output when this bit is changed from 0 to 1 while the TOIRQ flag is 1.

TUIEN Bit (Transmit Underflow Interrupt Enable)

This bit enables/disables the transmit underflow interrupts.

While this bit is 1, an interrupt request is output when the SSISR.TUIRQ flag becomes 1. An interrupt request is also output when this bit is changed from 0 to 1 while the TUIRQ flag is 1.

45.2.2 Status Register (SSISR)

Address(es): SSIE0.SSISR 0008 A504h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Bit Name	Description	R/W
b24 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25	IIRQ	Idle State Flag	0: Transfer state 1: Idle state	R
b26	ROIRQ	Receive Overflow Flag	0: No receive overflows occurred. 1: A receive overflow has occurred.	R/W
b27	RUIRQ	Receive Underflow Flag	0: No receive underflows occurred. 1: A receive underflow has occurred.	R/W
b28	TOIRQ	Transmit Overflow Flag	0: No transmit overflows occurred. 1: A transmit overflow has occurred.	R/W
b29	TUIRQ	Transmit Underflow Flag	0: No transmit underflows occurred. 1: A transmit underflow has occurred.	R/W
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register indicates the SSIE status.

IIRQ Flag (Idle State Flag)

This flag indicates whether SSIE is in the idle state or transfer state.

[Setting condition]

- When a transmission of a frame is completed after changing the SSICR.TEN bit from 1 to 0.
- When a reception of a frame is completed after changing the SSICR.REN bit from 1 to 0.
- When a transmission of a frame is completed after changing the SSICR.TEN and SSICR.REN bits from 1 to 0.

[Clearing condition]

- When a start trigger is generated by writing a transmit data in the SSIFTDR register, while the SSICR.TEN bit is 1.
- When a start trigger is detected while the SSICR.REN bit is 1.
- When a start trigger is generated by writing a transmit data to the SSIFTDR register, while the SSICR.TEN and SSICR.REN bits are 1.

ROIRQ Flag (Receive Overflow Flag)

This flag indicates a receive overflow. This flag indicates that the data processing of the received data is delayed. The received data is not transferred to the receive FIFO if a receive overflow occurs. This flag is not cleared by resetting a receive FIFO.
[Priority order for setting and clearing]
Setting is prioritized.
[Setting condition]

- When a reception of the next data is completed while the receive FIFO is full.

[Clearing condition]

- When 0 is written to this flag after confirming this flag is 1.
- When the SSICR.REN bit is changed from 0 to 1.

RUIRQ Flag (Receive Underflow Flag)

This flag indicates a receive underflow. This flag indicates that the SSIFRDR register is read while the receive FIFO is empty. Data read from the SSIFRDR register is invalid if a receive underflow occurs. This flag is not cleared by resetting a receive FIFO. Reading the SSIFRDR register while resetting the receive FIFO does not lead to setting of this flag.
[Priority order for setting and clearing]
Setting is prioritized.
[Setting condition]

- When the SSIFRDR register is read while the receive FIFO is empty.

[Clearing condition]

- When 0 is written to this flag after confirming this flag is 1.
- When the SSICR.REN bit is changed from 0 to 1.

TOIRQ Flag (Transmit Overflow Flag)

This flag indicates a transmit overflow. This flag indicates that a data is written to the SSIFTDR register while the transmit FIFO is full. The written data is discarded. This flag is not cleared by resetting a transmit FIFO.
[Priority order for setting and clearing]
Setting is prioritized.
[Setting condition]

- When a data is written to the SSIFTDR register while the transmit FIFO is full.

[Clearing condition]

- When 0 is written to this flag after confirming this flag is 1.
- When the SSICR.TEN bit is changed from 0 to 1.

TUIRQ Flag (Transmit Underflow Flag)

This flag indicates a transmit underflow. This flag indicates that supply of the transmit data was not enough to the transmission of the frame. The SSITXD0 pin becomes low if a transmit underflow occurs, and remains low even if the flag is cleared. This flag is not cleared by resetting a transmit FIFO.
[Priority order for setting and clearing]
Setting is prioritized.
[Setting condition]

- When a data transfer to the transmit shift register occurs while the transmit FIFO is empty.

[Clearing condition]

- When 0 is written to this flag after confirming this flag is 1.
- When the SSICR.TEN bit is changed from 0 to 1.

45.2.3 FIFO Control Register (SSIFCR)

Address(es): SSIE0.SSIFCR 0008 A510h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	AUCKE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSIRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	BSW	—	—	—	—	—	—	—	TIE	RIE	TFRST	RFRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RFRST	Receive FIFO Reset*1	0: The receive FIFO is released from the reset state. 1: The receive FIFO is reset.	R/W
b1	TFRST	Transmit FIFO Reset*1	0: The transmit FIFO is released from the reset state. 1: The transmit FIFO is reset.	R/W
b2	RIE	Receive Data Full Interrupt Enable	0: Receive data full interrupt is disabled. 1: Receive data full interrupt is enabled.	R/W
b3	TIE	Transmit Data Empty Interrupt Enable	0: Transmit data empty interrupt is disabled. 1: Transmit data empty interrupt is enabled.	R/W
b10 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11	BSW	Byte Swap*1	0: Byte swapping is disabled 1: Byte swapping is enabled	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	SSIRST	Software Reset	0: The software reset is released. 1: A software reset is applied.	R/W
b30 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31	AUCKE	MCK Supply Enable*1	0: MCK is stopped. 1: MCK is supplied.	R/W

Note 1. Rewrite these bits while SSIE is in idle state (SSISR.IIRQ flag = 1). If the value of these bits is changed while data transfer is in progress, the operation after the change is not guaranteed.

RFRST Bit (Receive FIFO Reset)

This bit is used to reset the receive FIFO. Writing 1 to this bit resets the receive FIFO and initializes the RDC[5:0] bits and RDF flag in the SSIFSR register. Since this bit is not automatically cleared, write 0 to this bit to release from the reset state. After writing 0 to this bit, be sure to check that this bit is 0 before starting the next procedure.
This bit is subject to the software reset by the SSIRST bit. Because the software reset by the SSIRST bit has priority over the reset by this bit, this bit cannot be set to 1 when the SSIRST bit is 1.

TFRST Bit (Transmit FIFO Reset)

This bit is used to reset the transmit FIFO. Writing 1 to this bit resets the transmit FIFO and initializes the TDC[5:0] bits and TDE flag in the SSIFSR register. Since this bit is not automatically cleared, write 0 to this bit to release from the reset state. After writing 0 to this bit, be sure to check that this bit is 0 before starting the next procedure.
This bit is subject to the software reset by the SSIRST bit. Because the software reset by the SSIRST bit has priority over the reset by this bit, this bit cannot be set to 1 when the SSIRST bit is set.

RIE Bit (Receive Data Full Interrupt Enable)

This bit enables or disables a receive data full interrupt. Set this bit to 1 after specifying the condition for generating a receive data full interrupt in the SSISCR.RDFS[4:0] bits.

TIE Bit (Transmit Data Empty Interrupt Enable)

This bit enables or disables a transmit data empty interrupt. Set this bit to 1 after specifying the condition for generating a transmit data empty interrupt in the SSISCR.TDES[4:0] bits.

BSW Bit (Byte Swap)

The BSW bit enables or disables byte swapping when the SSIFTDR is written and the SSIFRDR register is read. This bit is valid only when the SSIFTDR and SSIFRDR register are accessed in 16-bit or 32-bit units. For details, refer to Figure 45.5.

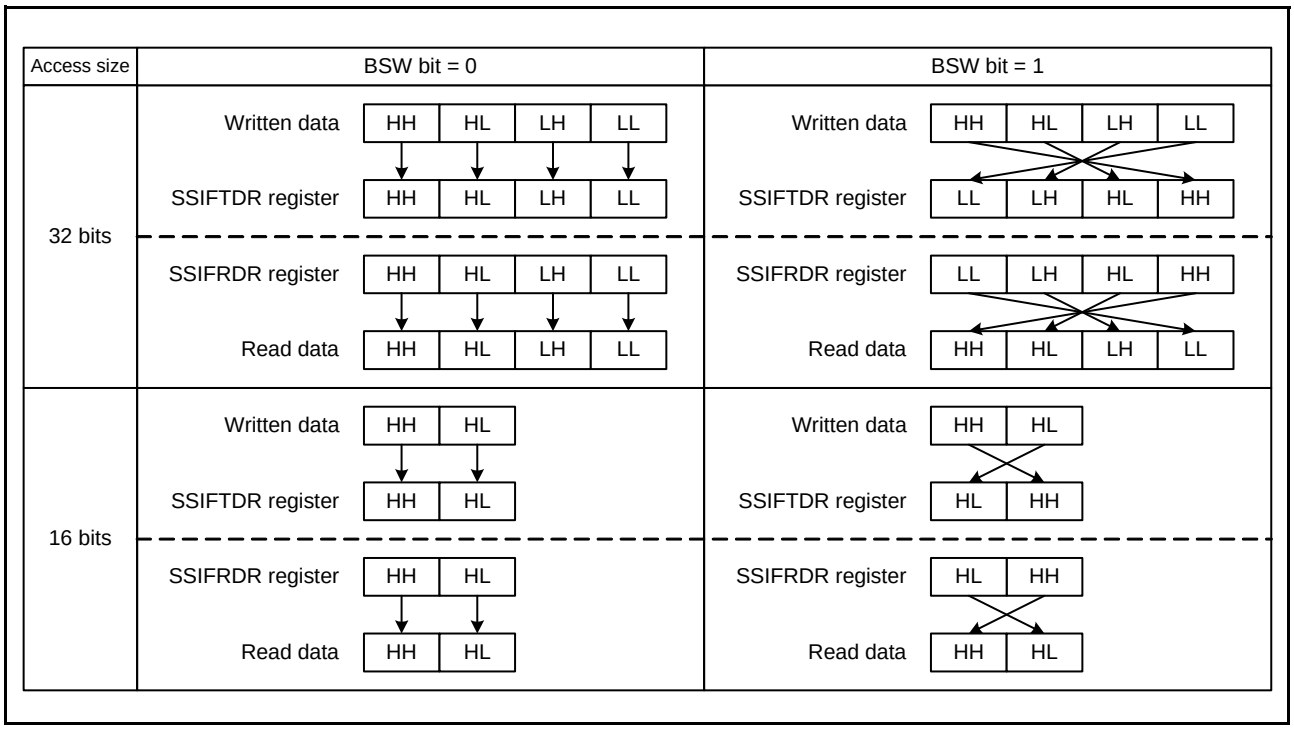


Figure 45.5 Examples of Byte-Swapping Operations

SSIRST Bit (Software Reset)

This bit is used to reset SSIE. Writing 1 to this bit initializes the internal state of SSIE. The registers and bits to be reset by setting this bit are listed below.

- SSICR.MUEN, TEN, and REN bits
- SSISR register
- SSIFCR.TFRST and RFRST bits
- SSIFSR register
- SSIFTDR register
- SSIFRDR register

Because this bit is not automatically cleared, write 0 to this bit to release from the reset state. After writing 0 to this bit, be sure to check that this bit is 0 before starting the next procedure.

Because writing 1 to this bit during data transfer leads to initialization by software reset regardless of the bit clock, be careful to stop data transfer by using this bit.

AUCKE Bit (MCK Supply Enable)

This bit enables/disables supply of MCK while in master-mode. The setting value of this bit is reflected at the timing shown in Figure 45.6.

Changes to settings related to MCK (the settings of the MST, BCKP, and CKDV[3:0] bits in the SSICR register) should only proceed while this bit is 0.

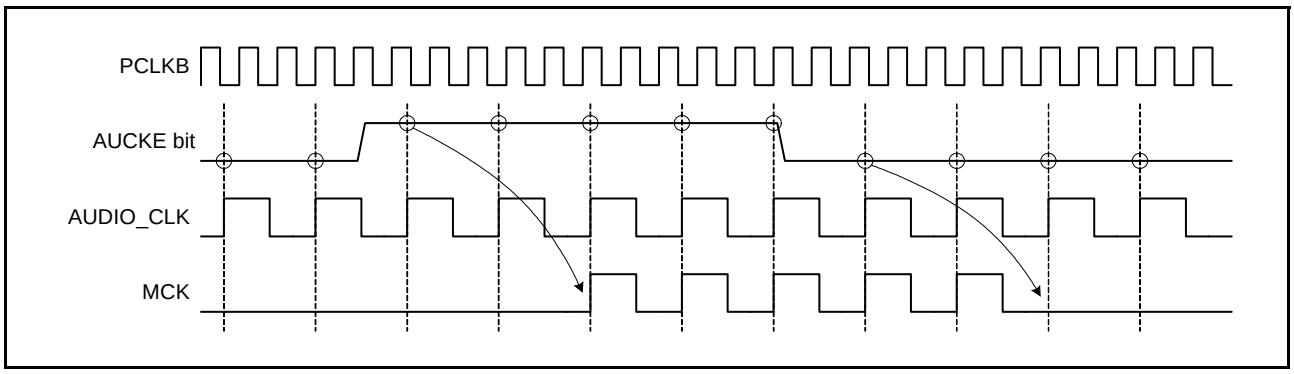


Figure 45.6 Supplying and Stopping MCK

45.2.4 FIFO Status Register (SSIFSR)

Address(es): SSIE0.SSIFSR 0008 A514h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	TDC[5:0]						—	—	—	—	—	—	—	TDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

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

Bit	Symbol	Bit Name	Description	R/W
b0	RDF	Receive Data Full Flag	0: The number of received data stored in the receive FIFO does not exceed the value of the SSISCR.RDFS[4:0] bits. 1: The number of received data stored in the receive FIFO exceeds the value of the RDFS[4:0] bits.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	RDC[5:0]	Receive FIFO Data Count	Number of data stored in the receive FIFO	R
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	TDE	Transmit Data Empty Flag	0: The number of spaces available in the transmit FIFO does not exceed the value of the SSISCR.TDES[4:0] bits 1: The number of spaces available in the transmit FIFO exceeds the value of TDES[4:0] bits.	R/W
b23 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b29 to b24	TDC[5:0]	Transmit FIFO Data Count	Number of data stored in the transmit FIFO	R
b31, b30	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register indicates the status of the transmit FIFO and the receive FIFO.

RDF Flag (Receive Data Full Flag)

This flag indicates that the number of unread data in the receive FIFO is more than the value of the SSISCR.RDFS[4:0] bits.

[Priority order for setting and clearing]

Clearing is prioritized.

[Setting condition]

- When the number of data stored in the receive FIFO exceeds the value of the SSISCR.RDFS[4:0] bits.

[Clearing condition]*1

- When 0 is written to this flag after confirming this flag is 1.
- When a received data is read from the SSIFRDR register by DMA/DTC transfer (the last transfer of the block if using block transfer).

Note 1. This flag is cleared by a software reset and receive FIFO reset.

RDC[5:0] Bits (Receive FIFO Data Count)

These bits indicate the number of valid data that are stored in the receive FIFO. When this bit is 00h, there is no received data. When 20h, the receive FIFO is full and there is no space available.

TDE Flag (Transmit Data Empty Flag)

This flag indicates that the number of spaces available in the transmit FIFO is more than the value of the SSISCR.TDES[4:0] bits.
[Priority order for setting and clearing]
Clearing is prioritized.
[Setting condition]

- When the number of spaces available in the transmit FIFO exceeds the value of the SSISCR.TDES[4:0] bits.

[Clearing condition]*1

- When 0 is written to this flag after confirming this flag is 1.
- When a transmit data is written to the SSIFTDR register by DMA/DTC transfer (the last transfer of the block if using block transfer).

Note 1. This flag is cleared by a software reset and transmit FIFO reset.

TDC[5:0] Bits (Transmit FIFO Data Count)

These bits indicate the number of valid data that are stored in the transmit FIFO. When this bit is 00h, there is no data to be transmitted. When 20h, the transmit FIFO is full and there is no space available.

45.2.5 Transmit FIFO Data Register (SSIFTDR)

- When accessing in longword size

Address(es): SSIE0.SSIFTDR 0008 A518h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

- When accessing in word size

Address(es): SSIE0.SSIFTDR.H 0008 A518h

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

- When accessing in byte size

Address(es): SSIE0.SSIFTDR.HH 0008 A518h

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

This register is a write-only register used to write a transmit data to the 32-stage and 4-byte-wide transmit FIFO. Access to this register according to the access size listed in Table 45.6.

Table 45.6 Access Size to the SSIFTDR and SSIFRDR Registers

		Access Size		
SSICR.DWL[2:0] Bits	**Data Word Length**	**Byte**	**Word**	**Longword**
000b	8	Available	Not available	Not available
001b	16	Not available	Available	Not available
010b	18	Not available	Not available	Available
011b	20	Not available	Not available	Available
100b	22	Not available	Not available	Available
101b	24	Not available	Not available	Available
110b	32	Not available	Not available	Available

45.2.6 Receive FIFO Data Register (SSIFRDR)

- When accessing in longword size

Address(es): SSIE0.SSIFRDR 0008 A51Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

- When accessing in word size

Address(es): SSIE0.SSIFRDR.H 0008 A51Ch

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

- When accessing in byte size

Address(es): SSIE0.SSIFRDR.HH 0008 A51Ch

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

This register is a read-only register used to read a received data from the 32-stage and 4-byte-wide receive FIFO. Access to this register according to the access size listed in Table 45.6.

45.2.7 Audio Format Register (SSIOFR)

Address(es): SSIE0.SSIOFR 0008 A520h

	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
	—	—	—	—	—	—	BCKASTP	LRCONT	—	—	—	—	—	—	OMOD[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	OMOD[1:0]	Data Format Select*1	b1 b0 0 0: I²S, left-justified, or right-justified format 0 1: TDM format 1 0: Monaural format 1 1: Setting prohibited	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	LRCONT	LRCK Continuous Output Enable*2, *3	0: LRCK stops in the idle state. 1: Output of LRCK continues in the idle state.	R/W
b9	BCKASTP	BCK Continuous Output Disable*2, *3	0: BCK is always output. 1: BCK stops in idle state.	R/W
b31 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Rewrite these bits while SSIE is in idle state (SSISR.IIRQ flag = 1). If the value of these bits is changed while data transfer is in progress, the operation after the change is not guaranteed.
Note 2. This bit is valid only in master-mode. The setting is invalid in slave-mode.
Note 3. Do not set both of the BCKASTP and LRCONT bits to 1.

This register is used to set an audio format (which includes the settings of data format, LR clock continuous output mode, and BCK continuous output disable).

LRCONT Bit (LRCK Continuous Output Enable)

This bit enables or disables the output from SSILRCK pin when SSIE is in the idle state. This bit is effective only in master mode.

BCKASTP Bit (BCK Continuous Output Disable)

This bit enables or disables the output from SSIBCK pin when SSIE is in the idle state. This bit is effective only in master mode.

Set this bit to 0 to output BCK from the SSIBCK pin before starting data transfer. If this bit is set to 1 during data transfer, BCK output from the SSIBCK pin stops when the SSIE enters the idle state.

Set this bit after the data format to be used is specified. Do not rewrite this bit while the supply of MCK stops.

45.2.8 FIFO Status Control Register (SSISCR)

Address(es): SSIE0.SSISCR 0008 A524h

	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
	—	—	—	TDES[4:0]					—	—	—	RDFS[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	RDFS[4:0]	Receive Data Full Condition Setting*1	Specifies the condition for setting the RDF flag to 1. b4 b0 00000: When receive FIFO has unread received data. 00001: When receive FIFO has two or more unread received data. ... 11110: When receive FIFO has 31 or more unread received data. 11111: When receive FIFO is full.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12 to b8	TDES[4:0]	Transmit Data Empty Condition Setting*1	Specifies the condition for setting the TDE flag to 1. b12 b8 00000: When transmit FIFO has space available. 00001: When transmit FIFO has at least two stages available. ... 11110: When transmit FIFO has at least 31 stages available. 11111: When transmit FIFO is empty.	R/W
b31 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Rewrite these bits while SSIE is in idle state (SSISR.IIRQ flag = 1). If the value of these bits is changed while data transfer is in progress, the operation after the change is not guaranteed.

45.3 Operation

45.3.1 Data Formats

SSIE supports five data formats. Table 45.7 lists the supported data formats.

Table 45.7 Supported Data Formats and Bit Setting

Data Format	SSIOFR.OMOD[1:0] Bits	SSICR.DEL Bit	SSICR.LRCKP Bit	SSICR.SDTA Bit
I^2S format	00b	0	0	0
Left-justified format	00b	1	1	0
Right-justified format	00b	1	1	1
TDM format	01b	0 or 1	1	0 or 1
Monaural format	10b	0 or 1	1	0 or 1

The following describes the serial data structure shared by data formats.
A serial data structure is defined by the system word length (set in the SSICR.SWL[2:0] bits) and the data word length (set in the SSICR.DWL[2:0] bits). If the data word length is shorter than the system word length, padding bits are inserted (Figure 45.7).

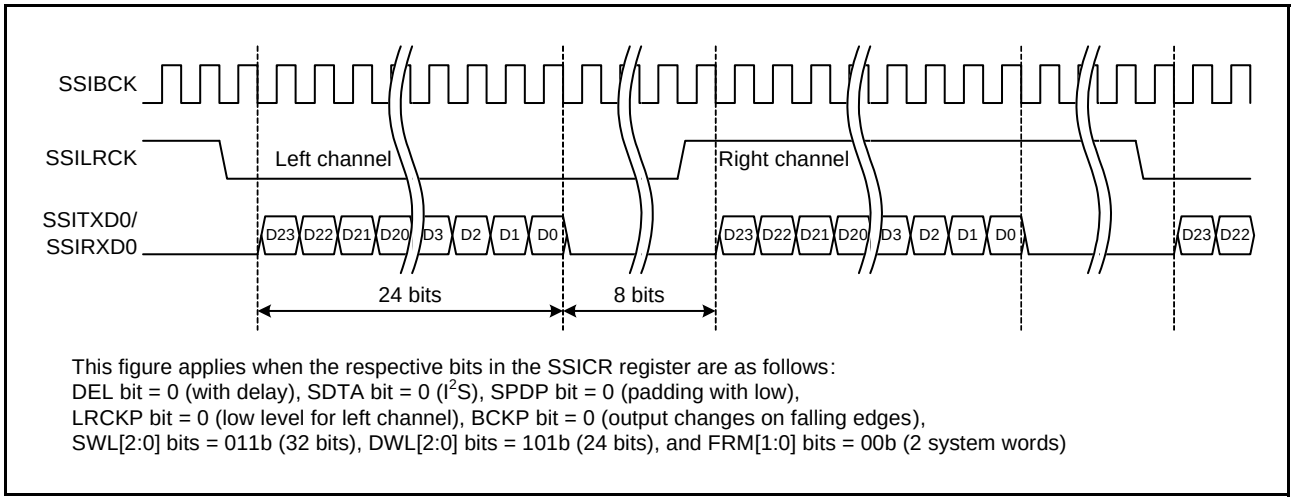


Figure 45.7 Example of Padding Bit Insertion (I^2S Format, System Word Length > Data Word Length)

The number of padding bits is calculated from the system word length minus the data word length.
Table 45.8 lists the number of padding bits to be inserted with each combination of system word length (SWL[2:0] bits) and data word length (DWL[2:0] bits).

Table 45.8 Number of Padding Bits

System Word Length (SSICR.SWL[2:0] Bits)	Data Word Length (SSICR.DWL[2:0] Bits)						
	8 (000b)	**16 (001b)**	**18 (010b)**	**20 (011b)**	**22 (100b)**	**24 (101b)**	**32 (110b)**
8 (000b)	0	Prohibited setting	Prohibited setting	Prohibited setting	Prohibited setting	Prohibited setting	Prohibited setting
16 (001b)	8	0	Prohibited setting	Prohibited setting	Prohibited setting	Prohibited setting	Prohibited setting
24 (010b)	16	8	6	4	2	0	Prohibited setting
32 (011b)	24	16	14	12	10	8	0
48 (100b)	40	32	30	28	26	24	16
64 (101b)	56	48	46	44	42	40	32
128 (110b)	120	112	110	108	106	104	96
256 (111b)	248	240	238	236	234	232	224

45.3.1.1 I²S Format

The I²S format configures a frame with two system words of left channel and right channel. The SSILRCK signals becomes low for left channel and high for right channel. The effective data are output one bit clock period after the SSILRCK changes and then the padding bits are output.
Figure 45.8 shows an example of I²S format.

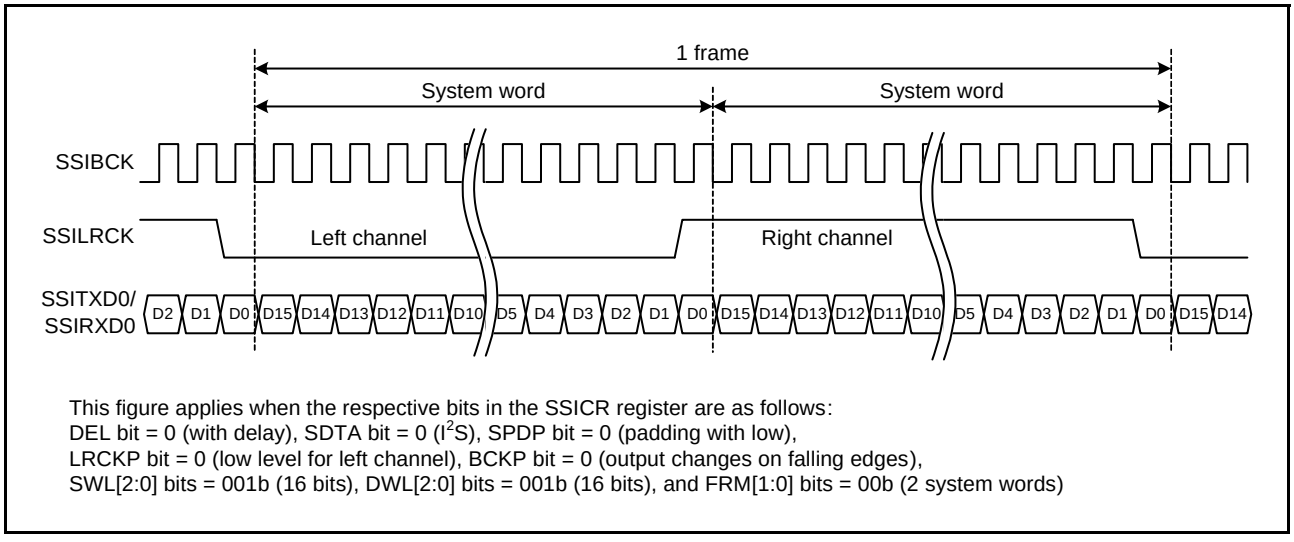
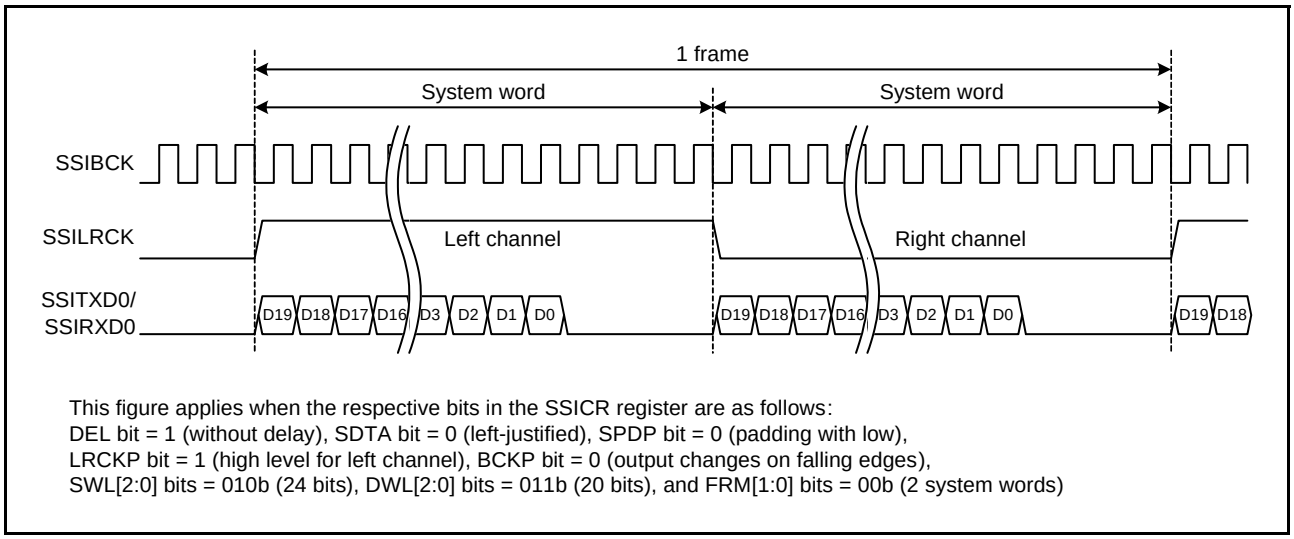


Figure 45.8 Example of I²S Format

45.3.1.2 Left-Justified Format

The left-justified format configures a frame with two system words of left channel and right channel. The SSILRCK signals becomes high for left channel and low for right channel. The effective data are output at the same time as the SSILRCK changes and then the padding bits are output.
Figure 45.9 shows an example of left-justified format.

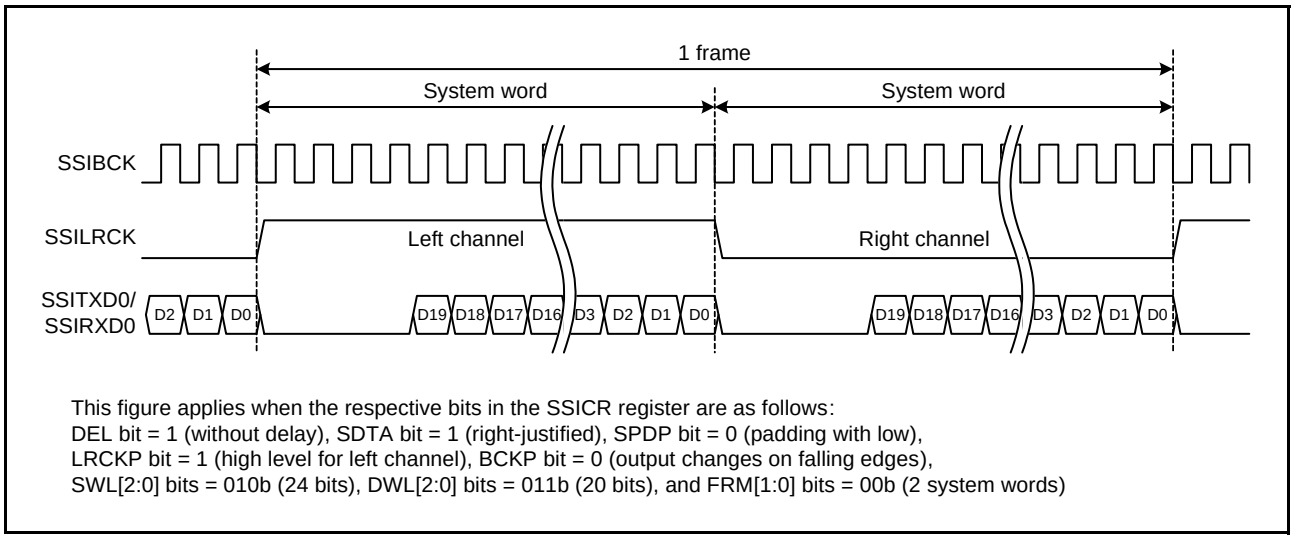


This figure applies when the respective bits in the SSICR register are as follows:
DEL bit = 1 (without delay), SDTA bit = 0 (left-justified), SPDP bit = 0 (padding with low),
LRCKP bit = 1 (high level for left channel), BCKP bit = 0 (output changes on falling edges),
SWL[2:0] bits = 010b (24 bits), DWL[2:0] bits = 011b (20 bits), and FRM[1:0] bits = 00b (2 system words)

Figure 45.9 Example of Left-Justified Format

45.3.1.3 Right-Justified Format

The right-justified format configures a frame with two system words of left channel and right channel. The SSILRCK signals becomes high for left channel and low for right channel. The padding bits are output at the same time as the SSILRCK changes and then the effective data are output.
Figure 45.10 shows an example of right-justified format.



This figure applies when the respective bits in the SSICR register are as follows:
DEL bit = 1 (without delay), SDTA bit = 1 (right-justified), SPDP bit = 0 (padding with low),
LRCKP bit = 1 (high level for left channel), BCKP bit = 0 (output changes on falling edges),
SWL[2:0] bits = 010b (24 bits), DWL[2:0] bits = 011b (20 bits), and FRM[1:0] bits = 00b (2 system words)

Figure 45.10 Example of Right-Justified Format

45.3.1.4 Monaural Format

The monaural format configures a frame with one system word. The SSILRCK signal is used as a transfer start trigger. Figure 45.11, Figure 45.12, and Figure 45.13 show examples of monaural formats with 0 as the value of the SSICR.DEL bit and no padding bits, with 0 as the value of the DEL bit and padding bits, and with 1 as the value of the DEL bit and no padding bits, respectively.

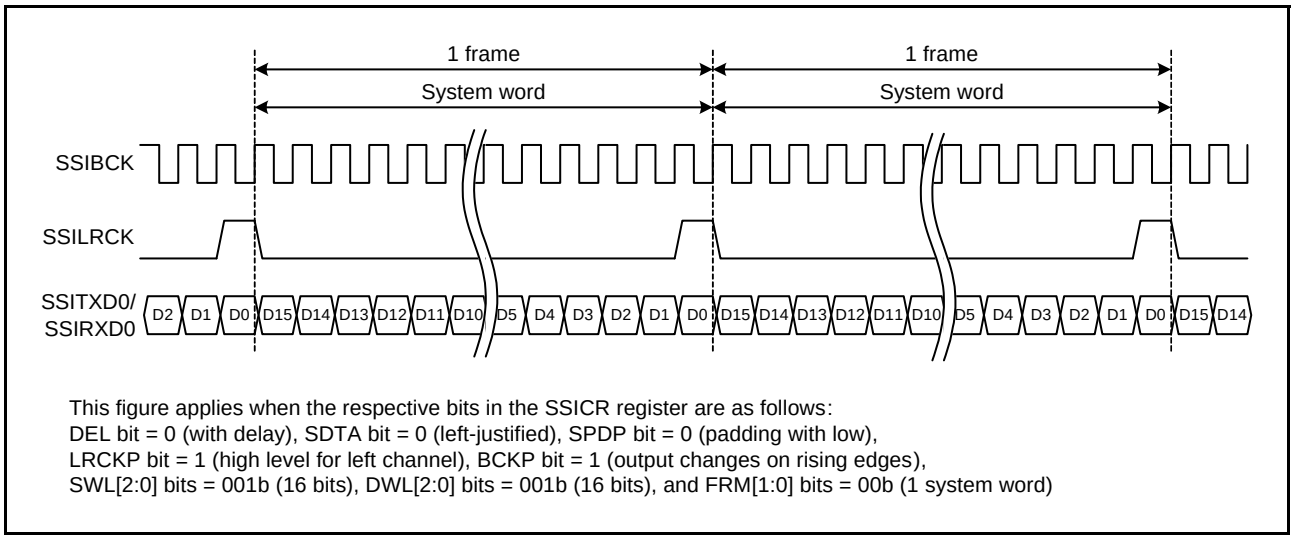


Figure 45.11 Example of Monaural Format (1): SSICR.DEL Bit = 0 and No Padding

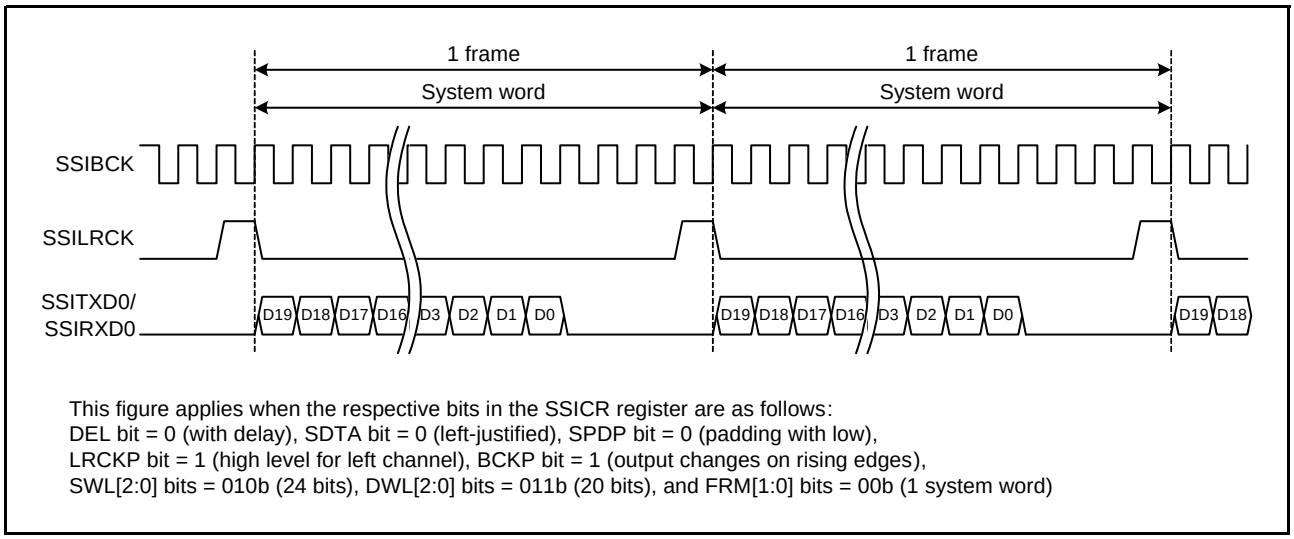


Figure 45.12 Example of Monaural Format (2): SSICR.DEL Bit = 0 and Padding Bits

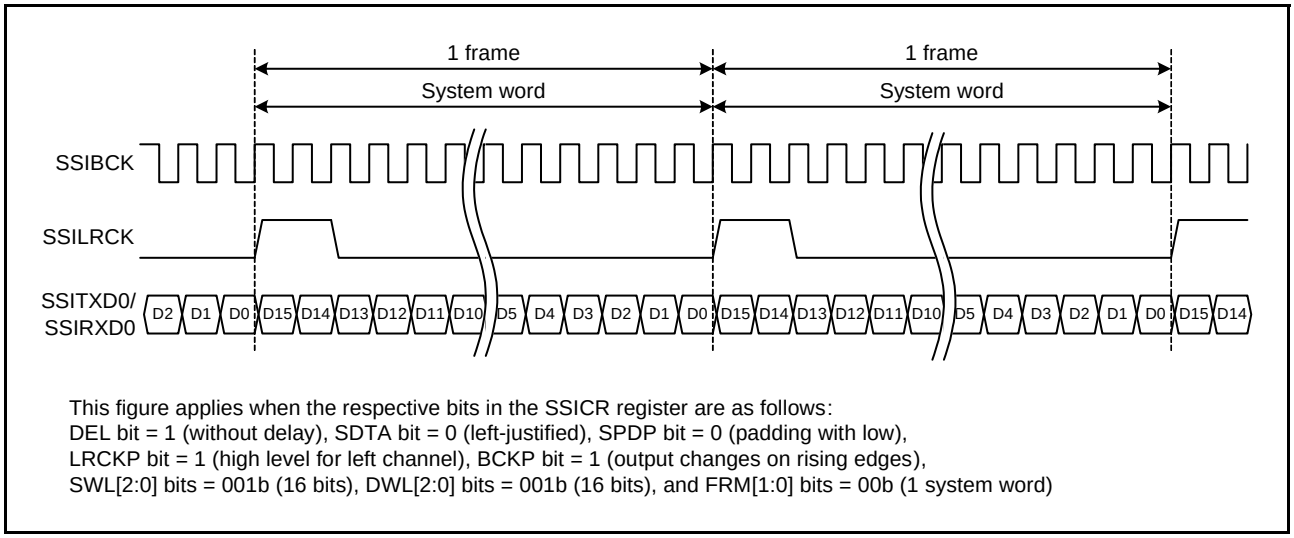


Figure 45.13 Example of Monaural Format (3): SSICR.DEL Bit = 1 and No Padding

45.3.1.5 TDM Format

The TDM format configures a frame with four to eight system words. The SSILRCK signals become high for the first system word and low for the other system words. The SSILRCK signal is defined as the synchronous pulse and its rising edge means a start of a frame.

Figure 45.14 and Figure 45.15 show examples of the TDM formats with 0 as the value of the SSICR.DEL bit and no padding bits, and with 1 as the value of the SSICR.DEL bit and padding bits, respectively.

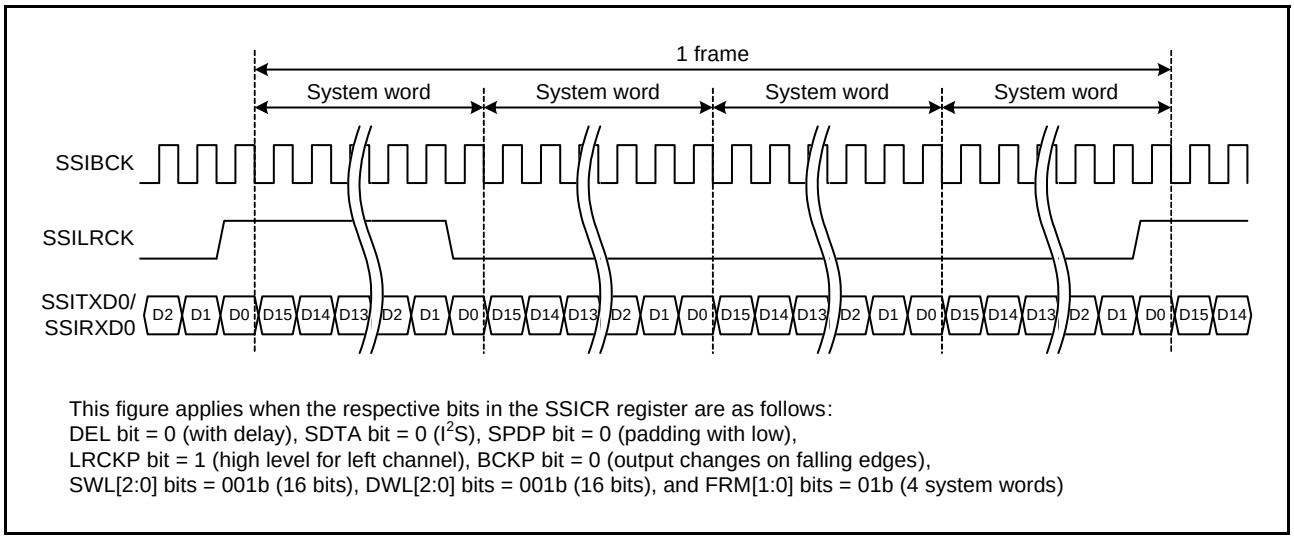
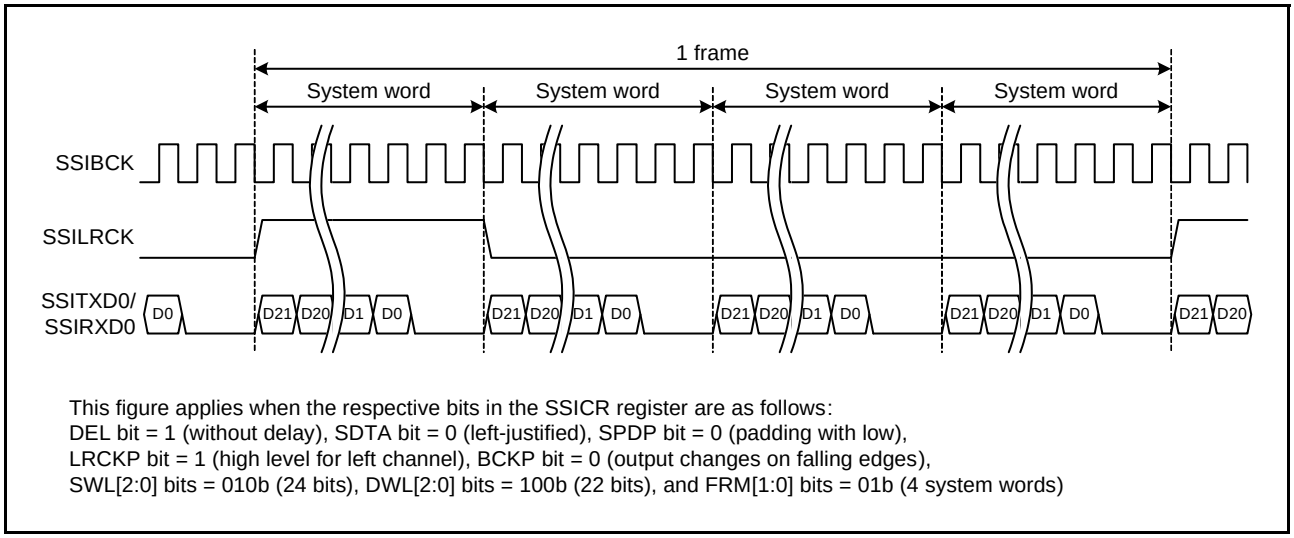


Figure 45.14 Example of TDM Format (1): SSICR.DEL Bit = 0 and No Padding



This figure applies when the respective bits in the SSICR register are as follows:
DEL bit = 1 (without delay), SDTA bit = 0 (left-justified), SPDP bit = 0 (padding with low),
LRCKP bit = 1 (high level for left channel), BCKP bit = 0 (output changes on falling edges),
SWL[2:0] bits = 010b (24 bits), DWL[2:0] bits = 100b (22 bits), and FRM[1:0] bits = 01b (4 system words)

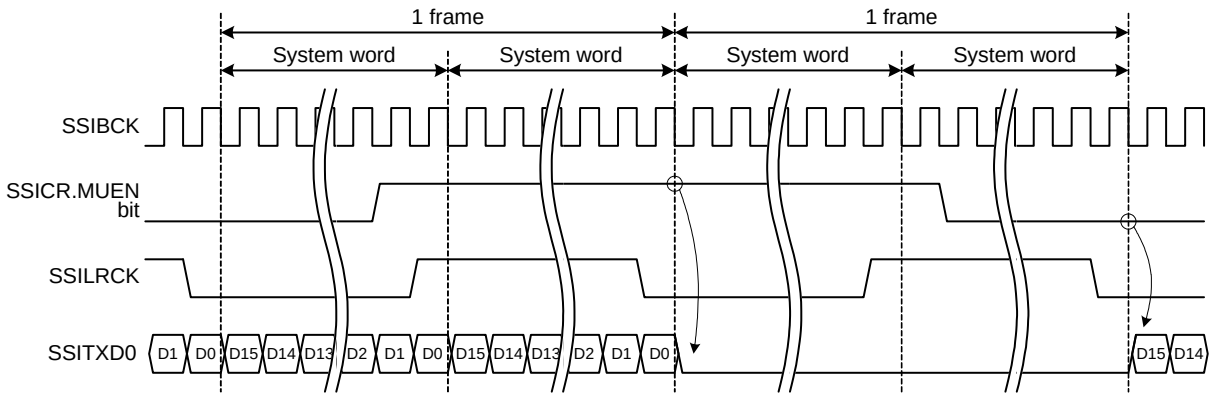
Figure 45.15 Example of TDM Format (2): SSICR.DEL Bit = 1 and Padding Bits

45.3.2 Output Control

SSIE has functions to control the outputs of data and clocks such as mute function, LRCK continuous output function, and BCK continuous output disable function.

45.3.2.1 Mute Function

The output from the SSITXD0 pin can be fixed to the low level by using the mute function. If the SSICR.MUEN bit is changed during a frame, the setting takes effect at the next frame boundary. Even while the MUEN bit is set to 1, status flags are updated and interrupt request signals are generated.
Figure 45.16 shows an example of the mute operation.



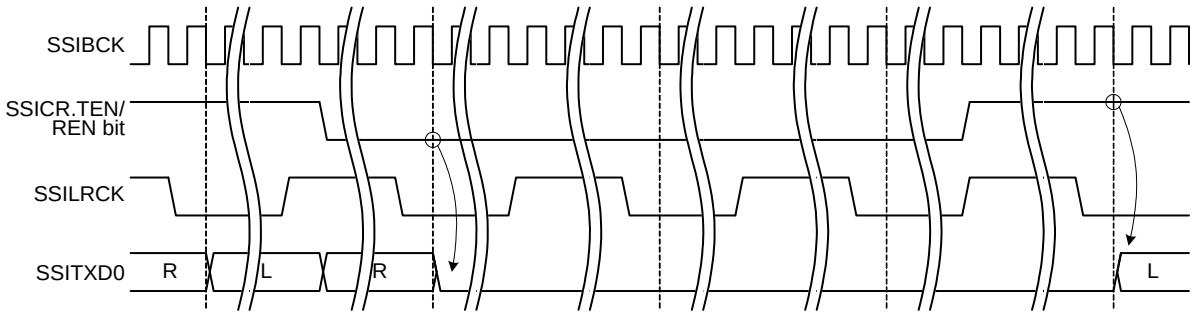
This figure applies when the respective bits in the SSICR register are as follows:
DEL bit = 0 (with delay), SDTA bit = 0 (I^2S), SPDP bit = 0 (padding with low),
LRCKP bit = 0 (low level for left channel), BCKP bit = 0 (output changes on falling edges),
SWL[2:0] bits = 001b (16 bits), DWL[2:0] bits = 001b (16 bits), and FRM[1:0] bits = 00b (2 system words)

Figure 45.16 Example of Mute Operation

45.3.2.2 LRCK Continuous Output Function

LRCK can be output from the SSILRCK pin even in the idle state by using the LRCK continuous output function. It is controlled by the SSIOFR.LRCONT bit.

Figure 45.17 shows an example of the operation with LRCK continuous output function.



This figure applies when the respective bits in the SSICR register are as follows:
DEL bit = 0 (with delay), SDTA bit = 0 (I^2S), SPDP bit = 0 (padding with low),
LRCKP bit = 0 (low level for left channel), BCKP bit = 0 (output changes on falling edges),
and FRM[1:0] bits = 00b (2 system words)

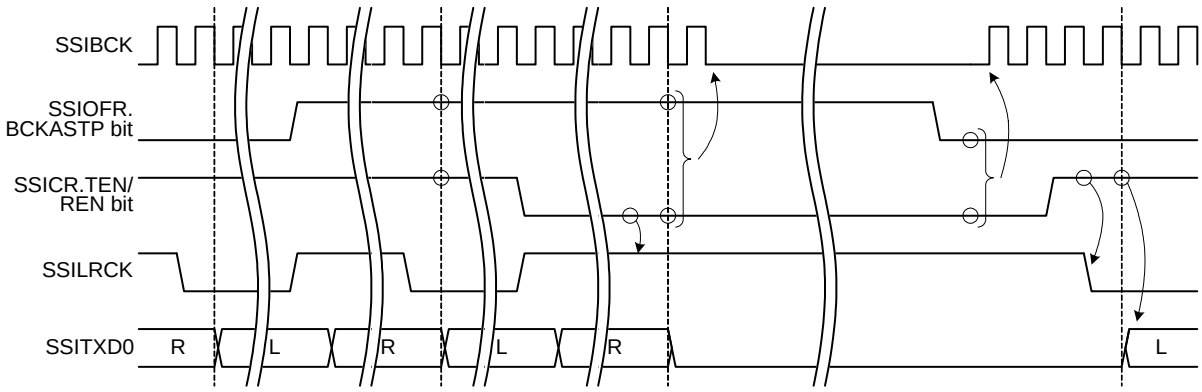
Figure 45.17 Example of Operation with LRCK Continuous Output Function

45.3.2.3 BCK Continuous Output Disable Function

BCK output from the SSIBCK pin can be stopped in the idle state by using the BCK continuous output disable function. It is controlled by the SSIOFR.BCKASTP bit. When the BCKASTP bit is set to 1, BCK stops in the idle state.

Set the BCKASTP bit to 0 and then start data transfer. If this bit is set to 1 during data transfer, BCK stops when SSIE enters the idle state. To resume the data transfer, set the BCKSTP bit to 0 to output BCK and then set the SSICR.TEN/REN bit to 1.

Figure 45.18 shows an example of the operation with BCK continuous output disable function.



This figure applies when the SSIOFR.LRCONT bit is 0 and the respective bits in the SSICR register are as follows:
DEL bit = 0 (with delay), SDTA bit = 0 (I^2S), SPDP bit = 0 (padding with low),
LRCKP bit = 0 (low level for left channel), BCKP bit = 0 (output changes on falling edges),
and FRM[1:0] bits = 00b (2 system words)

Figure 45.18 Example of Operation with BCK Continuous Output Disable Function

45.3.3 Transfer Modes

SSIE supports the transfer modes listed in Table 45.9. Table 45.10 lists the control bits valid in each transfer mode.

Table 45.9 Transfer Modes

Transfer Mode	SSICR.MST Bit	SSICR.REN Bit	SSICR.TEN Bit
Slave-mode transmission	0	0	1
Slave-mode reception	0	1	0
Slave-mode transception	0	1	1
Master-mode transmission	1	0	1
Master-mode reception	1	1	0
Master-mode transception	1	1	1

Table 45.10 Control Bits Valid in Each Transfer Mode*1

Control Bit		Transfer Mode					
Register	Bit	Slave-mode Reception	Slave-mode Transmission	Slave-mode Transception	Master-mode Reception	Master-mode Transmission	Master-mode Transception
SSICR	CKDV[3:0]	Ineffective	Ineffective	Ineffective	Effective	Effective	Effective
	MUEN	Ineffective	Effective	Effective	Ineffective	Effective	Effective
SSIFCR	AUCKEN	Ineffective	Ineffective	Ineffective	Effective	Effective	Effective
	TIE	Ineffective	Effective	Effective	Ineffective	Effective	Effective
	RIE	Effective	Ineffective	Effective	Effective	Ineffective	Effective
	TFRST	Ineffective	Effective	Effective	Ineffective	Effective	Effective
	RFRST	Effective	Ineffective	Effective	Effective	Ineffective	Effective
SSIOFR	BCKASTP	Ineffective	Ineffective	Ineffective	Effective	Effective	Effective
	LRCONT	Ineffective	Ineffective	Ineffective	Effective	Effective	Effective
	OMOD[1:0]	Effective	Effective	Effective	Effective	Effective	Effective
SSISCR	TDES[4:0]	Ineffective	Effective	Effective	Ineffective	Effective	Effective
	RDFS[4:0]	Effective	Ineffective	Effective	Effective	Ineffective	Effective

Note 1. “Ineffective” means that the setting of the bits does not affect operation.

45.3.3.1 Slave Mode

The SSIE operates in slave mode when the SSICR.MST bit is 0. The SSIBCK and SSILRCK signals must be supplied from an external device.

45.3.3.2 Master Mode

The SSIE operates in master mode when the SSICR.MST bit is 1. The SSIBCK and SSILRCK signals are internally generated from the audio clock (AUDIO_CLK).

45.3.3.3 Transmission

The SSIE operates as a transmitter when the SSICR.TEN bit is 1 and the SSICR.REN bit is 0. Receive circuits do not operate.

45.3.3.4 Reception

The SSIE operates as a receiver when the SSICR.TEN bit is 0 and the SSICR.REN bit is 1. Transmit circuits do not

operate.

45.3.3.5 Transception

The SSIE operates as a transceiver when both of the SSICR.TEN and SSICR.REN bits are 1.

45.3.4 State Transition

SSIE has the two main operation states: idle state and transfer state. The SSICR.TEN and SSICR.REN bits switch between these states. The SSISR.IIRQ flag can be used to confirm in which state SSIE is.
Figure 45.19 shows the state transition of the SSIE.

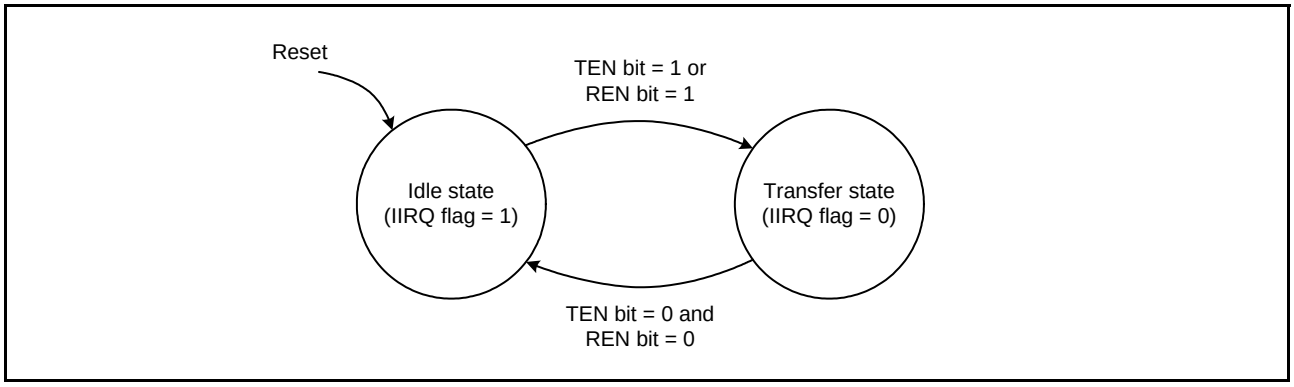


Figure 45.19 State Transition of the SSIE

45.3.4.1 Idle State

In the idle state, SSIE stops data transfer. When both or either of the SSICR.TEN and SSICR.REN bits are/is set to 1, the SSIE enters the transfer state.
Outputs of the BCK and LRCK in master mode can be controlled by the SSIOFR.BCKASTP and SSIOFR.LRCONT bits, respectively. For details, refer to Table 45.11.

Table 45.11 Output of Clock/Data in the Idle State

SSICR.MST Bit	SSIOFR.BCKASTP Bit	SSIOFR.LRCONT Bit	SSIBCK	SSILRCK	SSITXD0
0	Invalid	Invalid	— (input)	— (input)	Stop
1	0	0	Supply	Stop	Stop
1	0	1	Supply	Supply	Stop
1	1	0	Stop	Stop	Stop
1	1	1	Stop	Supply	Stop

45.3.4.2 Transfer States

In this state, SSIE continues data transfer. When the SSICR.TEN and SSICR.REN bit is set to 0, SSIE enters the idle state.

For details, refer to Figure 45.20.

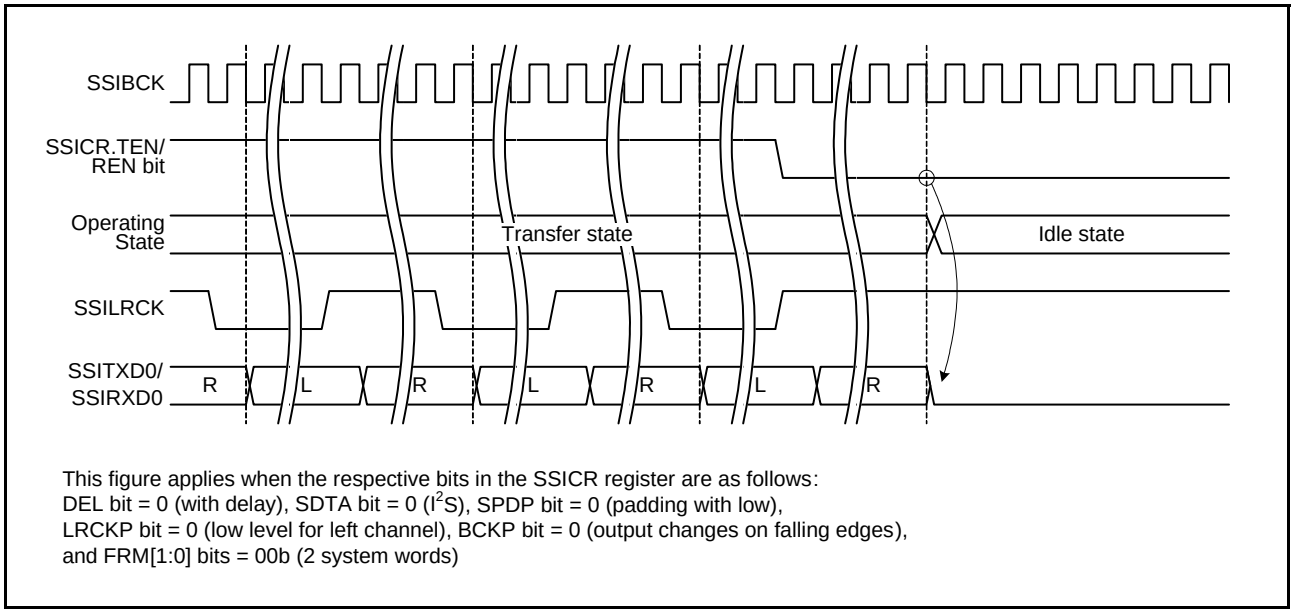


Figure 45.20 Transfer State and Transition to Idle State

45.3.5 SSIE Initialization

Figure 45.21 shows an example of the flow to initialize SSIE.

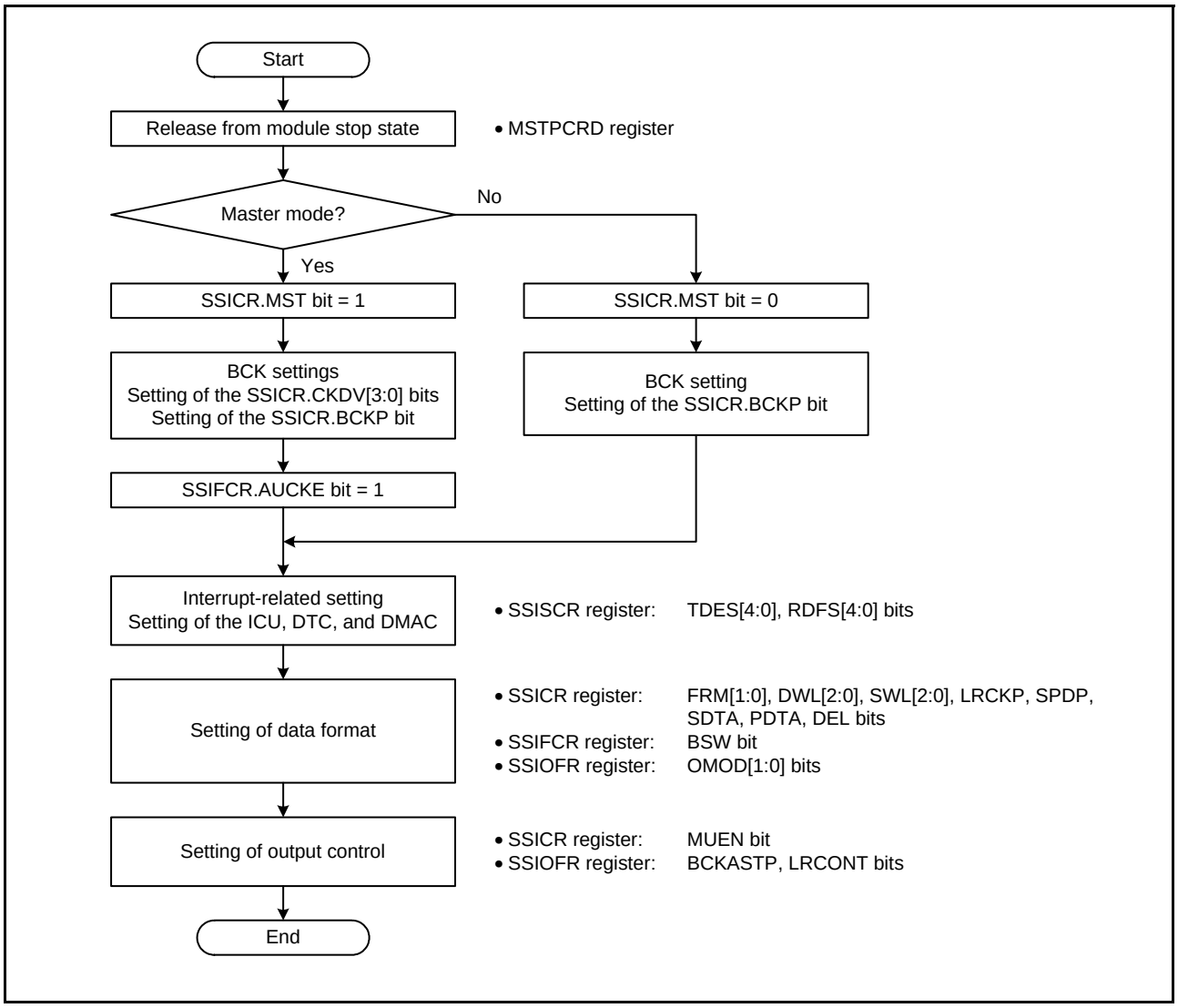
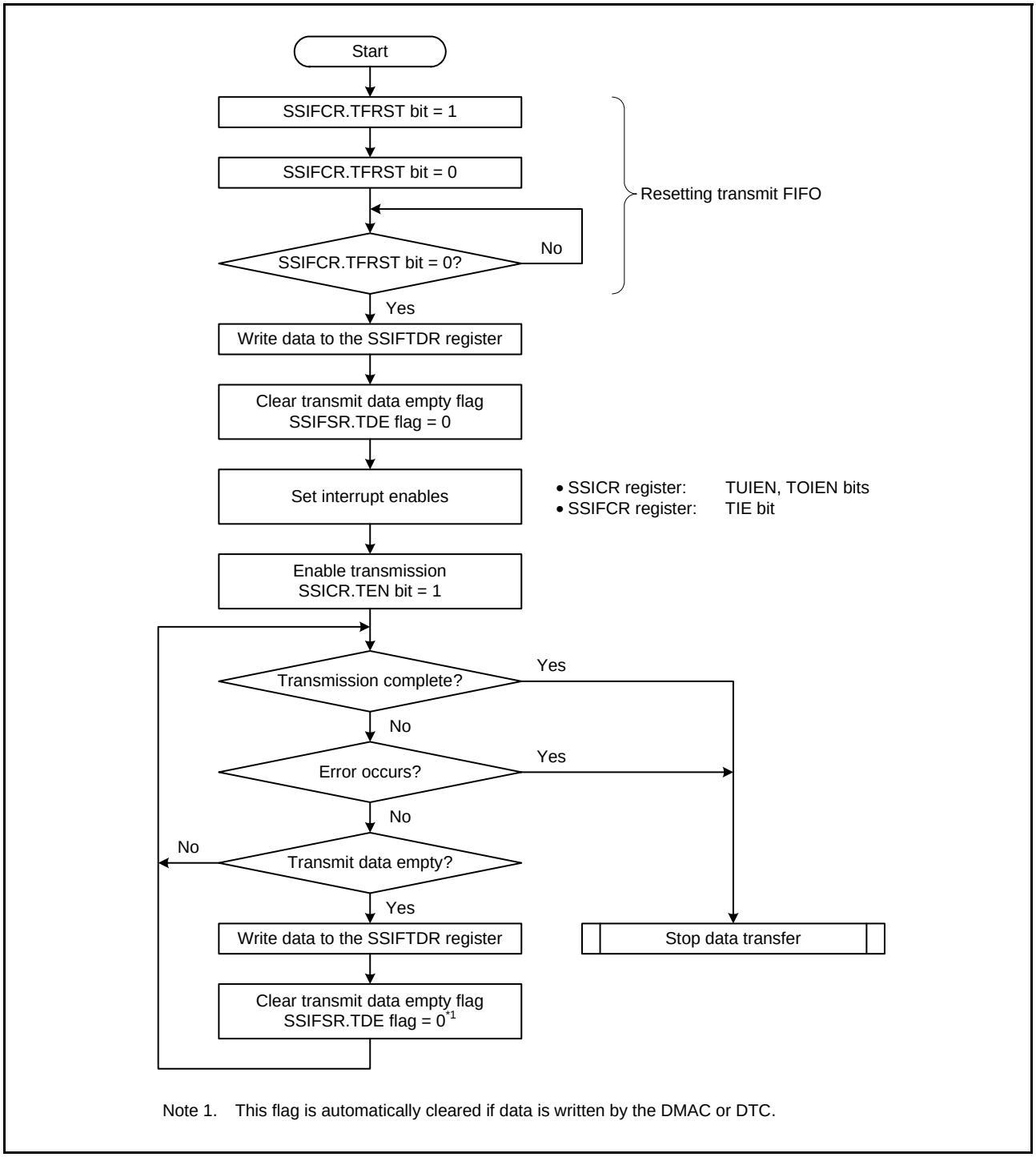


Figure 45.21 Example Flowchart of SSIE Initialization

45.3.6 Transmission

Figure 45.22 shows an example of the flow to transmit data.
When the SSICR.TEN bit is set to 1 after setting the data for one frame in the SSIFTDR register, the SSIE starts transmission. When the number of spaces available in the transmit FIFO reaches the value specified by the SSISCR.TDES[4:0] bits while the SSIFCR.TIE bit is 1, a transmit data empty interrupt occurs. The number of data to be written to the SSIFTDR register should be determined in accordance with the space available in the transmit FIFO specified by the TDES[4:0] bits.

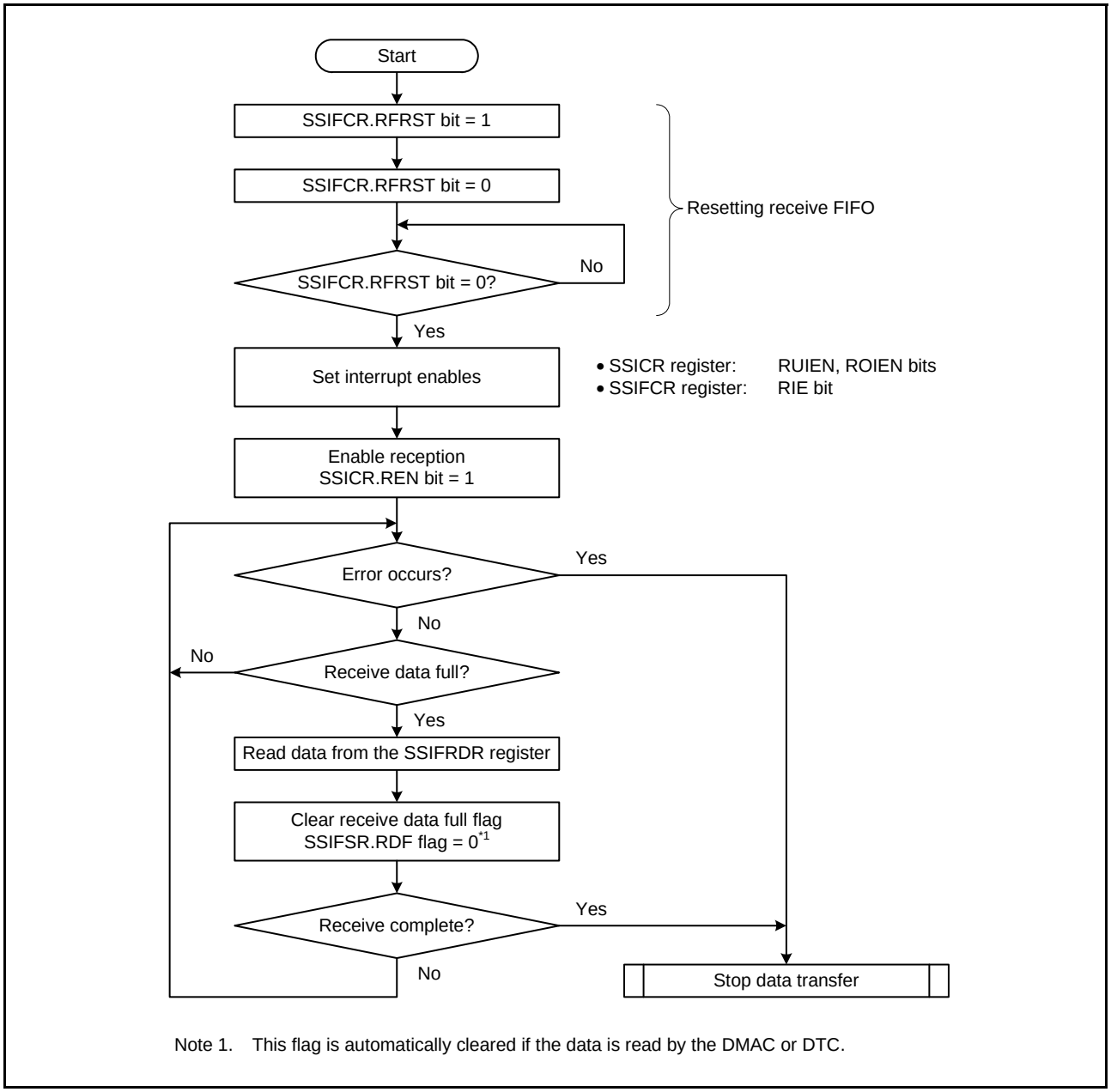


Note 1. This flag is automatically cleared if data is written by the DMAC or DTC.

Figure 45.22 Example Flowchart of Transmission

45.3.7 Reception

Figure 45.23 shows an example of the flow to receive data.
When the SSICR.REN bit is set to 1, SSIE starts reception. When the number of data stored in the receive FIFO reaches to the value specified by the SSISCR.RDFS[4:0] bits while the SSIFCR.RIE bit is 1, a receive data full interrupt occurs. The number of data to be read from the SSIFRDR register should be determined in accordance with the number of data stored in the receive FIFO specified by the RDFS[4:0] bits.



Note 1. This flag is automatically cleared if the data is read by the DMAC or DTC.

Figure 45.23 Example Flowchart of Reception

45.3.8 Stop Data Transfer

Figure 45.24 shows an example of the flow to stop data transfer.

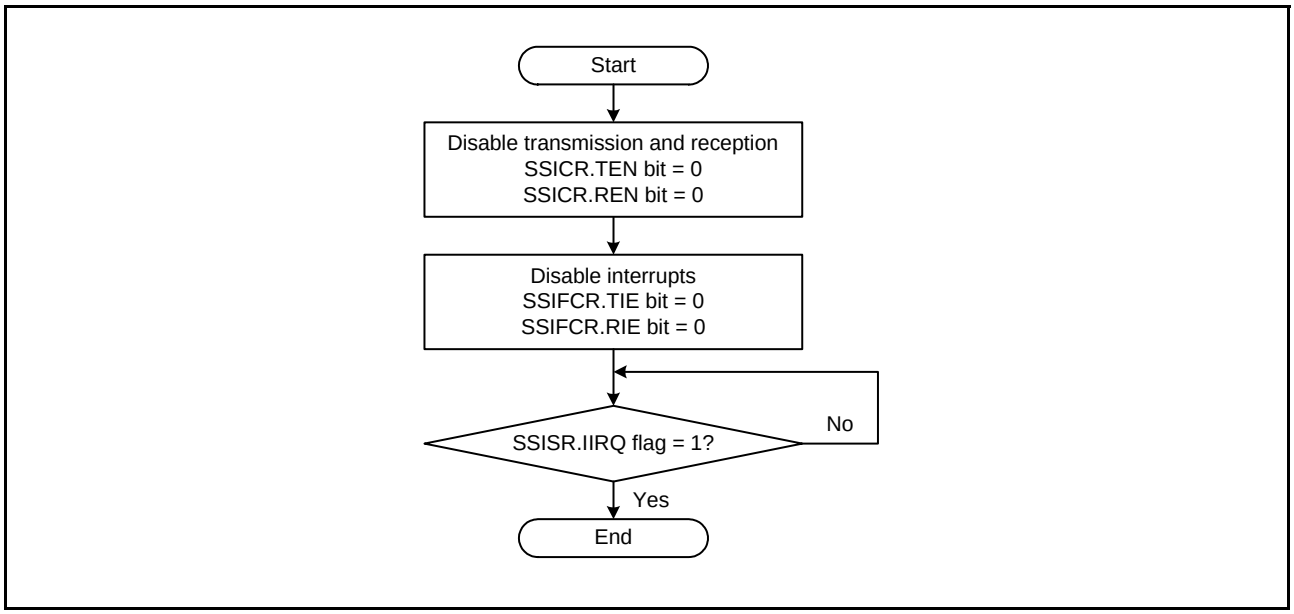


Figure 45.24 Example Flowchart of Stopping Data Transfer

To resume data transfer, conform with the example flowcharts of transmission and reception in Figure 45.22 and Figure 45.23, respectively. When changing the settings of clocks or switching a transfer mode between master and slave, start from initialization.

45.3.9 Error Processing

SSIE has the following four errors.

- Transmit underflow
- Transmit overflow
- Receive underflow
- Receive overflow

If an error has occurred, follow the procedure to stop data transfer shown in Figure 45.24 and then clear the corresponding flag. When an error has occurred while the corresponding interrupt enable bit in the SSICR register is 1, an interrupt is generated.

(1) Transmit Underflow

When a data transfer from the transmit FIFO to transmit shift register has performed while the transmit FIFO is empty, a transmit underflow is informed. If a transmit underflow has occurred, the SSITXD0 pin is driven low.

(2) Transmit Overflow

When a data is written to the SSIFTDR register while the transmit FIFO is full, a transmit overflow is informed. The data to be written is discarded.

(3) Receive Underflow

When a data is read from the SSIFRDR register while the receive FIFO is empty, a receive underflow is informed. The read data is undefined.

(4) Receive Overflow

When a data transfer from receive shift register to the receive FIFO has performed while the receive FIFO is full, a receive overflow is informed. The data to be transferred is discarded and is not stored in the receive FIFO.

45.4 Interrupts

Table 45.12 lists the interrupt sources. Figure 45.25 shows an block diagram for interrupts.

Table 45.12 SSIE Interrupt Sources

Channel	Name	Interrupt Source	Interrupt Request Flag	Interrupt Enable Bit	DMAC/DTC Triggerable
SSIE0	SSIF0	Transmit underflow interrupt Transmit overflow interrupt Receive underflow interrupt Receive overflow interrupt Idle state interrupt	SSISR.TUIRQ SSISR.TOIRQ SSISR.RUIRQ SSISR.ROIRQ SSISR.IIRQ	SSICR.TUIEN SSICR.TOIEN SSICR.RUIEN SSICR.ROIEN SSICR.IIEN	No
	SSIRXI0	Receive data full interrupt	SSIFSR.RDF	SSIFCR.RIE	Yes
	SSITXI0	Transmit data empty interrupt	SSIFSR.TDE	SSIFCR.TIE	Yes

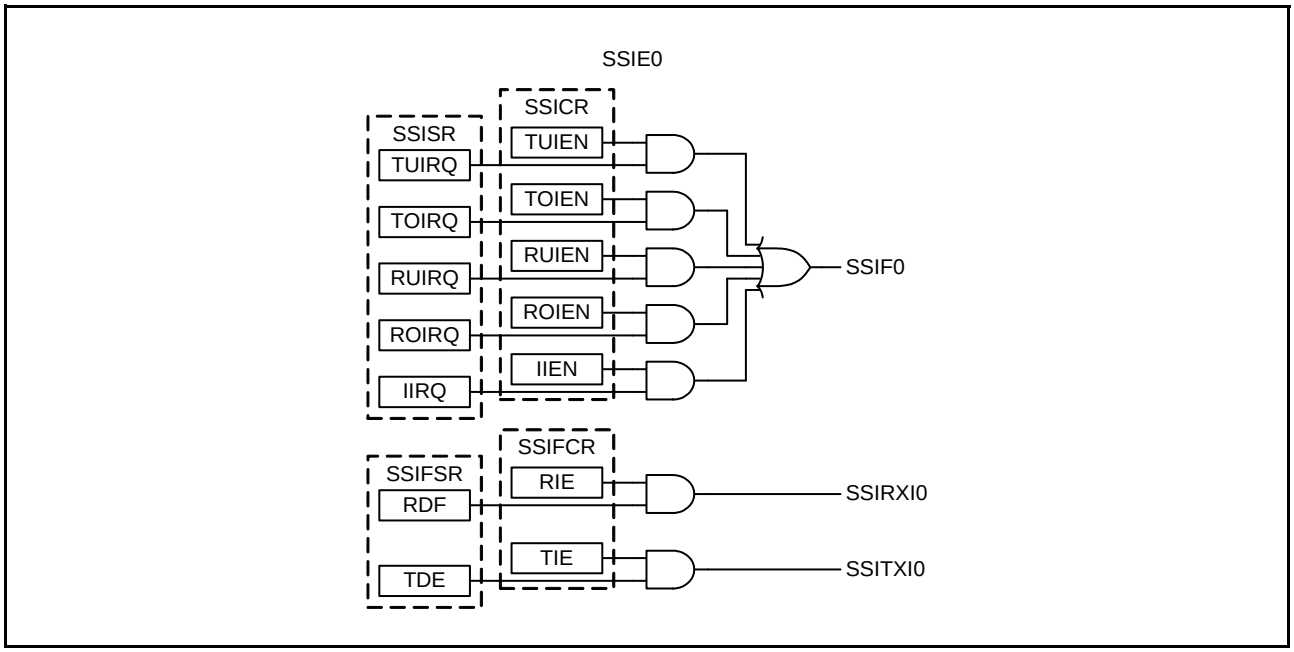


Figure 45.25 Interrupt System Block Diagram

45.4.1 SSIF0 Interrupt

The SSIF0 interrupt combines five interrupt sources. Each interrupt source has a status flag and an interrupt enable bit, so the interrupt can be enabled or disabled individually.

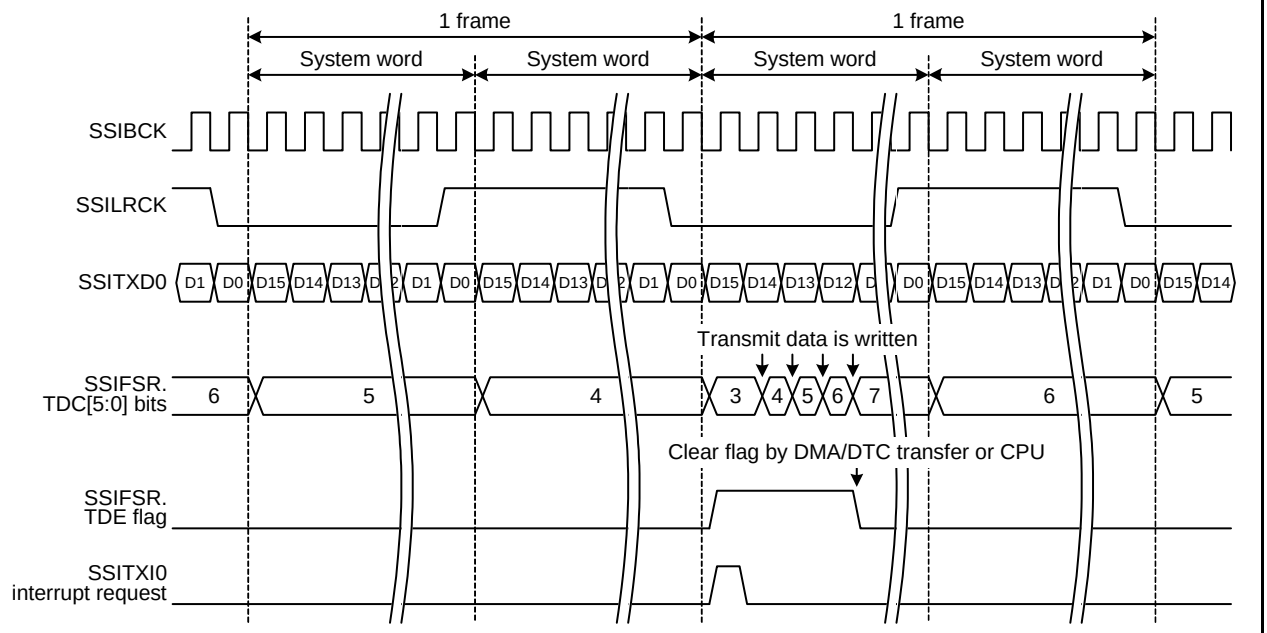
If an SSIF0 interrupt occurs, read the SSISR register in the interrupt handling routine to check the source of the interrupt request, and then perform proper operation. To clear the interrupt request, set the corresponding interrupt enable bit to 0 or clear the interrupt request flag.

45.4.2 SSITXI0 Interrupt

The SSITXI0 interrupt (channel 0 transmit data empty interrupt) is an edge detection interrupt. This interrupt request is generated when either of the following conditions is satisfied.

- The SSIFSR.TDE flag has changed from 0 to 1 while the SSIFCR.TIE bit is 1.
- The SSIFCR.TIE bit has been changed from 0 to 1 while the SSIFSR.TDE flag is 1.

Figure 45.26 shows a timing chart when an interrupt request is generated when the TDE flag changes.



This figure applies when the SSISCR.TDES[4:0] bits are 11100b (at least 29) and when the respective bits in the SSICR register are as follows:
DEL bit = 0 (with delay), SDTA bit = 0 (I^2S), SPDP bit = 0 (padding with low),
LRCKP bit = 0 (low level for left channel), BCKP bit = 0 (output changes on falling edges),
SWL[2:0] bits = 001b (16 bits), DWL[2:0] bits = 001b (16 bits), and FRM[1:0] bits = 00b (2 system words)

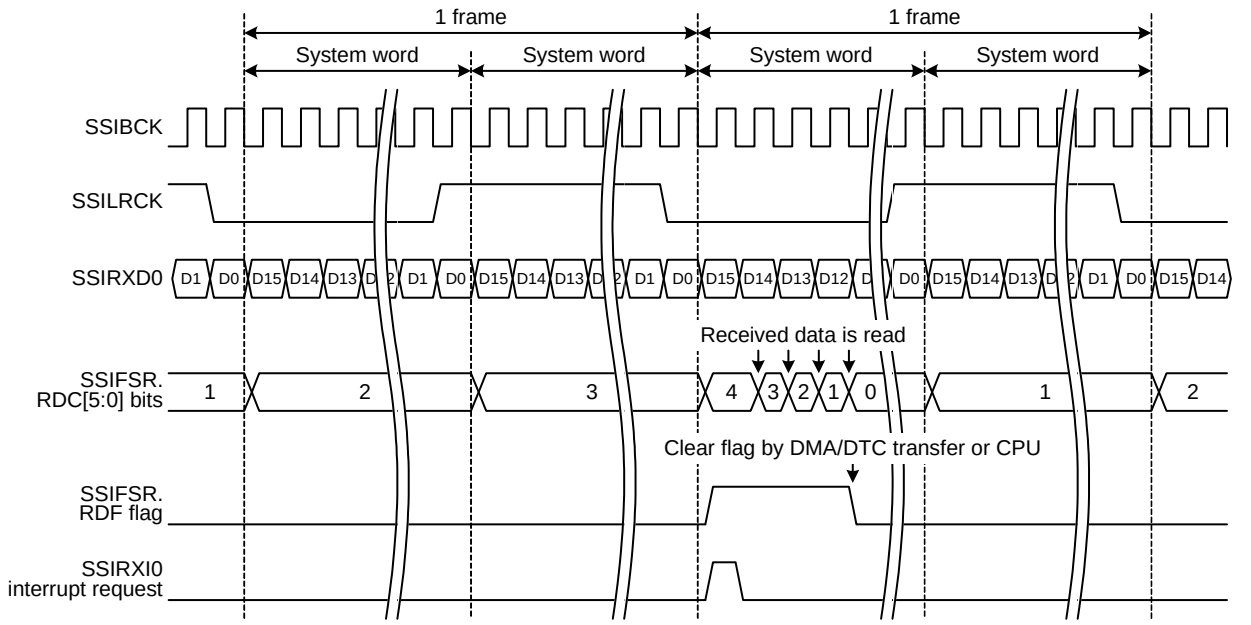
Figure 45.26 SSITXI0 Interrupt Timing Chart

45.4.3 SSIRXI0 Interrupt

The SSIRXI0 interrupt (channel 0 receive data full interrupt) is an edge detection interrupt. This interrupt request is generated when either of the following conditions is satisfied.

- The SSIFSR.RDF flag has changed from 0 to 1 while the SSIFCR.RIE bit is 1.
- The SSIFCR.RIE bit has been changed from 0 to 1 while the SSIFSR.RDF flag is 1.

Figure 45.27 shows a timing chart when an interrupt request is generated when the RDF flag changes.



This figure applies when the SSISCR.RDFS[4:0] bits are 00011b (at least 4) and when the respective bits in the SSICR register are as follows:
DEL bit = 0 (with delay), SDTA bit = 0 (I^2S), SPDP bit = 0 (padding with low),
LRCKP bit = 0 (low level for left channel), BCKP bit = 0 (output changes on falling edges),
SWL[2:0] bits = 001b (16 bits), DWL[2:0] bits = 001b (16 bits), and FRM[1:0] bits = 00b (2 system words)

Figure 45.27 SSIRXI0 Interrupt Timing Chart

45.5 Notes

45.5.1 Setting the Module Stop Function

Operation of the SSIE can be enabled or disabled by setting a bit in the module stop control register D (MSTPCRD). The SSIE is initially disabled after a reset. Registers in the SSIE only become accessible after it has been released from the module stop state. For details, refer to section 11, Low Power Consumption.

45.5.2 Note on LRCK in Slave Mode

The SSIE uses the SSILRCK signal as a start trigger in slave mode. Neither valid edges after the start trigger nor signal levels are used, so if a bit slippage occurs by a noise, the bit slippage cannot be repaired even at a frame boundary.

46. Remote Control Signal Receiver (REMCa)

This MCU has a remote control signal receiver (REMC0). The REMC can receive data by checking the width and period of an external pulse input signal.

46.1 Overview

Table 46.1 lists the REMC specifications. Figure 46.1 shows a block diagram of the REMC.

Table 46.1 REMC Specifications

Item	Description
External pulse input	PMC0
Operating clock sources*1	• Sub-clock • TMR compare match output (TMO0) • PCLKB
Detection patterns	• Header pattern • Data ‘0’ pattern • Data ‘1’ pattern • Special data pattern
Receive buffer	8 bytes (64 bits)
Interrupt request signal	REMCI0
Interrupt request source	• Compare match (Compare bit count: 1 to 16 bits) • Receive error • Data reception complete • Receive buffer full • Header pattern match • Data ‘0’ pattern or data ‘1’ pattern match • Special data pattern match
Interrupt mode	Either of the following two interrupt modes can be selected for the four interrupt sources of compare match, data reception complete, header pattern match, and special data pattern match. • Normal interrupt mode An interrupt request is generated when any of the interrupt request generation condition is met. • Sequential interrupt mode An interrupt request is generated when the interrupt request generation conditions are met for all the enabled interrupt request sources.
Selectable functions	• Input signal inversion • Digital filter (matching three or two times)*2 • Pattern end setting
Low power consumption	• Module stop state can be set. • Signal reception during low power consumption state and recovery from low power consumption state in response to the REMC interrupt request are available.

Note 1. Satisfy the frequency of the peripheral module clock (PCLKB) > the frequency of the REMC operating clock.
Note 2. The sampling clock of the digital filter is an operating clock selected by the REMCON1.CSRC[3:0] bits.

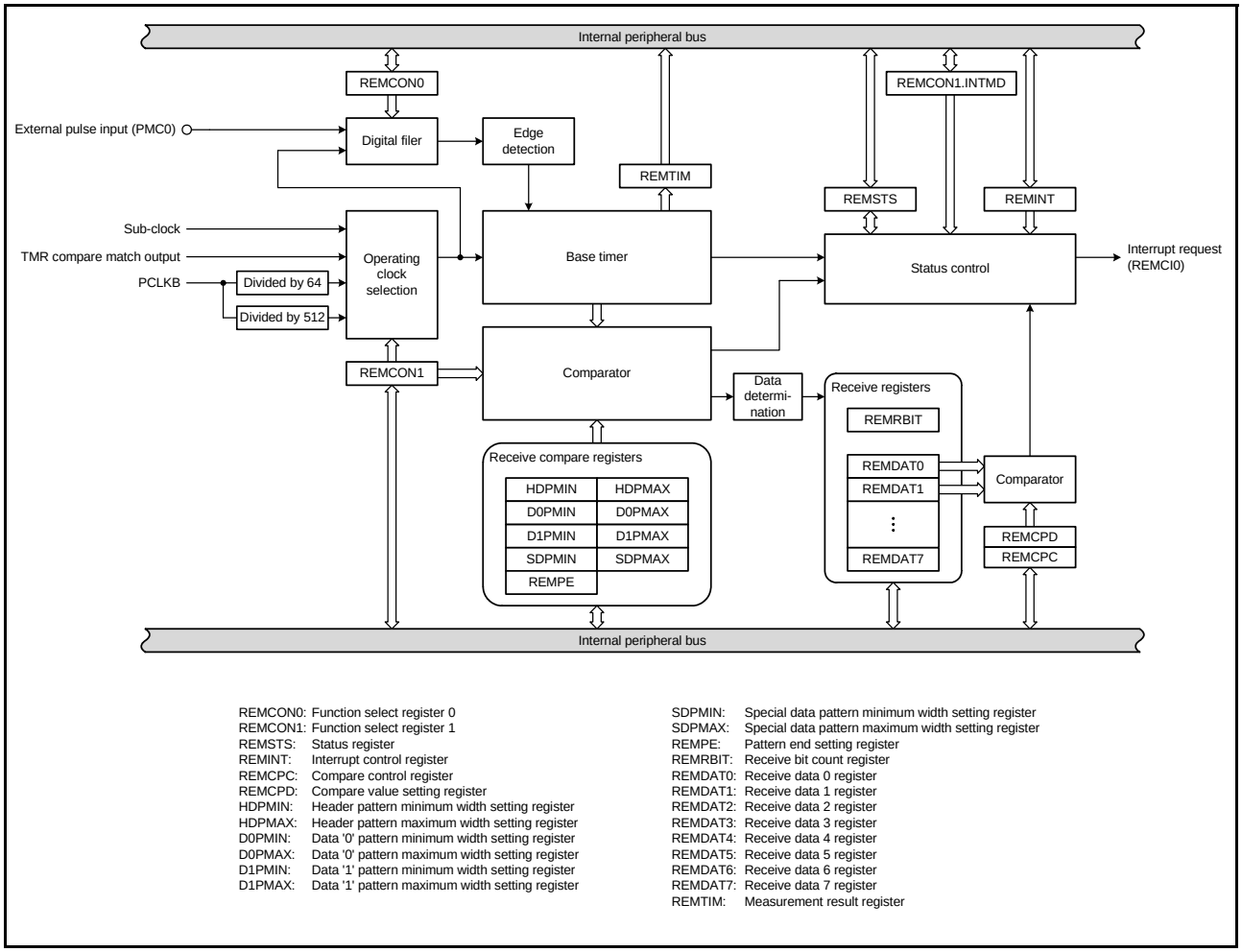


Figure 46.1 REMC Block Diagram

Table 46.2 lists the input pins used for the REMC.

Table 46.2 REMC Pin Configuration

Channel	Pin Name	I/O	Function
REMC0	PMC0	Input	External pulse signal input

46.2 Registers

46.2.1 Function Select Register 0 (REMCON0)

Address(es): REMC0.REMCON0 000A 0B00h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	FILSEL	—	EC	INFLG	FIL	INV	ENFLG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ENFLG	Remote Control Status Flag*1	0: Stopped 1: Operating	R
b1	INV	Input Signal Inversion*2	0: Not inverted 1: Inverted	R/W
b2	FIL	Digital Filter Enable/Disable Setting*2	0: Disables the digital filter for matching three or two times. 1: Enables the digital filter for matching three or two times.	R/W
b3	INFLG	Input Signal Flag*1	0: The level of the internal input signal of the remote control signal receiver is low. 1: The level of the internal input signal of the remote control signal receiver is high.	R
b4	EC	Receive Error Capture Operation Select*2	0: Captures the data after an error pattern is received. 1: Does not capture the data after an error pattern is received.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	FILSEL	Digital Filter Function Select*2	0: Digital filter for matching three times 1: Digital filter for matching two times	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. These flags become 0 when the REMCON1.EN bit is set to 0.
Note 2. These bits can be rewritten when the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC is stopped).

ENFLG Flag (Remote Control Status Flag)

This flag can be used to confirm whether the remote control signal receiver is stopped or operating.
This flag changes after zero to one clock when a value is written to the REMCON1.EN bit.

FIL Bit (Digital Filter Enable/Disable Setting)

This bit enables or disables the digital filter.

INFLG Flag (Input Signal Flag)

This flag can be used to confirm the level of the internal input signal of the remote control signal receiver. The confirmed level is the result set by the INV and FIL bits.

EC Bit (Receive Error Capture Operation Select)

This bit can be used to set capture operation to the REMRBIT and REMDATj registers (j = 0 to 7) after an error pattern is received.

46.2.2 Function Select Register 1 (REMCON1)

Address(es): REMC0.REMCON1 000A 0B01h

	b7	b6	b5	b4	b3	b2	b1	b0
	INTMD	CSRC[3:0]				EN	TYP[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TYP[1:0]	Receive Mode Select*1	These bits can be used to select the format for capturing the remote control signal waveform. b1 b0 0 0: Format A shown in section 46.3.3, Pattern Setting. 0 1: Format B shown in section 46.3.3, Pattern Setting. 1 0: Format C shown in section 46.3.3, Pattern Setting. 1 1: Setting prohibited	R/W
b2	EN	Remote Control	0: Operation disabled 1: Operation enabled	R/W
b6 to b3	CSRC[3:0]	Operating Clock Select*2	b6 b3 x 0 1 0: TMR compare match output x 1 0 0: Sub-clock 0 1 1 0: PCLKB/64 1 1 1 0: PCLKB/512 Settings other than those listed above are prohibited.	R/W
b7	INTMD	Interrupt Mode Select*2	0: Normal interrupt mode 1: Sequential interrupt mode	R/W

x: Don't care

Note 1. To rewrite the TYP[1:0] bits when the REMCON1.EN bit or REMCON0.ENFLG flag is 1 (REMC is operating), change the values of these bits one bit at a time.

Note 2. These bits can be rewritten when the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC is stopped).

EN Bit (Remote Control)

This bit enables or disables REMC operation.

Use the REMCON0.ENFLG flag to confirm whether operation has started or not.

CSRC[3:0] Bits (Operating Clock Select)

These bits select the operating clock for the REMC.

Satisfy the frequency of the operating clock < the frequency of the PCLKB.

INTMD Bit (Interrupt Mode Select)

This bit selects the interrupt mode.

In normal interrupt mode, an interrupt is generated when the result of the logical OR of the flags for interrupt sources that have been enabled (the corresponding bit set to 1) in the interrupt control register (REMINT) is "true".

In sequential interrupt mode, an interrupt is generated when the result of the logical AND of the flags for interrupt sources that have been enabled (the corresponding bit set to 1) in the REMINT register is "true".

For details on the available interrupt sources and operation in each interrupt mode, see section 46.3.12, Interrupts.

46.2.3 Status Register (REMSTS)

Address(es): REMC0.REMSTS 000A 0B02h

	b7	b6	b5	b4	b3	b2	b1	b0
	SDFLG	D1FLG	D0FLG	HDFLG	BFULFLG	DRFLG	REFLG	CPFLG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CPFLG	Compare Match Flag	0: Mismatch 1: Match	R
b1	REFLG	Receive Error Flag	0: No error has occurred. 1: An error has occurred.	R
b2	DRFLG	Data Receiving Flag	0: Waiting for data reception. 1: Data is being received.	R
b3	BFULFLG	Receive Buffer Full Flag	0: Receive buffer is empty. 1: Receive buffer is full (64 bits received).	R/(W) *1
b4	HDFLG	Header Pattern Match Flag	0: Mismatch 1: Match	R
b5	D0FLG	Data ‘0’ Pattern Match Flag	0: Mismatch 1: Match	R
b6	D1FLG	Data ‘1’ Pattern Match Flag	0: Mismatch 1: Match	R
b7	SDFLG	Special Data Pattern Match Flag	0: Mismatch 1: Match	R

Note: If updating and reading data overlap, an undefined value may be read. For details on reading this register, see section 46.4.7, Reading Registers.

Note: This register becomes 00h when the REMCON1.EN bit is set to 0.

Note 1. Only 0 can be written to clear the flag. However, if this flag is written when changing the REMCON0.INFLG flag, the value read from this flag may become undefined.

CPFLG Flag (Compare Match Flag)

This flag indicates the comparison result between the value of the REMCPD register specified by the REMCPC.CPN[3:0] bits and the data to be stored in the REMDAT1 and REMDAT0 registers.

[Setting condition]

- When the value of the REMCPD register matches the value to be stored in the REMDAT1 and REMDAT0 registers (when the setting value of the REMCPC.CPN[3:0] bits is n, bits n to 0 in the REMCPD register match bits n to 0 in the REMDAT1 and REMDAT0 registers)

[Clearing conditions]

- When the DRFLG flag changes from 0 to 1 (next frame reception starts)
- When the HDFLG flag changes from 0 to 1

REFLG Flag (Receive Error Flag)

This flag indicates that a receive error has occurred. The setting conditions differ depending on the value of the REMCON1.TYP[1:0] bits.

[Setting condition]

When the REMCON1.TYP[1:0] bits are 00b (format A):

- The data ‘0’, data ‘1’, or special data pattern is detected prior to receiving the header pattern
- The width between a rising edge and the next rising edge of the input signal is not the header, data ‘0’, data ‘1’, or special data pattern (when the REMCON0.INV bit is 0)
- A conflict occurs between when data reception is completed (timing when the DRFLG flag changes from 1 to 0) and when the new input signal changes.

When the REMCON1.TYP[1:0] bits are 01b (format B):

- The data ‘0’, data ‘1’, or special data pattern is detected prior to receiving the header pattern
- The width between a falling edge and the next falling edge of the input signal is not the data ‘0’, data ‘1’, or special data pattern (when the REMCON0.INV bit is 0)
- A conflict occurs between when data reception is completed (timing when the DRFLG flag changes from 1 to 0) and when the new input signal changes.

When the REMCON1.TYP[1:0] bits are 10b (format C):

- The width between a rising edge and the next rising edge of the input signal is not the header, data ‘0’, data ‘1’, or special data pattern (when the REMCON0.INV bit is 0)
- A conflict occurs between when data reception is completed (timing when the DRFLG flag changes from 1 to 0) and when the new input signal changes.

[Clearing conditions]

- The header pattern is detected
- When the DRFLG flag changes from 0 to 1 (next frame reception starts).

DRFLG Flag (Data Receiving Flag)

This flag indicates the state of receiving the remote control signal.

[Setting condition]

- Rising edge of REMC internal input signal (when the REMCON0.INV bit is 0)

[Clearing condition]

- This flag becomes 0 after one cycle of the operating clock when the value of the base timer is greater than any value of the HDPMAX, D0PMAX, D1PMAX, SDPMAX, and REMPE registers.

BFULFLG Flag (Receive Buffer Full Flag)

[Setting condition]

- When the value of the REMRBIT register becomes 64

[Clearing conditions]

- When the HDFLG flag changes from 0 to 1
- When the DRFLG flag changes from 0 to 1 (next frame reception starts)
- This flag becomes 0 after one to two cycles when 0 is written to the BFULFLG flag.

HDFLG Flag (Header Pattern Match Flag)

[Setting condition]

- See Table 46.3, Measurement Results and Flags.

[Clearing conditions]

- When the DRFLG flag changes from 0 to 1 (next frame reception starts)
- When the REFLG flag changes from 0 to 1
- See Table 46.3, Measurement Results and Flags.

D0FLG Flag (Data '0' Pattern Match Flag)

[Setting condition]

- See Table 46.3, Measurement Results and Flags.

[Clearing conditions]

- When the DRFLG flag changes from 0 to 1 (next frame reception starts)
- When the REFLG flag changes from 0 to 1
- See Table 46.3, Measurement Results and Flags.

D1FLG Flag (Data '1' Pattern Match Flag)

[Setting condition]

- See Table 46.3, Measurement Results and Flags.

[Clearing conditions]

- When the DRFLG flag changes from 0 to 1 (next frame reception starts)
- When the REFLG flag changes from 0 to 1
- See Table 46.3, Measurement Results and Flags.

SDFLG Flag (Special Data Pattern Match Flag)

[Setting condition]

- See Table 46.3, Measurement Results and Flags.

[Clearing conditions]

- When the DRFLG flag changes from 0 to 1 (next frame reception starts)
- When the REFLG flag changes from 0 to 1
- See Table 46.3, Measurement Results and Flags.

Table 46.3 Measurement Results and Flags

Comparison Result between REMTIM Register Value (Measurement Result) and Each Register	Flag Value			
	HDFLG	D0FLG	D1FLG	SDFLG
Between HDPMIN and HDPMAX	1	0	0	0
Between D0PMIN and D0PMAX	0	1*1	0	0
Between D1PMIN and D1PMAX	0	0	1*1	0
Between SDPMIN and SDPMAX	0	0	0	1*1
Values not listed above	0	0	0	0

Note 1. When the REMCON1.TYP[1:0] bits are 00b or 01b, the D0FLG, D1FLG, and SDFLG flags remain unchanged until the header pattern is detected.

46.2.4 Interrupt Control Register (REMINT)

Address(es): REMC0.REMINT 000A 0B03h

	b7	b6	b5	b4	b3	b2	b1	b0
	SDINT	—	DINT	HDINT	BFULINT	DRINT	REINT	CPINT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CPINT	Compare Match Interrupt Enable*1	0: Disabled 1: Enabled	R/W
b1	REINT	Receive Error Interrupt Enable*1	0: Disabled 1: Enabled	R/W
b2	DRINT	Data Reception Complete Interrupt Enable	0: Disabled 1: Enabled	R/W
b3	BFULINT	Receive Buffer Full Interrupt Enable*1	0: Disabled 1: Enabled	R/W
b4	HDINT	Header Pattern Match Interrupt Enable*1	0: Disabled 1: Enabled	R/W
b5	DINT	Data ‘0’ Pattern or Data ‘1’ Pattern Match Interrupt Enable	0: Disabled 1: Enabled	R/W
b6	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b7	SDINT	Special Data Pattern Match Interrupt Enable*1	0: Disabled 1: Enabled	R/W

Note 1. These bits can be rewritten when the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC is stopped).

46.2.5 Compare Control Register (REMCPC)

Address(es): REMC0.REMCPC 000A 0B05h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CPN[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CPN[3:0]	Compare Bit Count Specification*1	b3 b0 0 0 0 0: Bit 0 in the REMCPD register and bit 0 in the REMDAT0 register are compared. 0 0 0 1: Bit 1 and 0 in the REMCPD register and bit 1 and 0 in the REMDAT0 register are compared. : 0 1 1 1: Bit 7 to 0 in the REMCPD register and bit 7 to 0 in the REMDAT0 register are compared. : 1 0 0 1: Bit 9 to 0 in the REMCPD register and bit 1 and 0 in the REMDAT1 register, bit 7 to 0 in the REMDAT0 register are compared. : 1 1 1 1: Bit 15 to 0 in the REMCPD register and bit 7 to 0 in the REMDAT1 register, bit 7 to 0 in the REMDAT0 register are compared.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits can be rewritten when the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC is stopped).

46.2.6 Compare Value Setting Register (REMCPD)

Address(es): REMC0.REMCPD 000A 0B06h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CPD[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	CPD[15:0]	Compare Value Setting*1	Set the value to be compared with the data in the REMDAT1, REMDAT0 registers when the compare function is used. The REMCPC.CPN[3:0] bits can be used to set the number of bits to be compared.	R/W

Note 1. These bits can be rewritten when the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC is stopped).

46.2.7 Header Pattern Minimum Width Setting Register (HDPMIN)

Address(es): REMC0.HDPMIN 000A 0B08h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	HDPMIN[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	HDPMIN[10:0]	Header Pattern Minimum Width Setting*1	Set the minimum width of header pattern. Setting range: 000h to 7FFh	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits can be rewritten when the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC is stopped).

46.2.8 Header Pattern Maximum Width Setting Register (HDPMAX)

Address(es): REMC0.HDPMAX 000A 0B0Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	HDPMAX[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	HDPMAX[10:0]	Header Pattern Maximum Width Setting*1	Set the maximum width of header pattern. Setting range: 000h to 7FFh	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits can be rewritten when the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC is stopped).

46.2.9 Data ‘0’ Pattern Minimum Width Setting Register (D0PMIN)

Address(es): REMC0.D0PMIN 000A 0B0Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	D0PMIN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	D0PMIN[7:0]	Data ‘0’ Pattern Minimum Width Setting *1	Set the minimum width of data ‘0’ pattern. Setting range: 00h to FFh	R/W

Note 1. These bits can be rewritten when the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC is stopped).

46.2.10 Data ‘0’ Pattern Maximum Width Setting Register (D0PMAX)

Address(es): REMC0.D0PMAX 000A 0B0Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	D0PMAX[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	D0PMAX[7:0]	Data ‘0’ Pattern Maximum Width Setting *1	Set the maximum width of data ‘0’ pattern. Setting range: 00h to FFh	R/W

Note 1. These bits can be rewritten when the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC is stopped).

46.2.11 Data ‘1’ Pattern Minimum Width Setting Register (D1PMIN)

Address(es): REMC0.D1PMIN 000A 0B0Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	D1PMIN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	D1PMIN[7:0]	Data ‘1’ Pattern Minimum Width Setting *1	Set the minimum width of data ‘1’ pattern. Setting range: 00h to FFh	R/W

Note 1. These bits can be rewritten when the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC is stopped).

46.2.12 Data ‘1’ Pattern Maximum Width Setting Register (D1PMAX)

Address(es): REMC0.D1PMAX 000A 0B0Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	D1PMAX[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	D1PMAX[7:0]	Data ‘1’ Pattern Maximum Width Setting *1	Set the maximum width of data ‘1’ pattern. Setting range: 00h to FFh	R/W

Note 1. These bits can be rewritten when the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC is stopped).

46.2.13 Special Data Pattern Minimum Width Setting Register (SDPMIN)

Address(es): REMC0.SDPMIN 000A 0B10h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SDPMIN[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	SDPMIN[10:0]	Special Data Pattern Minimum Width Setting*1	Set the minimum width of special data pattern. Setting range: 000h to 7FFh	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits can be rewritten when the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC is stopped).

46.2.14 Special Data Pattern Maximum Width Setting Register (SDPMAX)

Address(es): REMC0.SDPMAX 000A 0B12h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SDPMAX[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	SDPMAX[10:0]	Special Data Pattern Maximum Width Setting*1	Set the maximum width of special data pattern. Setting range: 000h to 7FFh	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits can be rewritten when the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC is stopped).

46.2.15 Pattern End Setting Register (REMPE)

Address(es): REMC0.REMPE 000A 0B14h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	PE[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	PE[10:0]	Pattern End Width Setting*1	Set the width of pattern end. Setting range: 000h to 7FFh These bits can be used to set the timing at which the REMSTS.DRFLG flag changes from 1 to 0.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits can be rewritten when the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC is stopped).

46.2.16 Receive Bit Count Register (REMRBIT)

Address(es): REMC0.REMRBIT 000A 0B17h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	RBIT[6:1]						RBIT[0]
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RBIT[0]	Receive Bit Count Check 0	Receive bit count can be read. These bits indicate the bit position of the buffer to be stored by counting the detected data ‘0’ pattern or data ‘1’ pattern. • When the receive bit count exceeds 64 (40h), the value returns to 1. • The header pattern and special data pattern are not counted. • If an error is detected while the REMCON0.EC bit is 1, the value is not incremented even when the data ‘0’ pattern or data ‘1’ pattern is detected. • The REMRBIT register becomes 00h when the REMSTS.DRFLG flag changes from 0 to 1. • The REMRBIT register becomes 00h when the REMSTS.HDFLG flag changes from 0 to 1. When 0 is written to the REMRBIT.RBIT[0] bit, the value of the REMRBIT register becomes 00h after one to two cycles of the operating clock.	R/W
b6 to b1	RBIT[6:1]	Receive Bit Count Check 6 to 1		R
b7	—	Reserved	The read value is undefined. The write value should be 0.	R/W

Note: If updating and reading data overlap, an undefined value may be read. For details on reading this register, see section 46.4.7, Reading Registers.

Note: The values of this register are initialized when the REMCON1.EN bit is 0.

46.2.17 Receive Data 0 Register (REMDAT0)

Address(es): REMC0.REMDAT0 000A 0B18h

	b7	b6	b5	b4	b3	b2	b1	b0
	DAT0[7:1]							DAT0[0]
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DAT0[0]	Receive Data 0 Store Bit 0	Receive data is stored. The values of the REMDAT0 to REMDAT7 registers become all 00h after one to two cycles of the operating clock when 0 is written to bit 0 in the REMDAT0 register.	R/W
b7 to b1	DAT0[7:1]	Receive Data 0 Store Bits 7 to 1		R

Note: If updating and reading data overlap, an undefined value may be read. For details on reading this register, see section 46.4.7, Reading Registers.
Note: The values of this register are initialized when the REMCON1.EN bit is 0.

46.2.18 Receive Data j Register (REMDATj) (j = 1 to 7)

Address(es): REMC0.REMDAT1 000A 0B19h, REMC0.REMDAT2 000A 0B1Ah, REMC0.REMDAT3 000A 0B1Bh, REMC0.REMDAT4 000A 0B1Ch, REMC0.REMDAT5 000A 0B1Dh, REMC0.REMDAT6 000A 0B1Eh, REMC0.REMDAT7 000A 0B1Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	DATj[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DATj[7:0]	Receive Data j Store	Receive data is stored.	R

Note: If updating and reading data overlap, an undefined value may be read. For details on reading this register, see section 46.4.7, Reading Registers.
Note: The values of this register are initialized when the REMCON1.EN bit is 0.

When data ‘0’ pattern or data ‘1’ pattern is detected, the result is stored bit by bit as received data. For details on storing received data, see section 46.3.8, Receive Data Buffer.

46.2.19 Measurement Result Register (REMTIM)

Address(es): REMC0.REMTIM 000A 0B20h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TIM[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	TIM[10:0]	Measurement Result	The measurement result of each pattern width can be read. The value of the base timer is captured when one of the following patterns is detected. • Header pattern • Data ‘0’ pattern • Data ‘1’ pattern • Special data pattern • Data pattern other than the above (receive error)	R
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: If updating and reading data overlap, an undefined value may be read. For details on reading this register, see section 46.4.7, Reading Registers.

Note: The values of this register are initialized when the REMCON1.EN bit is 0.

46.2.20 Remote Control Signal Receive Pin Control Register (REMPC)

Address(es): REMC0.REMPC 0008 C29Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	PMC0S[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	PMC0S[2:0]	PMC0 Pin Select	b2 b0 0 0 0: The remote control signal is not received. 0 0 1: The PMC0 signal is input from the PC3 pin. 0 1 0: The PMC0 signal is input from the PB3 pin. 1 0 0: The PMC0 signal is input from the P53 pin. Settings other than those listed above are prohibited.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

46.3 Operation

46.3.1 Overview of REMC Operation

Figure 46.2 shows an example of the remote control signal. The signal begins with a header, followed by a sequence of data. This header differs from the subsequent sequence of data in waveform, allowing the header and the data to be distinguished. The sequence of data contains custom code and data code, and 0 or 1 is distinguished depending on the bit length. After a stop bit, there is an interval during which the signal does not change (frame space), thus constituting a frame.
The time between the edges of the external input signal is measured using the base timer in the REMC. The patterns of the remote control signal are detected and the data is captured according to the measurement results.

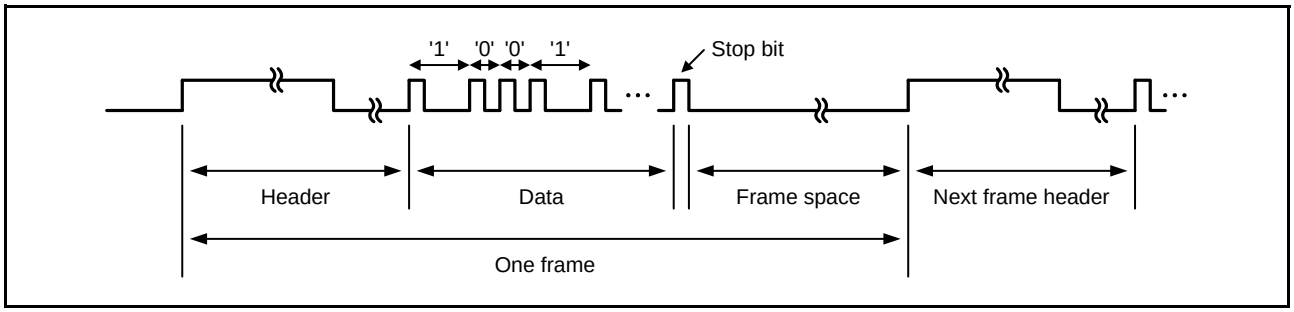


Figure 46.2 Example of Remote Control Signal

46.3.2 Initial Setting

Initialize the REMC according to the procedure shown in Figure 46.3 to receive the remote control signal.
Set the REMCON1.EN bit to 0 if the REMC is operating. Then the REMCON0.ENFLG flag becomes 0 and the REMC stops the operation.
Set the format for the remote control signal waveform by the REMCON1.TYP[1:0] bits; select the signal inversion or non-inversion by the REMCON0.INV bit; select the operating clock by the REMCON1.CSRC[3:0] bits; and set the digital filter by the REMCON0.FIL and REMCON0.FILSEL bits, while the REMCON0.ENFLG flag is 0. Set the detecting width for each data pattern into the HDPMIN, HDPMAX, D0PMIN, D0PMAX, D1PMIN, D1PMAX, SDPMIN, SDPMAX, and REMPE registers. Make any other settings such as enabling interrupts by the REMINT register and setting of the compare function by the REMCPC and REMCPD registers if required.
After all necessary register settings are completed, set the REMCON1.EN bit to 1 to start REMC operation.

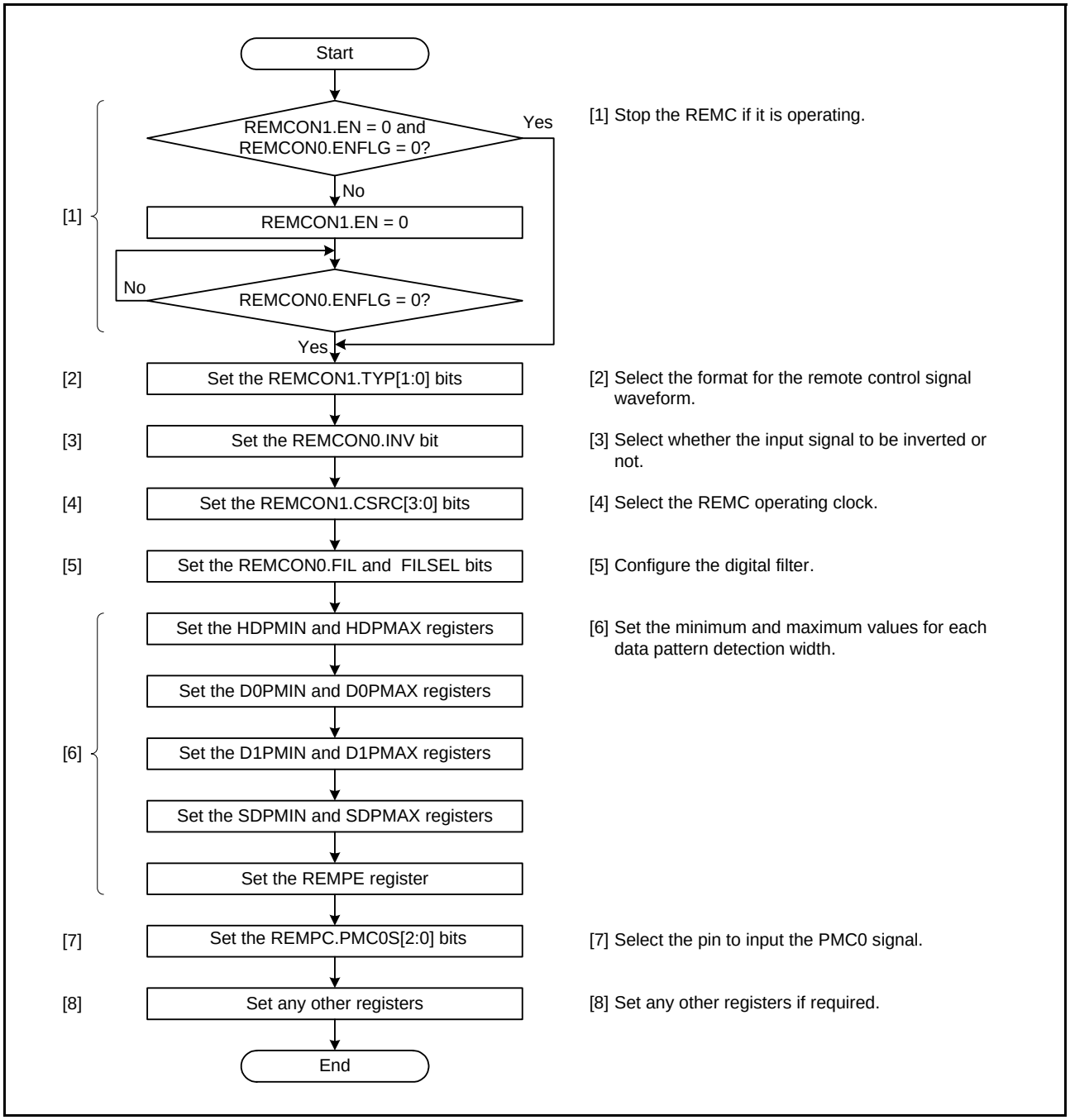


Figure 46.3 Example of Flowchart for Initial Settings of REMC

46.3.3 Pattern Setting

The format for capturing the remote control signal reception waveform can be set by setting the REMCON1.TYP[1:0] bits. Figure 46.4 and Figure 46.5 show examples of a remote control signal reception waveform captured by setting the REMCON1.TYP[1:0] bits.

When the REMCON1.TYP[1:0] bits are 00b (format A)

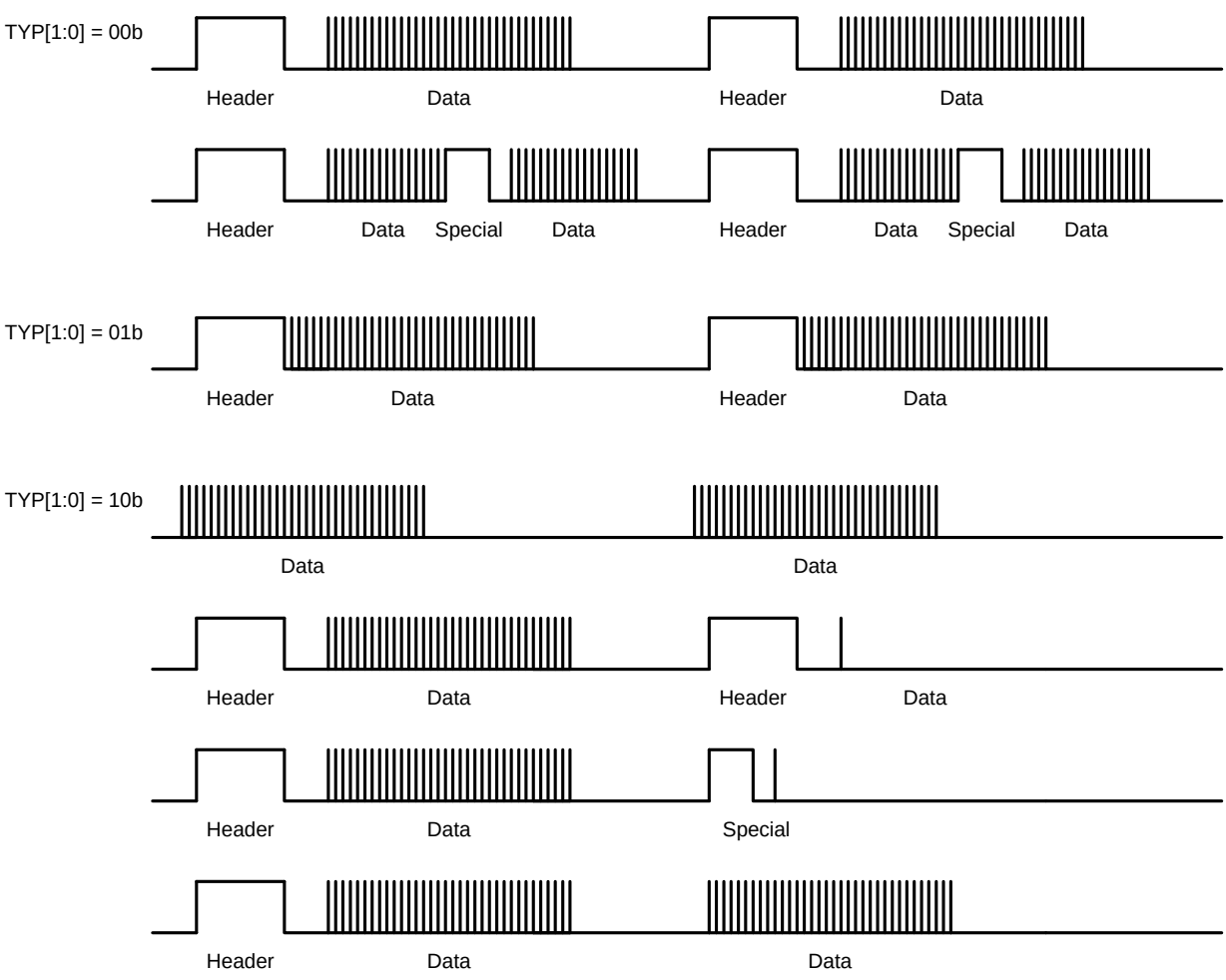
The measured result is determined from the setting value of the header pattern at the rising edge of the internal input signal.
When the header pattern is received, the measured result is determined from the setting values of the data ‘0’, data ‘1’ and special data patterns at the rising edge of the internal input signal.

When the REMCON1.TYP[1:0] bits are 01b (format B)

The measured result is determined from the setting value of the header pattern at the falling edge of the internal input signal.
When the header pattern is received, the measured result is determined from the setting values of the data ‘0’, data ‘1’ and special data patterns at the falling edge of the internal input signal.
The header pattern is detected once within one frame.

When the REMCON1.TYP[1:0] bits are 10b (format C)

The measured result is determined from the setting values of the header, data ‘0’, data ‘1’ and special data patterns at the rising edge of the internal input signal.



TYP[1:0]	Header Pattern	Data '0' Pattern	Data '1' Pattern	Special Data Pattern
TYP[1:0] = 00b				
TYP[1:0] = 01b				
TYP[1:0] = 10b				

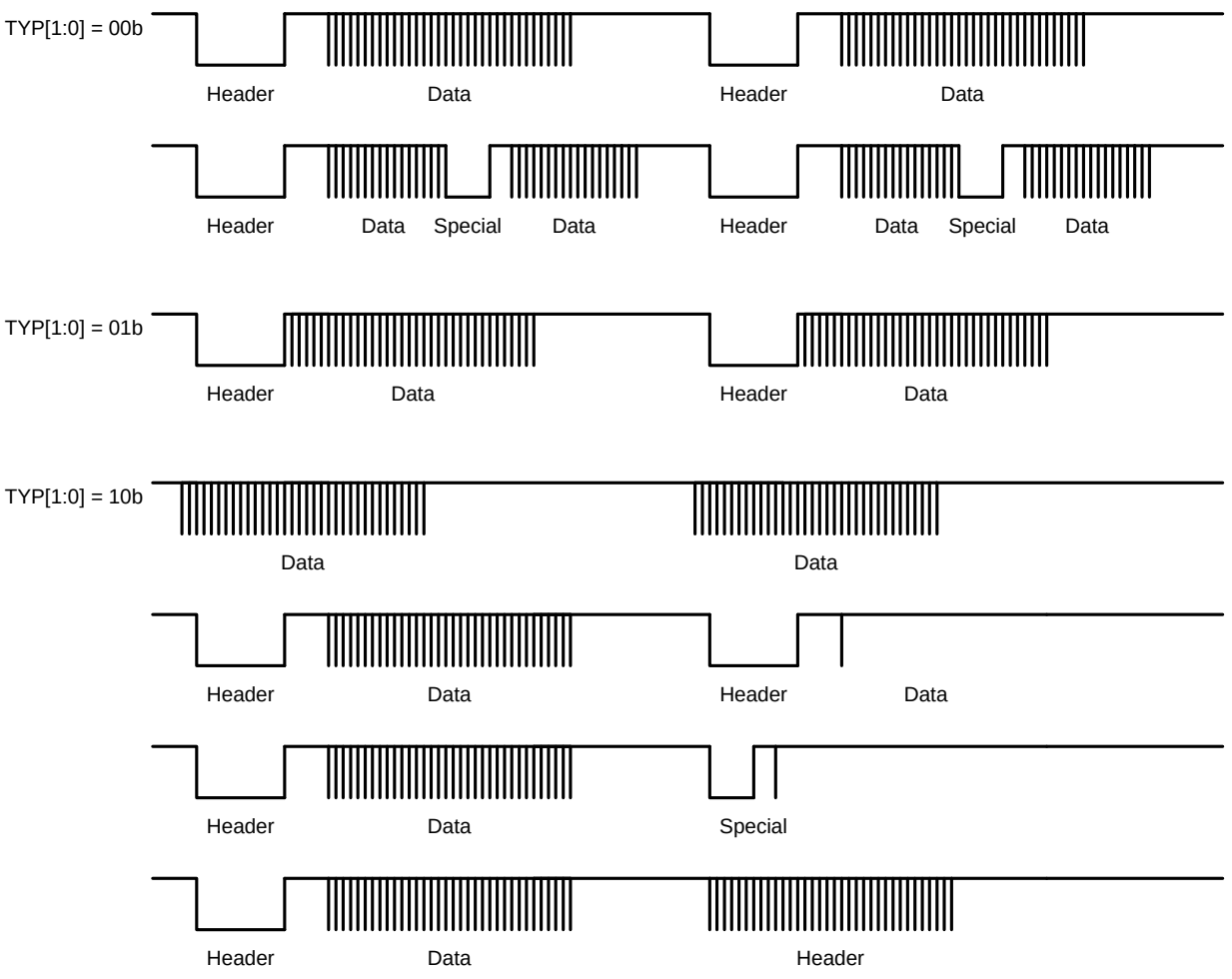
TYP[1:0]: Bits in the REMCON1 register

↑↓・・・Measurement starts

↑↓・・・Pattern measurement

When TYP[1:0] = 01b, the number of detecting the header pattern within the receive frame is one.
The above diagram applies when the REMCON0.INV bit is 0.

Figure 46.4 Example of Remote Control Signal Reception Waveform Captured by Setting REMCON1.TYP[1:0] Bits (REMCON0.INV = 0)



TYP[1:0]	Header Pattern	Data '0' Pattern	Data '1' Pattern	Special Data Pattern
TYP[1:0] = 00b				
TYP[1:0] = 01b				
TYP[1:0] = 10b				

TYP[1:0]: Bits in the REMCON1 register

↑↓ · · · Measurement starts

↑↓ · · · Pattern measurement

When TYP[1:0] = 01b, the number of detecting the header pattern within the receive frame is one.
The above diagram applies when the REMCON0.INV bit is 1.

Figure 46.5 Example of Remote Control Signal Reception Waveform Captured by Setting REMCON1.TYP[1:0] Bits (REMCON0.INV = 1)

46.3.4 Operating Clocks

The REMC can use one of the following clocks as its operating clock: the divided clock of the peripheral module clock (PCLKB), the sub-clock supplied from the sub-clock oscillator, or TMR compare match output.
When supplying the sub-clock to the REMC, take note of the procedure for supplying the clock. The following describes how to supply these clocks.

46.3.4.1 When Using Sub-Clock as the REMC Operating Clock

The sub-clock can be used as the REMC operating clock. For the procedure to start the sub-clock oscillation, refer to section 9, Clock Generation Circuit. After the sub-clock oscillation is stabilized, set the REMCON1.CSRC[3:0] bits to x100b (sub-clock).

46.3.4.2 Using TMR Compare Match Output as REMC Operating Clock

The TMR compare match output can be supplied as the REMC operating clock. TMO0 can be supplied to the REMC0, respectively. For details on the TMR compare match output, refer to section 28, 8-Bit Timer (TMRb).

46.3.5 PMC0 Input

The options below can be selected in PMC0 input.

- Input polarity
- Digital filter

Figure 46.6 shows the configuration of PMC0 internal input signal generation.

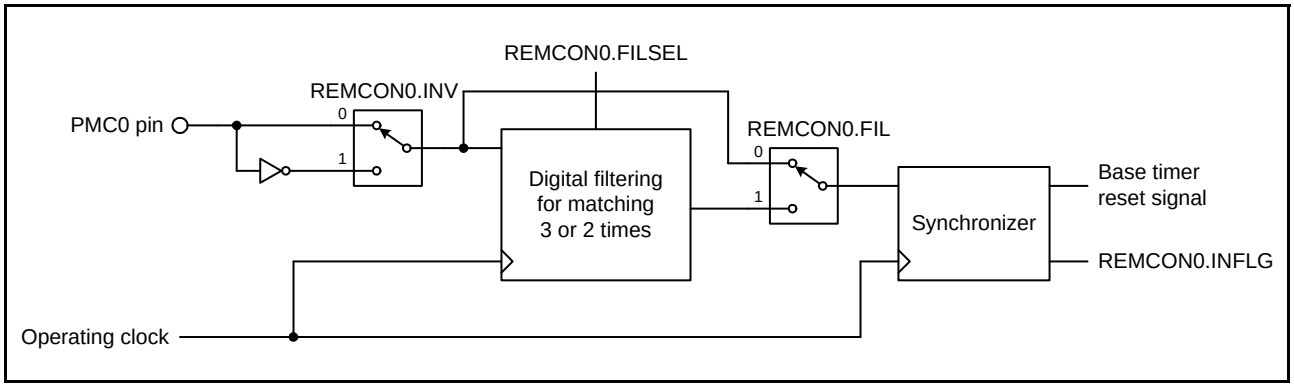
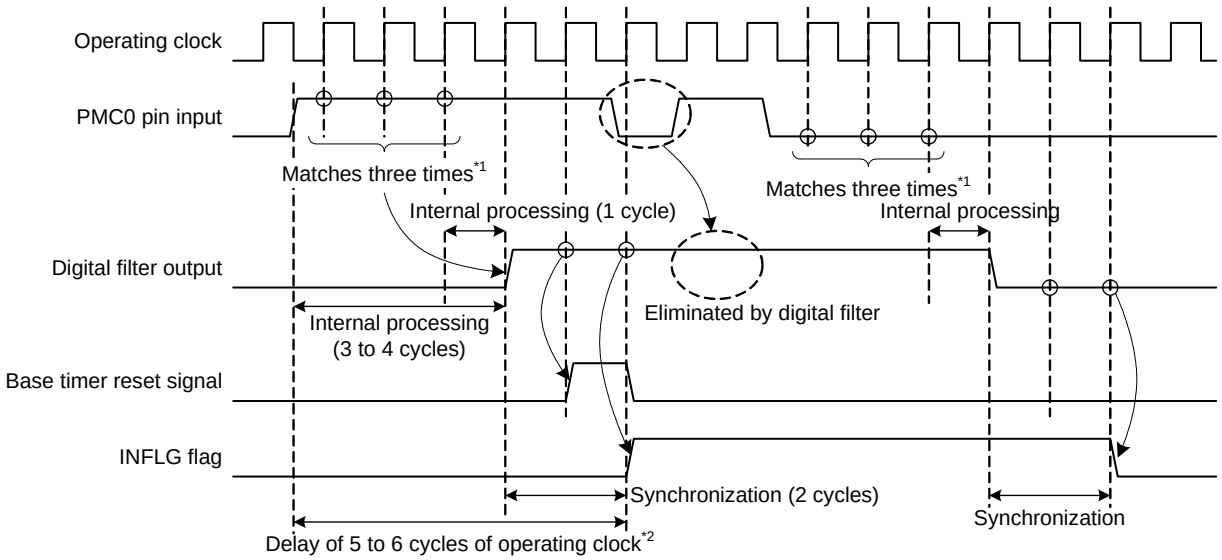


Figure 46.6 PMC0 Internal Input Signal Generation Configuration

The input polarity of the PMC0 pin can be inverted. Whether to invert or not can be selected by the REMCON0.INV bit.
When the REMCON0.FIL bit is 1 (digital filter enabled), if the signal input to the PMC0 pin holds the same level for k sequential cycles (k = 3 or 2; value selected by the REMCON0.FILSEL bit), that level is transferred to the internal circuit. This enables noise to be eliminated from k cycles of the sampling clock.
Input to the PMC0 pin is transferred as the REMCON0.INFLG flag (input signal flag) and the base timer reset signal to the internal circuit in synchronization with the operating clock. The base timer reset signal is used to initialize the internal base timer to the pattern detection corresponding to the REMCON1.TYP[1:0] setting. There is a delay caused by internal processing after the input to the PMC0 pin is changed and before these signals are generated. Figure 46.7 shows digital filtering for PMC0 input.



Note 1. The above applies when the REMCON0.FIL bit is 1 (digital filter enabled) and the REMCON0.FILSEL bit is 0 (matches three times). When the FILSEL bit is 1 (matches twice), the signal is output earlier by one cycle of operating clock than that in this figure.

Note 2. When the FIL bit is 0 (digital filter disabled), a delay becomes one to two cycles of the operating clock for synchronization.

Figure 46.7 Digital Filtering for PMC0 Input

46.3.6 Pattern Detection

The REMC has a function that detects the following patterns.

- Header pattern
- Data ‘0’ pattern
- Data ‘1’ pattern
- Special data pattern

Using the base timer included in the REMC, the time between the edges of the external input signal is measured to determine which pattern matches the measurement result. This enables detection of the remote control signal and capturing the data. The width for determining each pattern can be set to any value using each pattern setting register. Figure 46.8 shows the waveform of REMC operation.

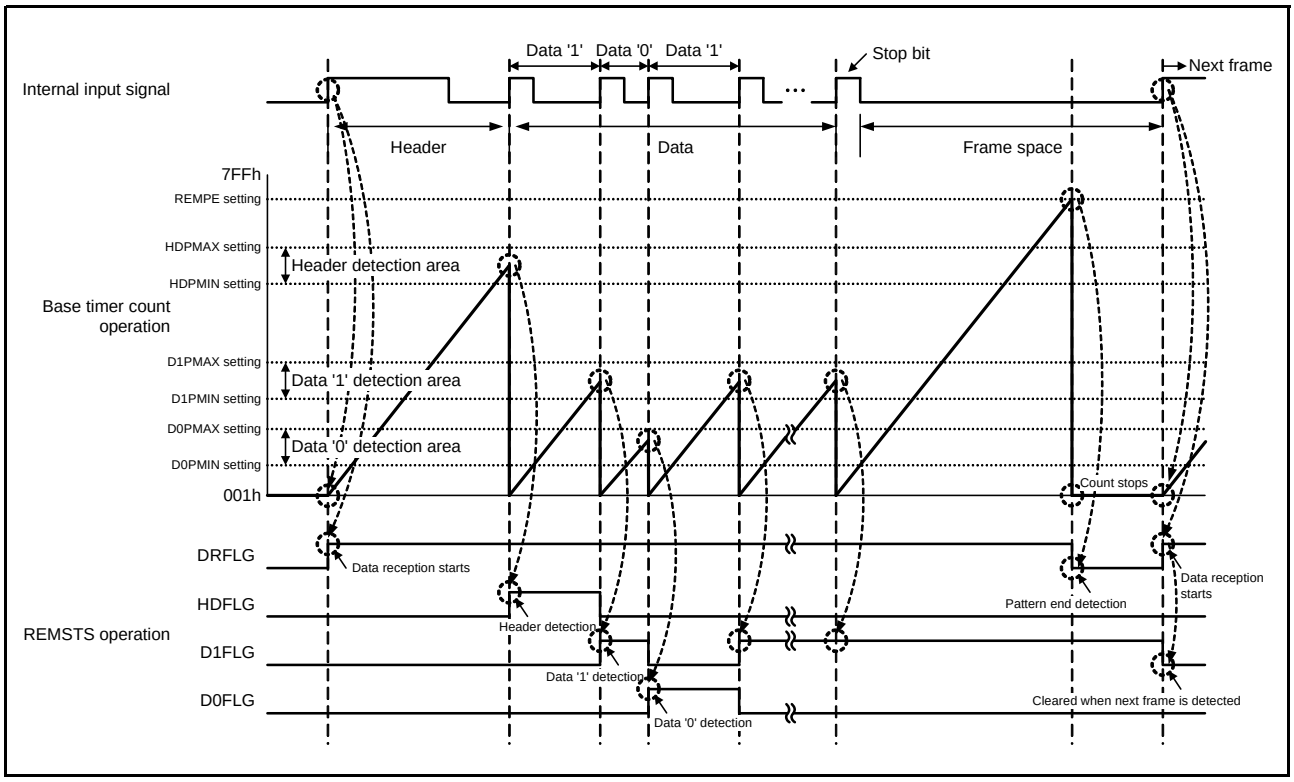


Figure 46.8 Waveform of REMC Operation

46.3.6.1 Header Pattern Detection

The header pattern can be detected by setting the minimum width of the header pattern in the HDPMIN register and the maximum width in the HDPMAX register.
The minimum and maximum widths of the header pattern must be “1 < HDPMIN register value ≤ HDPMAX register value”.

$$\text{Setting value n} = \frac{\text{Minimum width (maximum width) of header pattern}}{\text{Operating clock cycle time}}$$

When not using the header pattern, set the HDPMIN and HDPMAX registers to 000h.
Make sure that the setting value of the header pattern is different from the setting values of data ‘0’, data ‘1’, and special data patterns, and the setting ranges are not overlapped.

When the REMCON1.TYP[1:0] bits are 00b or 01b, if the data ‘0’, data ‘1’, or special data pattern is detected before the header pattern is detected, the following occur:

- The REMSTS.REFLG flag becomes 1 (an error has occurred).
- The REMSTS.D0FLG, REMSTS.D1FLG, and REMSTS.SDFLG flags remain unchanged.
- The REMDAT0 to REMDAT7 registers remain unchanged.

When the REMCON1.TYP[1:0] bits are 01b, the number of detecting the header pattern is one while the DRFLG flag is 1.

46.3.6.2 Data ‘0’ Pattern Detection

The data ‘0’ pattern can be detected by setting the minimum width of the data ‘0’ pattern in the D0PMIN register and the maximum width in the D0PMAX register.
The minimum and maximum widths of the data ‘0’ pattern must be “1 < D0PMIN register value ≤ D0PMAX register value”.

$$\text{Setting value n} = \frac{\text{Minimum width (maximum width) of data ′0′ pattern}}{\text{Operating clock cycle time}}$$

When not using the data ‘0’ pattern, set the D0PMIN and D0PMAX registers to 00h.
Make sure that the setting value of the data ‘0’ pattern is different from the setting values of the header, data ‘1’, and special data patterns, and the setting ranges are not overlapped.

When the REMCON1.TYP[1:0] bits are 00b or 01b, if the data ‘0’ pattern is detected before the header pattern is detected, the following occur:

- The REMSTS.REFLG flag becomes 1 (an error has occurred).
- The REMSTS.D0FLG, REMSTS.D1FLG, and REMSTS.SDFLG flags remain unchanged.
- The REMDAT0 to REMDAT7 registers remain unchanged.

46.3.6.3 Data ‘1’ Pattern Detection

The data ‘1’ pattern can be detected by setting the minimum width of the data ‘1’ pattern in the D1PMIN register and the maximum width in the D1PMAX register.
The minimum and maximum widths of the data ‘1’ pattern must be “1 < D1PMIN register value ≤ D1PMAX register value”.

$$\text{Setting value n} = \frac{\text{Minimum width (maximum width) of data ′1′ pattern}}{\text{Operating clock cycle time}}$$

When not using the data ‘1’ pattern, set the D1PMIN and D1PMAX registers to 00h.
Make sure that the setting value of the data ‘1’ pattern is different from the setting values of the header, data ‘0’, and special data patterns, and the setting ranges are not overlapped.

When the REMCON1.TYP[1:0] bits are 00b or 01b, if the data ‘1’ pattern is detected before the header pattern is detected, the following occur:

- The REMSTS.REFLG flag becomes 1 (an error has occurred).
- The REMSTS.D0FLG, REMSTS.D1FLG, and REMSTS.SDFLG flags remain unchanged.
- The REMDAT0 to REMDAT7 registers remain unchanged.

46.3.6.4 Special Data Pattern Detection

The special data pattern can be detected by setting the minimum width of the special data pattern in the SDPMIN register and the maximum width in the SDPMAX register.
The minimum and maximum widths of the special data pattern must be “1 < SDPMIN register value ≤ SDPMAX register value”.

$$\text{Setting value n} = \frac{\text{Minimum width (maximum width) of special data pattern}}{\text{Operating clock cycle time}}$$

When not using the special data pattern, set the SDPMIN and SDPMAX registers to 000h.
Make sure that the setting value of the special data pattern is different from the setting values of the header, data ‘0’, and data ‘1’ patterns, and the setting ranges are not overlapped.

When the REMCON1.TYP[1:0] bits are 00b or 01b, if the special data pattern is detected before the header pattern is detected, the following occur:

- The REMSTS.REFLG flag becomes 1 (an error has occurred).
- The REMSTS.D0FLG, REMSTS.D1FLG, and REMSTS.SDFLG flags remain unchanged.
- The REMDAT0 to REMDAT7 registers remain unchanged.

46.3.6.5 Examples of Setting Pattern Setting Registers

For the header, data ‘0’, data ‘1’, and special data setting registers, make sure that the minimum to maximum values of each pattern are different, and the setting ranges do not overlap as shown in Figure 46.9.

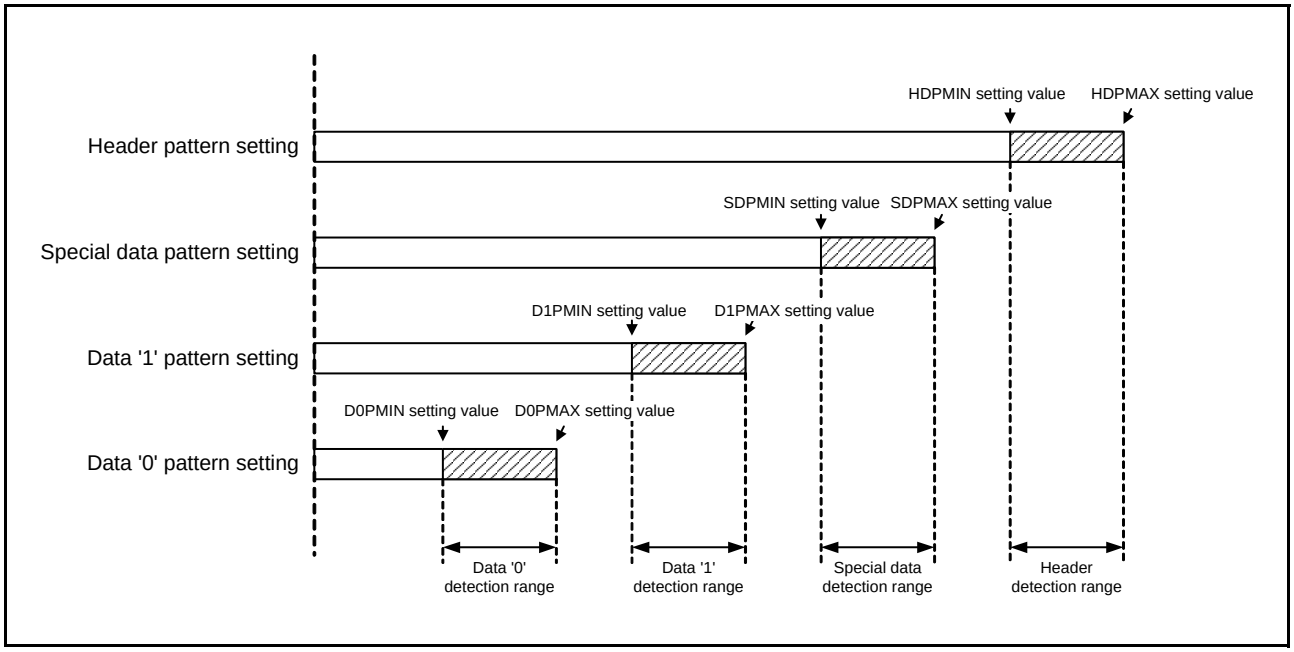


Figure 46.9 Examples of Setting Pattern Setting Registers

46.3.6.6 Updating Status Flags upon Pattern Detection

The detected patterns can be confirmed by reading the following flags: header pattern match flag (REMSTS.HDFLG), data ‘0’ pattern match flag (REMSTS.D0FLG), data ‘1’ pattern match flag (REMSTS.D1FLG), and special data pattern match flag (REMSTS.SDFLG). These flags are negated when a different pattern is detected. If a pattern other than the above patterns is detected, it is detected as an error pattern. This can be confirmed by reading the receive error flag (REMSTS.REFLG). This flag is negated when the next frame is received. Figure 46.10 shows pattern detection and an example of flag operation.

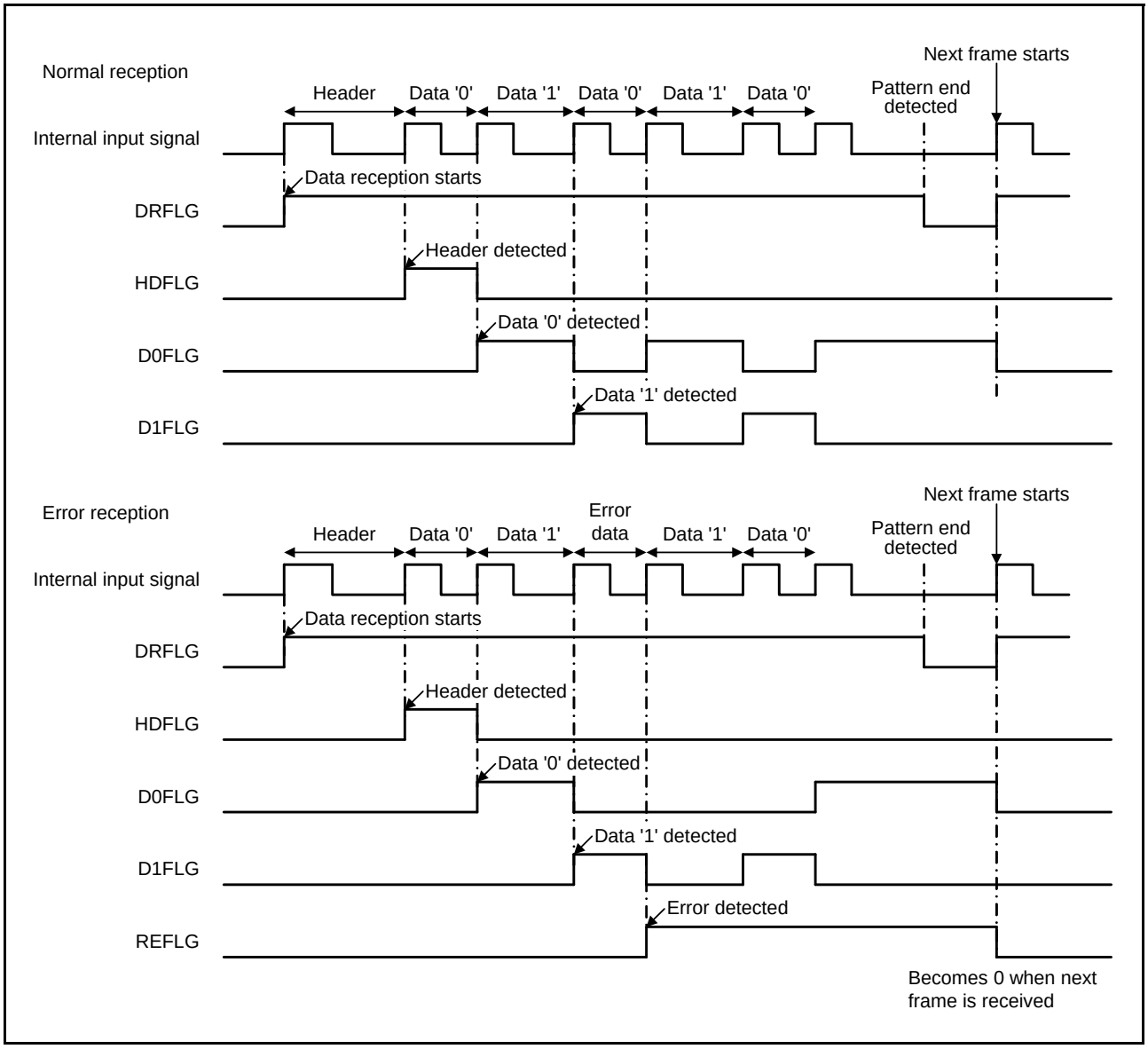


Figure 46.10 Example of Flag Operation

46.3.7 Pattern End

The timing when the REMSTS.DRFLG flag becomes 0 can be set.

When setting the REMPE register, be sure to set that the REMPE value > HDPMAX, D0PMAX, D1PMAX, or SDPMAX value.

When the REMPE value ≤ HDPMAX, D0PMAX, D1PMAX, or SDPMAX value, the REMPE register cannot be used to set the timing when the REMSTS.DRFLG flag becomes 0. In this case, data reception is completed according to the largest value from among the setting values of the HDPMAX, D0PMAX, D1PMAX, and SDPMAX registers. Figure 46.11 shows operation of the data reception complete flag for each pattern end setting.

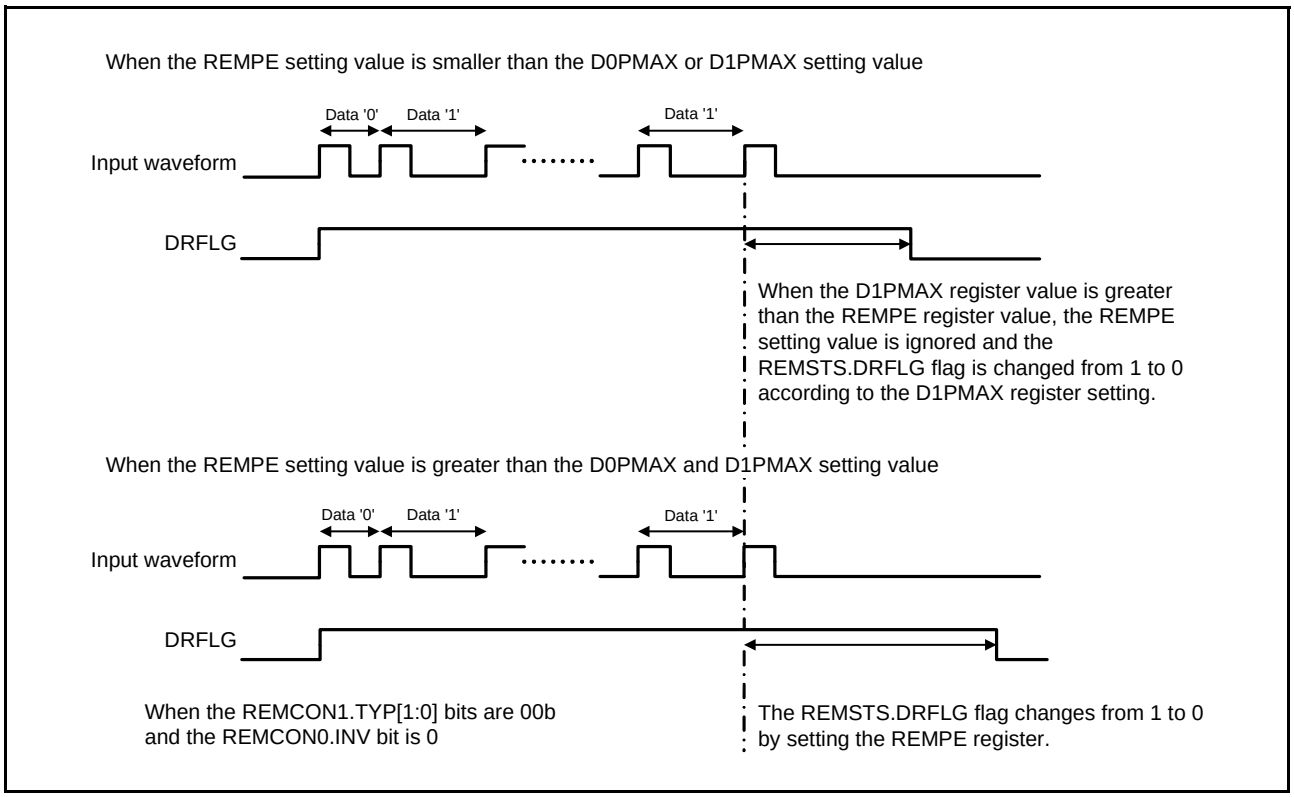


Figure 46.11 Operation of Data Reception Complete Flag for Each Pattern End Setting

46.3.8 Receive Data Buffer

The receive data j register (REMDATj) (j = 0 to 7) is an 8-byte (64-bit) buffer for storing received data. When data ‘0’ pattern or data ‘1’ pattern is detected, the detection result is sequentially stored starting from the REMDAT0.DAT0[0] bit as shown in Figure 46.12. The REMRBIT register is counted up at the same time, so the number of the current received bits can be checked by reading the REMRBIT register. See Table 46.4 for the relationship between the number of received bits and the location where data is stored. The values of the REMDATj and REMRBIT registers do not change even when the header pattern or special pattern is received. If the REMDATj or REMRBIT register is read while the data is being updated, the value read may be undefined.

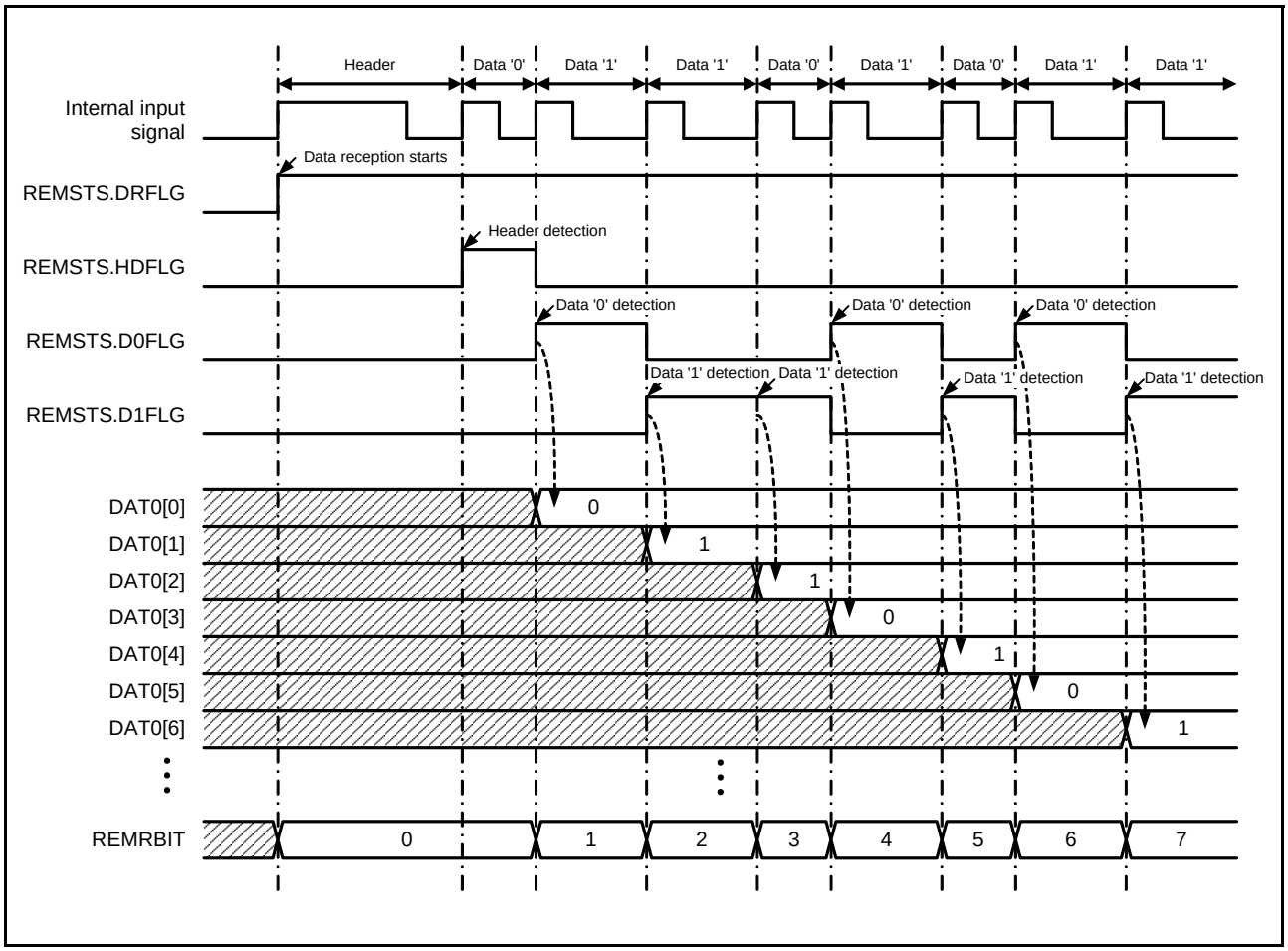


Figure 46.12 Operation of Receive Data Buffer

Table 46.4 Relationship between Number of Received Bits and Location Where Data is Stored

Number of Received Bits	Location Where Data is Stored	
	Register Name	Bit Name
1	REMDAT0	DAT0[0]
2		DAT0[1]
3		DAT0[2]
4		DAT0[3]
5		DAT0[4]
6		DAT0[5]
7		DAT0[6]
8		DAT0[7]
9	REMDAT1	DAT1[0]
10		DAT1[1]
11		DAT1[2]
12		DAT1[3]
13		DAT1[4]
14		DAT1[5]
15		DAT1[6]
16		DAT1[7]
17	REMDAT2	DAT2[0]
18		DAT2[1]
19		DAT2[2]
20		DAT2[3]
21		DAT2[4]
22		DAT2[5]
23		DAT2[6]
24		DAT2[7]
25	REMDAT3	DAT3[0]
26		DAT3[1]
27		DAT3[2]
28		DAT3[3]
29		DAT3[4]
30		DAT3[5]
31		DAT3[6]
32		DAT3[7]

Number of Received Bits	Location Where Data is Stored	
	Register Name	Bit Name
33	REMDAT4	DAT4[0]
34		DAT4[1]
35		DAT4[2]
36		DAT4[3]
37		DAT4[4]
38		DAT4[5]
39		DAT4[6]
40		DAT4[7]
41	REMDAT5	DAT5[0]
42		DAT5[1]
43		DAT5[2]
44		DAT5[3]
45		DAT5[4]
46		DAT5[5]
47		DAT5[6]
48		DAT5[7]
49	REMDAT6	DAT6[0]
50		DAT6[1]
51		DAT6[2]
52		DAT6[3]
53		DAT6[4]
54		DAT6[5]
55		DAT6[6]
56		DAT6[7]
57	REMDAT7	DAT7[0]
58		DAT7[1]
59		DAT7[2]
60		DAT7[3]
61		DAT7[4]
62		DAT7[5]
63		DAT7[6]
64		DAT7[7]

Note: When the data exceeds 64 bits, the REMDATj register is sequentially overwritten from the first bit.

When 0 is written to the REMDAT0.DAT0[0] bit, the values of the REMDAT0 to REMDAT7 registers become 00h after one to two cycles of the operating clock. Figure 46.13 shows the REMDATj/REMRBIT register operation when 00h is written to the REMDAT0 register.

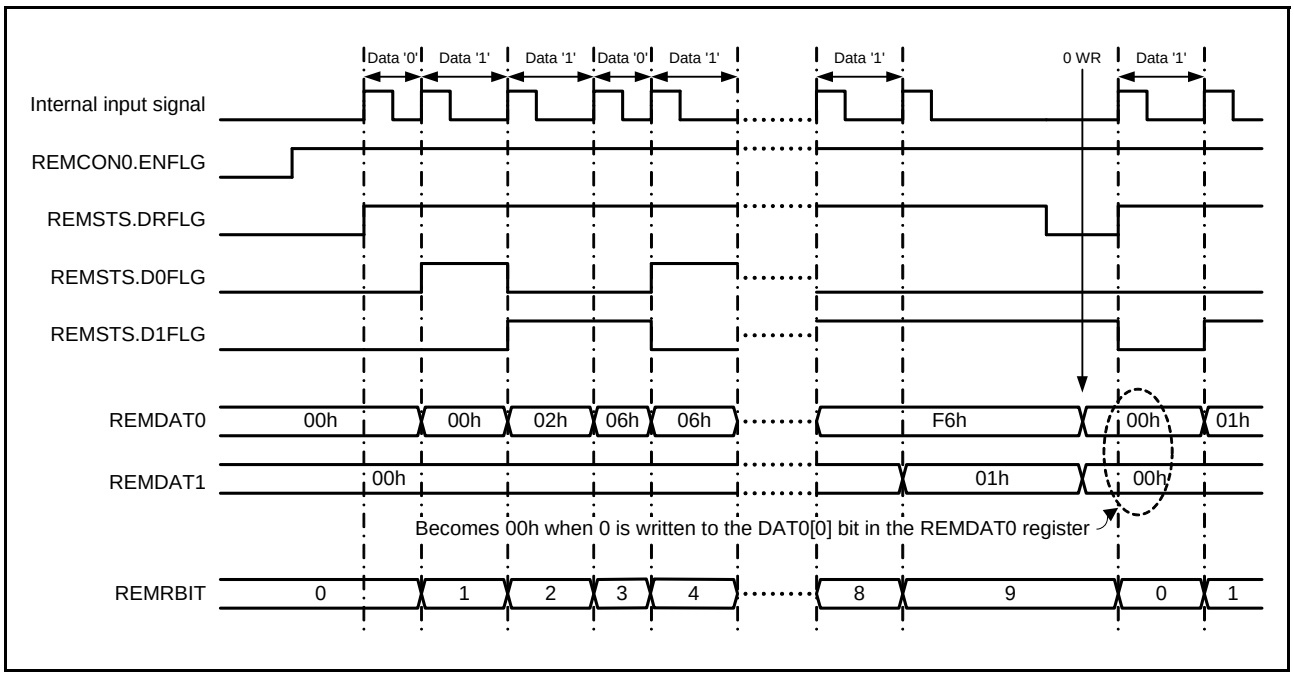


Figure 46.13 REMDATj/REMRBIT Register Operation (00h is Written to REMDAT0 Register)

When 0 is written to the REMRBIT.RBIT[0] bit, the value of the REMRBIT register becomes 00h after one to two cycles of the operating clock. When the REMCON1.TYP[1:0] bits are 00b or 10b, if the header pattern is detected during data reception, the value of the REMRBIT register is initialized to 00h and the received data is sequentially overwritten from the REMDAT0.DAT0[0] bit. Figure 46.14 shows operation of header pattern detection during data reception.

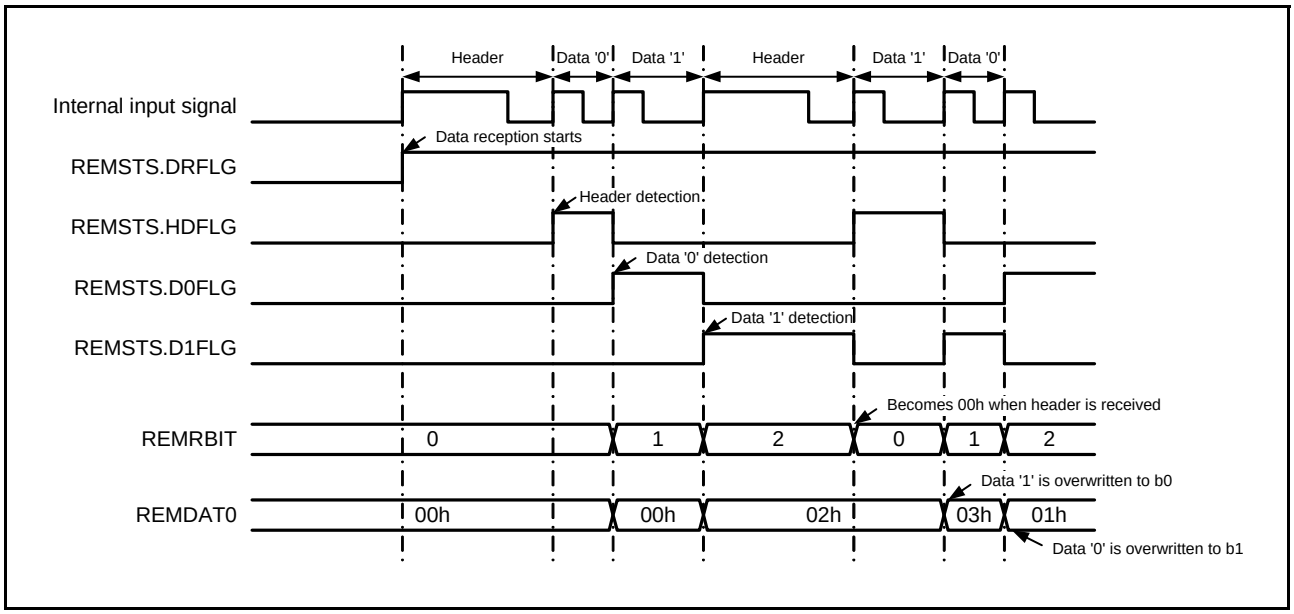


Figure 46.14 Operation of Header Pattern Detection during Data Reception

When the data exceeds 64 bits, the buffer is sequentially overwritten from the first bit. Figure 46.15 shows the REMRBIT register operation when the REMSTS.BFULFLG flag becomes 1.

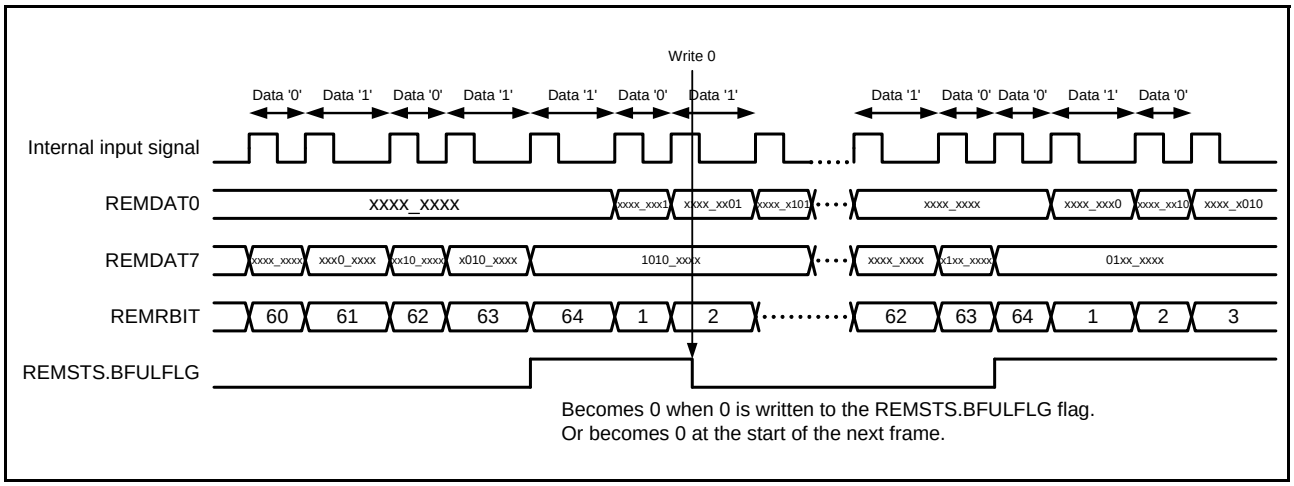


Figure 46.15 REMRBIT Register Operation (REMSTS.BFULFLG Flag = 1)

46.3.9 Compare Function

The REMC has a function to compare the value of the REMCPD register with the value of the REMDAT1 and REMDAT0 registers. As a result of comparison, it can be detected that the first 1 to 16 bits of the remote control signal are the specific values. Figure 46.16 shows the operation timing of the receive buffer and the compare function.

When using the compare function, set the following:

- Select bits to be compared by setting the REMCPC.CPN[3:0] bits (when the setting value is n, bits n to 0 are compared. n: 0 to 15).
- Set the compare data in the REMCPD register.
  When the value of the REMRBIT register becomes the bit count specified by the REMCPC.CPN[3:0] bits, if the stored comparison result between the REMCPD register and the REMDAT1 and REMDAT0 registers matches, the REMSTS.CPFLG flag becomes 1 (compare match).

When the value of the REMRBIT register matches the bit count specified by the REMCPC.CPN[3:0] bits during reception of 64 bits or more, even if the comparison result between the REMCPD register and the REMDAT1 and REMDAT0 registers matches, the REMSTS.CPFLG flag does not become 1 (compare match).

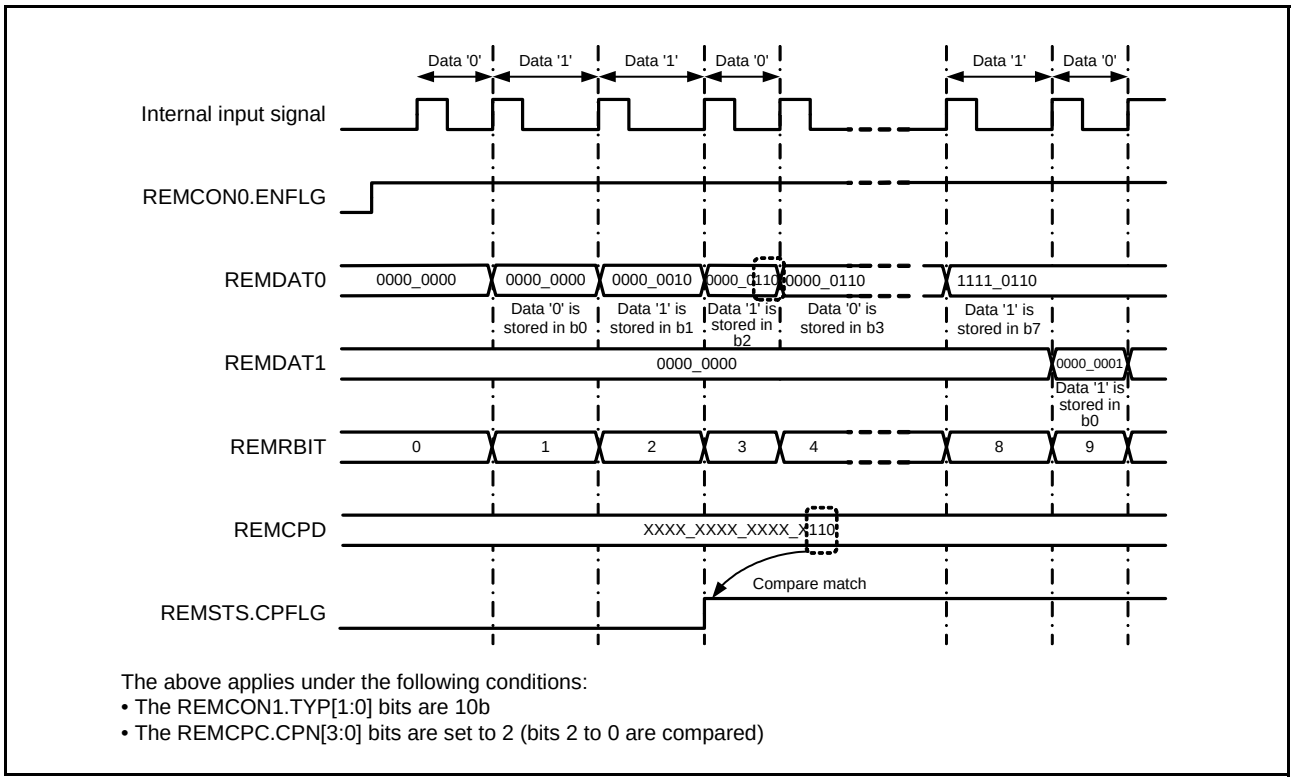


Figure 46.16 Receive Buffer and Compare Function

46.3.10 Error Pattern Reception

When the error pattern is detected during data reception, subsequent operation differs depending on the setting of the REMCON0.EC bit.

Figure 46.17 shows operation of the REMDAT0 and REMRBIT registers when the REMCON0.EC bits are set to 0. If an error is detected while the REMCON0.EC bit is 0, the data when the error is detected is not captured, but the data is captured when the data ‘0’ pattern or data ‘1’ pattern is detected later.

Figure 46.18 shows operation of the REMDAT0 and REMRBIT registers when the REMCON0.EC bits are set to 1. If an error is detected while the REMCON0.EC bit is 1, the values of the REMRBIT and REMDAT0 to REMDAT7 registers are not updated even when the data ‘0’ pattern or data ‘1’ pattern is detected later. Once the REMSTS.DRFLG flag is cleared and after data reception is completed, if data reception starts again, the REMSTS.REFLG flag is cleared and the data is captured.

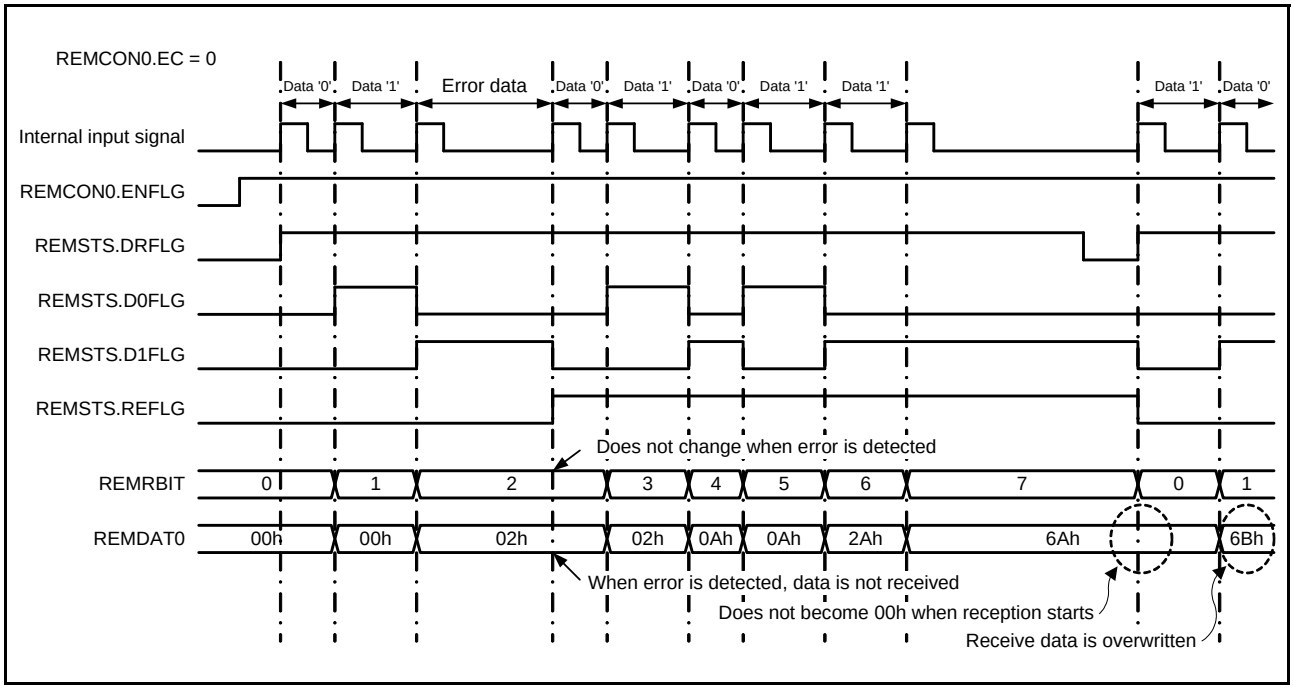


Figure 46.17 REMDAT0 and REMRBIT Registers Operation upon Error Detection (REMCON0.EC Bit = 0)

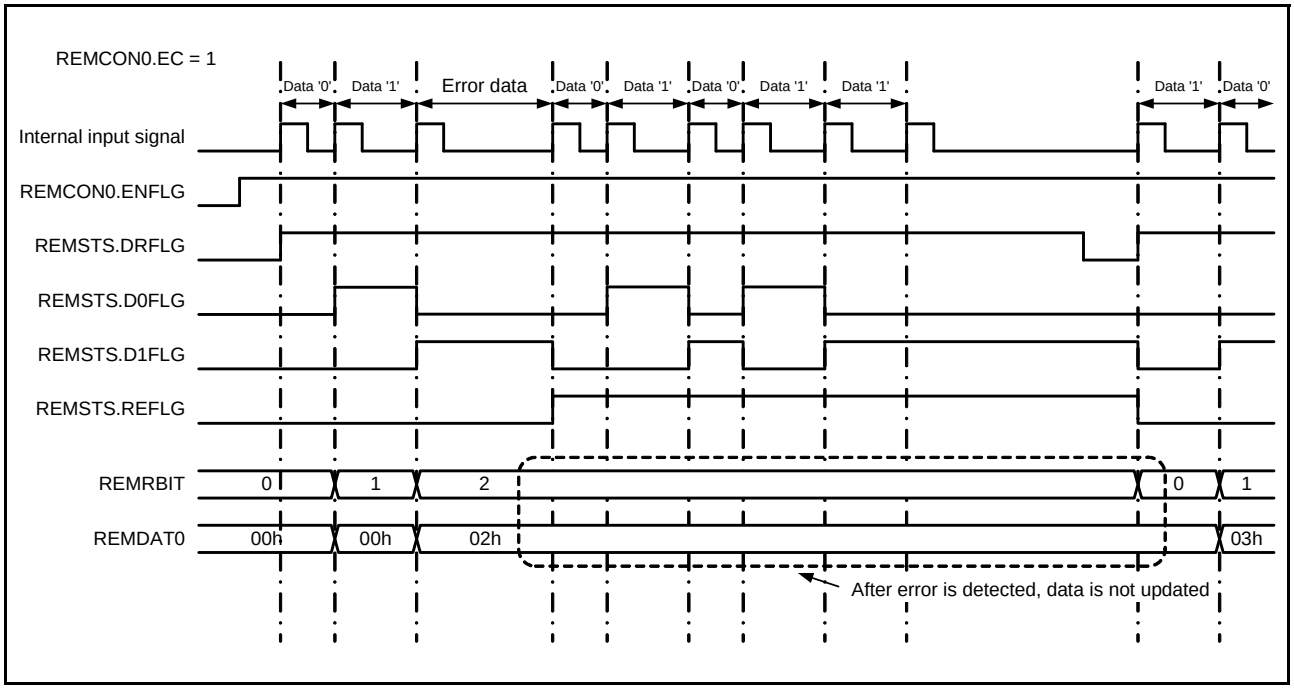


Figure 46.18 REMDAT0 and REMRBIT Registers Operation upon Error Detection (REMCON0.EC Bit = 1)

46.3.11 Storing Base Timer Value When Pattern is Detected

The measurement result register (REMTIM) stores the base timer value when one of the following patterns is detected. This makes it possible to measure each pattern width. Figure 46.19 shows an operation example of the measurement function.

- Header pattern
- Data ‘0’ pattern
- Data ‘1’ pattern
- Special data pattern
- Data pattern other than the above (receive error)

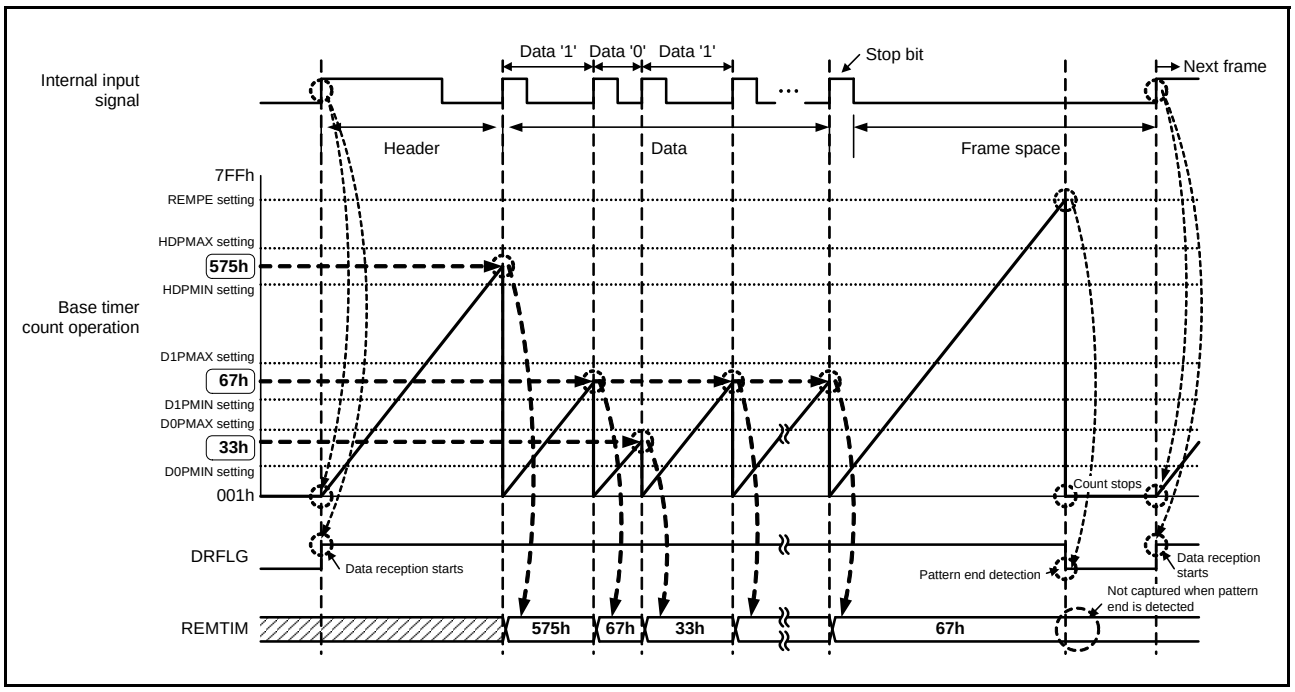


Figure 46.19 Operation Example of Measurement Function

46.3.12 Interrupts

The REMC has the following seven interrupt sources: compare match, receive error, data reception complete, receive buffer full, header pattern match, data ‘0’ pattern or data ‘1’ pattern match, and special data pattern match. All of these interrupt sources are assigned to a single vector number.

Table 46.5 lists the REMC interrupt sources, and Table 46.6 lists the interrupt modes and REMCI0 interrupt request generation conditions.

In normal interrupt mode, if the enable bit for an interrupt source in the REMINT register is set to 1 when the given interrupt request generation condition is satisfied, an REMCI0 interrupt request is output.

The condition for generating an interrupt request in sequential interrupt mode differs from that in normal interrupt mode. In sequential interrupt mode, an REMCI0 interrupt request is output when either of the following conditions is satisfied.

- If the enable bits for the generation of an interrupt request in response to the four interrupt sources: compare match, data reception complete, header pattern match, and special data pattern match are set to 1 by the corresponding bits in the REMINT register, the interrupt request is generated when the interrupt request generation conditions for all enabled interrupt sources among the four sources are satisfied.
- For any of the interrupt sources other than the four above, an interrupt is generated if this is enabled by the corresponding bit in the REMINT register.

Refer to section 15, Interrupt Controller (ICUE) for details on interrupt control.

Table 46.5 REMC Interrupt Sources

Interrupt Source	Status Flag	Interrupt Enable Bit	Each Interrupt Request Generation Condition
Compare match	REMSTS.CPFLG	REMINT.CPINT	When the REMSTS.CPFLG flag changes from 0 to 1
Receive error	REMSTS.REFLG	REMINT.REINT	When the REMSTS.REFLG flag changes from 0 to 1 (When a receive error is detected)
Data reception complete	REMSTS.DRFLG	REMINT.DRINT	When the REMSTS.DRFLG flag changes from 1 to 0
Receive buffer full	REMSTS.BFULFLG	REMINT.BFULINT	When the REMSTS.BFULFLG flag changes from 0 to 1
Header pattern match	REMSTS.HDFLG	REMINT.HDINT	When the REMSTS.HDFLG flag changes from 0 to 1 (When the header pattern is detected)
Data ‘0’ pattern or data ‘1’ pattern match	REMSTS.D0FLG, REMSTS.D1FLG	REMINT.DINT	• When the REMSTS.D0FLG flag changes from 0 to 1 (When the data ‘0’ pattern is detected) • When the REMSTS.D1FLG flag changes from 0 to 1 (When the data ‘1’ pattern is detected)
Special data pattern match	REMSTS.SDFLG	REMINT.SDINT	When the REMSTS.SDFLG flag changes from 0 to 1 (When the special data pattern is detected)

Table 46.6 Conditions for Generating an Interrupt Request for Each Interrupt Mode

	Interrupt Mode	
Item	**Normal Interrupt Mode**	**Sequential Interrupt Mode**
Bit setting	REMCON1.INTMD bit = 0	REMCON1.INTMD bit = 1
REMCI0 interrupt request generation condition	Satisfaction of any enabled interrupt source condition among the following seven leads to the generation of an interrupt request. • Compare match • Receive error • Data reception complete • Receive buffer full • Header pattern match • Data ‘0’ pattern or data ‘1’ pattern match • Special data pattern match	Satisfaction of all enabled interrupt source conditions among the following four leads to the generation of the interrupt request. • Compare match • Data reception complete • Header pattern match • Special data pattern match Satisfaction of any enabled interrupt source condition among the following three leads to the generation of an interrupt request. • Receive error • Receive buffer full • Data ‘0’ pattern or data ‘1’ pattern match

46.3.13 Data Reception in Low Power Consumption State

In this MCU, data can be received in a low power consumption state (sleep mode, all-module clock stop mode, software standby mode, or deep software standby mode).
To receive data in a low power consumption state, REMC communications should be set before transitioning to the state.

46.3.13.1 Using REMC Interrupt Request to Return from Low Power Consumption State

Power consumption while waiting for data reception can be reduced by using the REMC interrupt request to be output during data reception as the source for returning from the low power consumption state (see Figure 46.20). Pattern detection and compare function enable returning from the low power consumption state only when specified data is received.

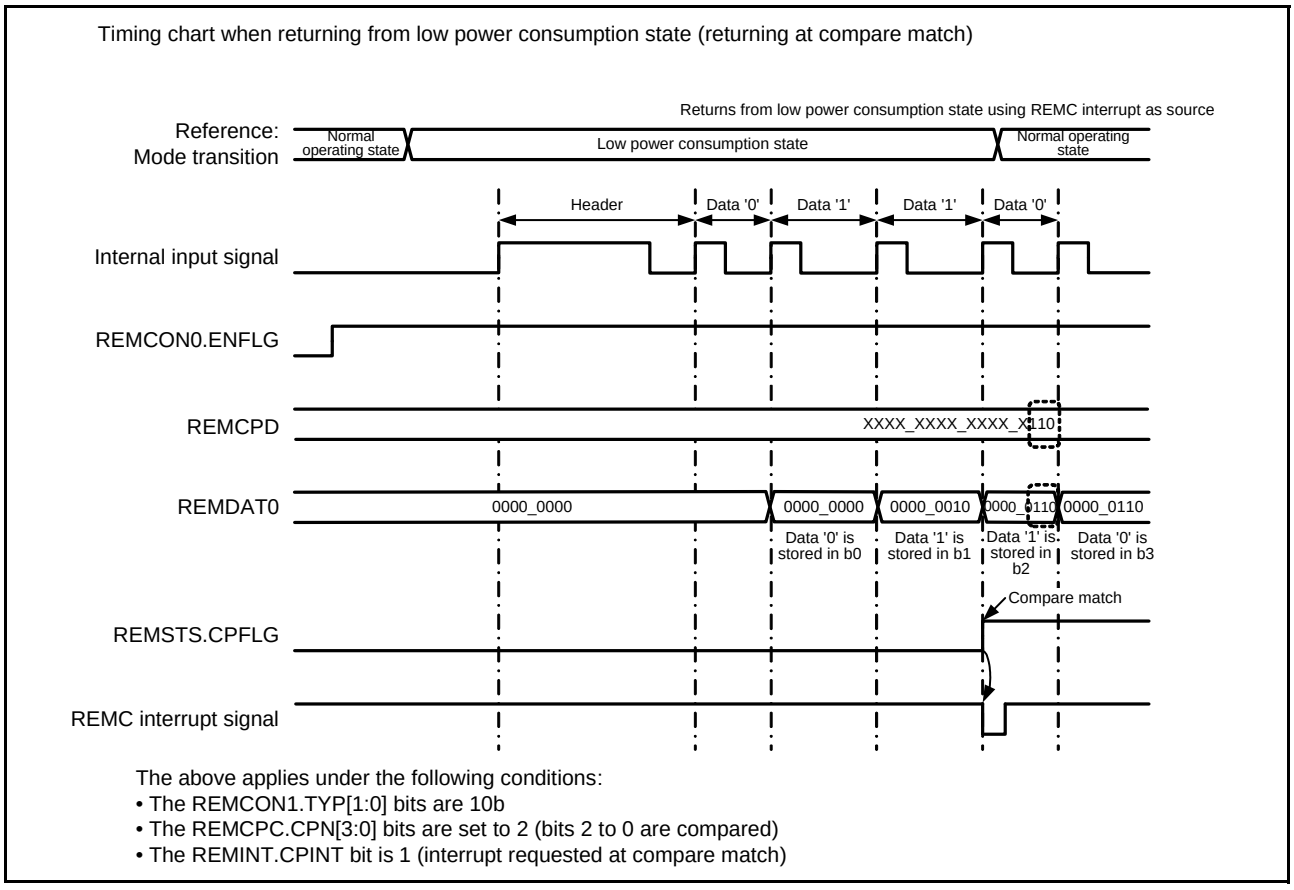


Figure 46.20 Using REMC Interrupt Request to Return from Low Power Consumption State (Normal Interrupt Mode)

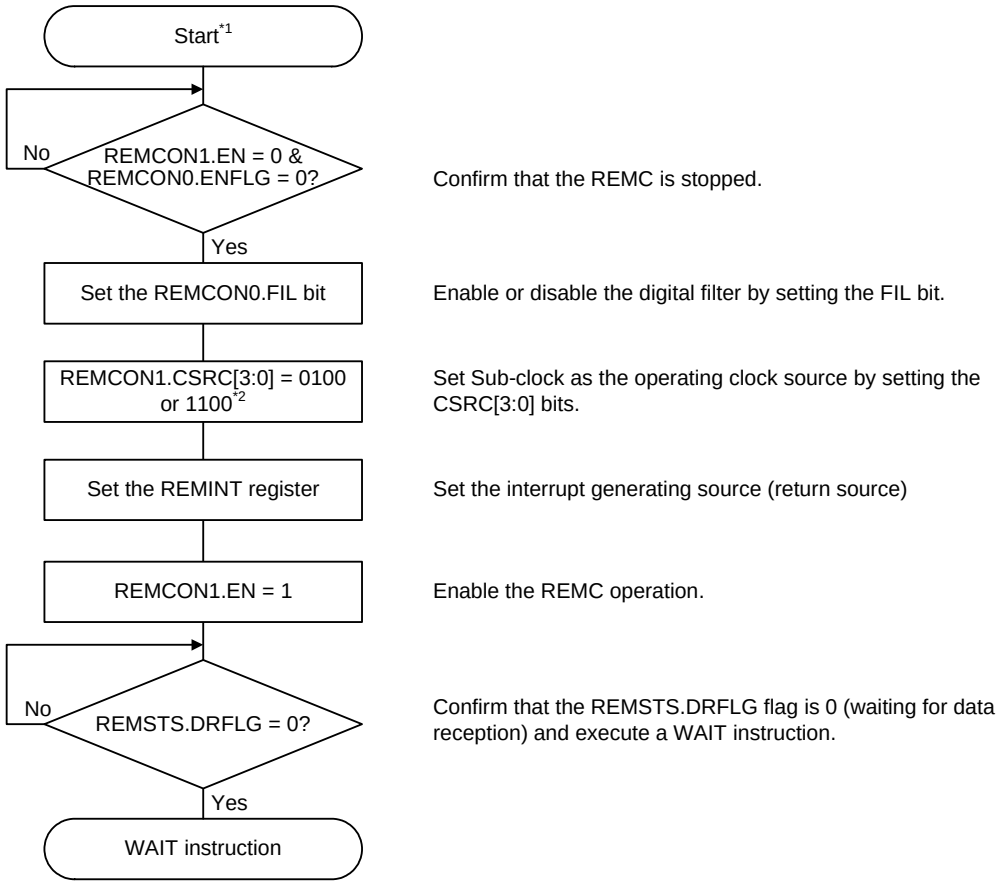
46.3.13.2 Data Reception in Software Standby Mode and Deep Software Standby Mode

Data can be received in software standby mode or deep software standby mode while the sub-clock is selected as the REMC operating clock.

When data is to be received in software standby mode or deep software standby mode, set the RCR3.RTCEN bit to 1 to select continuous supply of the sub-clock to the REMC as the operating clock in software standby mode or deep software standby mode. For details on how to supply the REMC operating clock, see section 46.3.4, Operating Clocks.

To return the chip from the software standby mode or deep software standby mode, select the conditions for generating REMC interrupt requests during data reception and the output conditions specified for the interrupt mode.

Pattern detection and comparison will only cause return from the software standby mode when the specified data is received. Figure 46.21 shows an example of a flowchart of the procedure for setting up data reception in software standby mode or deep software standby mode.



Note 1. Continuous supply of the sub-clock to the REMC is required in software standby mode or deep software standby mode. This is selected by setting the RCR3.RTCEN bit to 1.

Note 2. When the REMCON1.CSRC[3:0] bits are modified, re-setting of the REMCON0 register, REMCON1 register, REMCPC register, REMCPD register, REMPE register, and the respective data pattern width setting registers is required.

Figure 46.21 Flowchart for Setting Data Reception in Software Standby Mode or Deep Software Standby Mode

46.4 Usage Notes

46.4.1 Module Stop Function Setting

REMC operation can be disabled or enabled by setting the module stop control register. The REMC is stopped with the value after reset. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

46.4.2 Settings for Peripheral Module Clock and REMC Operating Clock

Set the peripheral module clock (PCLKB) frequency to higher than the REMC operating clock frequency.

46.4.3 Starting/Stopping Operation of Remote Control Signal Receiver

The REMCON1.EN bit controls starting/stopping of operation of the remote control signal receiver.
The REMCON0.ENFLG flag indicates that the operation is enabled or disabled. After the REMCON1.EN bit is set to 1 (operation enabled), it takes up to zero to one cycle of the operating clock before the REMC circuit starts operating and the REMCON0.ENFLG flag becomes 1. During this period, do not access the REMC related registers (listed in section 46.2.1 to section 46.2.20) except for the REMCON0.ENFLG flag.

46.4.4 Accessing Registers

Change the following registers only when the REMCON1.EN bit and REMCON0.ENFLG flag are both 0 (REMC is stopped)

- REMCON0 register
- REMCON1 register (except for bits 0 to 2)
- REMINT register (except for bits 2 and 5)
- REMCPC register
- REMCPD register
- Pattern width setting registers for header, data ‘0’, data ‘1’, and special data patterns
- Pattern end setting register

When rewriting the REMCON1.TYP[1:0] bits while the REMCON1.EN bit or REMCON0.ENFLG flag is 1 (REMC is operating), change the values of these bits one bit at a time. If the REMCON1.TYP[1:0] bits are rewritten when the REMCON0.INFLG flag changes, the signal captured into the remote control signal receiver may be undefined.
After 0 is written to bit 0 in the REMDAT0 or REMRBIT register or the REMSTS.BFULFLG flag, do not write 0 to the same bit again for two cycles of the operating clock. If 0 is written when the REMCON0.INFLG flag changes, the values of the REMDATj and REMRBIT registers and the REMSTS.BFULFLG flag may be undefined.

46.4.5 PMC0 Input Control

If the REMCON0.FILSEL, FIL, or INV bit is rewritten, the signal captured into the remote control signal receiver is undefined for three cycles of the digital filter sampling clock.

46.4.6 Notes on Changing the Operating Clock

When the REMCON1.CSRC[3:0] bits are rewritten, set the following registers again: REMCON0, REMCON1, REMINT, REMCPC, REMCPD, REMPE, and header, data ‘0’, data ‘1’, and special data pattern width setting registers.

46.4.7 Reading Registers

When the following registers are read while data changes, an undefined value may be read.

Flags in the REMCON0 and REMSTS registers (except for the REMSTS.DRFLG flag) and registers REMTIM, REMDAT0 to REMDAT7, and REMRBIT

Follow the procedures below to avoid reading an undefined value.

- Using an interrupt
  Set the REMINT.DRINT bit to 1 (data reception complete interrupt enabled) and read the registers within the REMC interrupt routine.
- Polling by a program 1
  Set the REMINT.DRINT bit to 1 (data reception complete interrupt enabled) and poll the ICU.IRn.IR flag by a program. Read the registers when the IF bit becomes 1 (interrupt request generated).
- Polling by a program 2
  (1) Poll the REMSTS.DRFLG flag.
  (2) When the REMSTS.DRFLG flag becomes 1, poll this flag until it becomes 0.
  (3) Read the necessary content of the registers when the REMSTS.DRFLG flag becomes 0.

47. Capacitive Touch Sensing Unit (CTSUa)

The capacitive touch sensing unit (CTSU) measures the electrostatic capacitance of the touch sensor. Changes in the electrostatic capacitance are determined by software, which enables the CTSU to detect whether a finger is in contact with the touch sensor. The electrode surface of the touch sensor is usually enclosed with a dielectric so that a finger does not come into contact with the electrode.

As shown in Figure 47.1, electrostatic capacitance (parasitic capacitance) exists between the electrode and the surrounding conductors. Because the human body is an electrical conductor, when a finger is placed close to the electrode, the value of electrostatic capacitance increases.

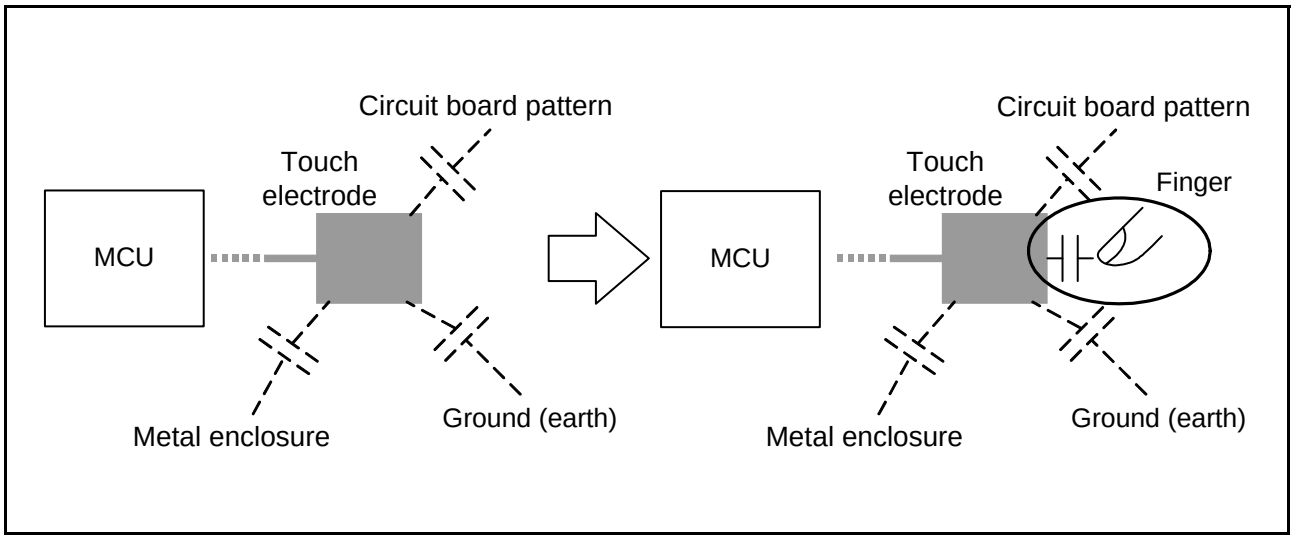


Figure 47.1 Increased Electrostatic Capacitance Due to Presence of Finger

Electrostatic capacitance is detected by the following methods: Self-capacitance and mutual capacitance.

In the self-capacitance method, the CTSU detects electrostatic capacitance generated between a finger and a single electrode. In the mutual capacitance method, two electrodes are used as a transmit electrode and a receive electrode, and the CTSU detects the change in the electrostatic capacitance generated between the two when a finger is placed close to them.

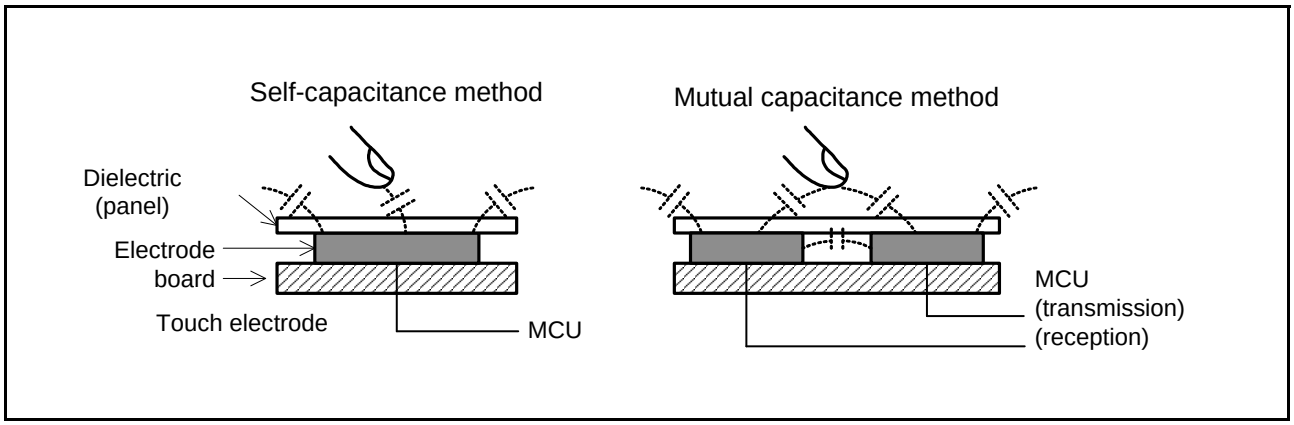


Figure 47.2 Self-Capacitance Method and Mutual Capacitance Method

Electrostatic capacitance is measured by counting a clock signal whose frequency changes according to the amount of charged/discharged current, for a specified period.

For details on the measurement principles of the CTSU, refer to section 47.3.1, Principles of Measurement Operation.

In this section, “PCLK” is used to refer to PCLKB.

47.1 Overview

Table 47.1 lists the specifications of the CTSU, and Figure 47.3 shows a block diagram of the CTSU.

Table 47.1 CTSU Specifications

Item		Description
Operating clock		PCLK, PCLK/2, or PCLK/4
Pins	TS0 to TS16	Electrostatic capacitance measurement pins (17 channels)
	TSCAP	LPF (low-pass filter) connection pin
Measurement modes	Self-capacitance single scan mode	Electrostatic capacitance on a channel is measured by the self-capacitance method.
	Self-capacitance multi-scan mode	Electrostatic capacitance on multiple channels is measured successively by the self-capacitance method.
	Mutual capacitance full scan mode	Electrostatic capacitance on multiple channels is measured successively by mutual capacitance.
Noise prevention		Synchronous noise prevention, high-pass noise prevention
Measurement start conditions		• Software trigger • External trigger (event input from the event link controller (ELC))

As shown in Figure 47.3, the CTSU consists of the status control block, trigger control block, clock control block, channel control block, port control block, sensor drive pulse generator, measurement block, interrupt block, and control registers.

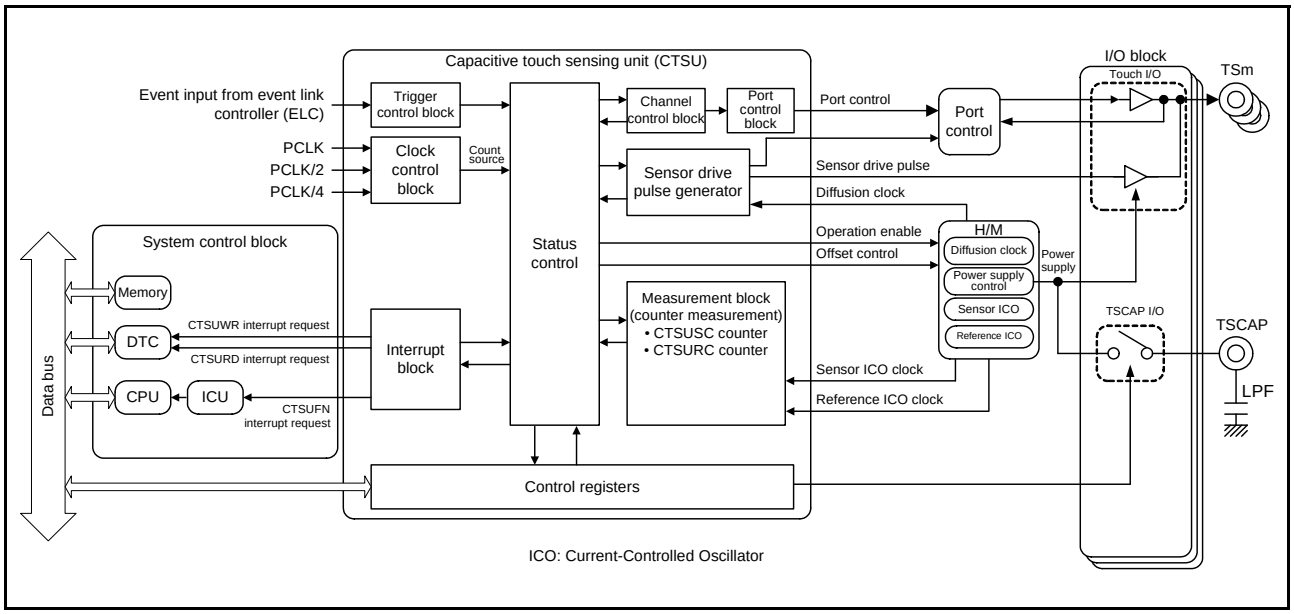


Figure 47.3 CTSU Block Diagram (m = 0 to 16)

Table 47.2 CTSU Pin Configuration

Pin Name	I/O	Function
TS0 to TS16	I/O	Electrostatic capacitive measurement pins (touch pins)
TSCAP	—	LPF connection pin

47.2 Register Descriptions

47.2.1 CTSU Control Register 0 (CTSUCR0)

Address(es): CTSU.CTSUCR0 000A 0900h

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUTXVSEL	—	—	CTSUINIT	CTSUIOC	CTSUSNZ	CTSUCAP	CTSUSTRT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CTSUSTRT	CTSU Measurement Operation Start	0: Measurement operation stops 1: Measurement operation starts	R/W
b1	CTSUCAP	CTSU Measurement Operation Start Trigger Select	0: Software trigger 1: External trigger	R/W
b2	CTSUSNZ	CTSU Wait State Power-Saving Enable	0: Power-saving function during wait state is disabled 1: Power-saving function during wait state is enabled	R/W
b3	CTSUIOC	CTSU Transmit Pin Control	0: The TS pins are driven low 1: The TS pins are driven high	R/W
b4	CTSUINIT	CTSU Control Block Initialization	Writing 1 to this bit initializes the CTSU control block and registers*1. This bit is read as 0.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	CTSUTXVSEL	CTSU Transmission Power Supply Select*2	0: VCC selected 1: Internal logic power supply selected	R/W

Note 1. The CTSUSC, CTSURC, CTSUMCH0, CTSUMCH1, and CTSUST registers are initialized.
Note 2. This bit can be set to 1 only in mutual capacitance full scan mode. Otherwise (in self-capacitance single scan mode or self-capacitance multi-scan mode), set this bit to 0.

The CTSUCAP, CTSUSNZ and CTSUTXVSEL bits should be set when the CTSUSTRT bit is 0. These bits can be set at the same time as starting measurement operation.

CTSUSTRT Bit (CTSU Measurement Operation Start)

This bit specifies whether CTSU operation starts or stops.
When the CTSUCAP bit is 0 (software trigger), measurement is started by writing 1 to the CTSUSTRT bit, and the CTSUSTRT bit becomes 0 when measurement is finished.
When the CTSUCAP bit is 1 (external trigger), the CTSU waits for an external trigger by writing 1 to the CTSUSTRT bit, and measurement is started at the rising edge of the external trigger. When measurement is finished, the CTSU waits for the next external trigger and operation is continued.
Table 47.3 lists the CTSU states.

Table 47.3 CTSU States

CTSUSTRT Bit	CTSUCAP Bit	CTSU State
0	0	Stopped
0	1	Stopped
1	0	During measurement
1	1	During measurement/wait for an external trigger*1

Note 1. The state can be read from the CTSUST.CTSUSTC[2:0] flags.
During measurement: CTSUST.CTSUSTC[2:0] flags ≠ 000b
Wait for an external trigger: CTSUST.CTSUSTC[2:0] flags = 000b

If the CTSUSTRT bit is set to 1 when the CTSUSTRT bit is 1, writing is ignored and operation is continued.

To forcibly stop operation (forced stop) when the CTSUSTRT bit is 1, set the CTSUSTRT bit to 0 and the CTSUINIT bit to 1 simultaneously.

CTSUCAP Bit (CTSU Measurement Operation Start Trigger Select)

This bit specifies the measurement start condition. For details, see the description of the CTSUSTRT bit.

CTSUSNZ Bit (CTSU Wait State Power-Saving Enable)

This bit enables or disables power-saving operation during a wait state. This bit can also be used to suspend the CTSU power supply, which decreases power consumption during the wait state.
In the suspended state, the CTSU power supply is turned off while the external TSCAP is still charged after the CTSU power supply has been turned on and the TSCAP has been charged.

Table 47.4 CTSU Power Supply State Control

CTSUCR1.CTSUPON Bit	CTSUSNZ Bit	CTSUCAP Bit	CTSUSTRT Bit	CTSU Power Supply State
0	0	0	0	Stopped
1	0	—	—	Operating
1	1	0	0	Suspended

Note: Settings other than the above are prohibited.

To start measurement from the suspended state, set the CTSUSNZ bit to 0 and wait for 16 µs before setting the CTSUSTRT bit to 1. To set the suspended state after measurement is finished, set the CTSUSNZ bit to 1.

CTSUIOC Bit (CTSU Transmit Pin Control)

This bit selects the logic level of the TS pin when the CTSUERRS.CTSUTSOD bit is set to 1.
This bit setting is ignored when the CTSUTSOD bit is 0.

CTSUINIT Bit (CTSU Control Block Initialization)

The internal control registers can be initialized by writing 1 to this bit. To forcibly stop the current operation, set the CTSUSTRT bit to 0 and the CTSUINIT bit to 1 simultaneously. In this case, the operation is stopped and the internal control registers are initialized.
Do not write 1 to the CTSUINIT bit at the same time as setting the CTSUSTRT bit to 1 (CTSU operation starts).

CTSUTXVSEL Bit (CTSU Transmission Power Supply Select)

This bit is used to switch the power supply for the transmit buffer in mutual capacitance full scan mode. Set this bit to 0 for any other mode.
This bit switches the power supply for touch I/O which is set for transmission by the CTSUCHTRCn registers.
Table 47.5 lists the power supply for TSm pin.
When the VCC voltage fluctuates greatly due to the switching of the output buffer, switching to the internal logic power supply can reduce the effect on the voltage fluctuation.

Table 47.5 Power Supply for TSm Pin

Setting of CTSUCHTRCn Registers	CTSUTXVSEL Bit	Power Supply for TSm Pin
0 (reception)	0 (VCC)	VCC
	1 (internal logic power supply)	
1 (transmission)	0 (VCC)	
	1 (internal logic power supply)	Internal logic power supply

47.2.2 CTSU Control Register 1 (CTSUCR1)

Address(es): CTSU.CTSUCR1 000A 0901h

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUMD[1:0]		CTSUCLK[1:0]		CTSUATUNE1	—	CTSUCSW	CTSUPON
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CTSUPON	CTSU Power Supply Enable	0: Powered off 1: Powered on	R/W
b1	CTSUCSW	CTSU LPF Capacitance Charging Control	0: Capacitance switch turned off 1: Capacitance switch turned on	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	CTSUATUNE1	CTSU Power Supply Capacity Adjustment	0: Normal output 1: High-current output	R/W
b5, b4	CTSUCLK[1:0]	CTSU Operating Clock Select	b5 b4 0 0: PCLK 0 1: PCLK/2 (PCLK divided by 2) 1 0: PCLK/4 (PCLK divided by 4) 1 1: Setting prohibited	R/W
b7, b6	CTSUMD[1:0]	CTSU Measurement Mode Select	b7 b6 0 0: Self-capacitance single scan mode 0 1: Self-capacitance multi-scan mode 1 0: Setting prohibited 1 1: Mutual capacitance full scan mode	R/W

The CTSUCR1 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUPON Bit (CTSU Power Supply Enable)

This bit controls power supply to the CTSU. Set the CTSUPON and CTSUCSW bits to the same value at the same time.

CTSUCSW Bit (CTSU LPF Capacitance Charging Control)

This bit controls charging of the LPF capacitor connected to the TSCAP pin (turning on/off of the capacitance switch). After the capacitance switch is turned on, wait until the capacitance connected to the TSCAP pin is charged for the specified time before starting measurement (CTSUCR0.CTSUSTRT = 1). Prior to measurement, use an I/O port to output a low level to the TSCAP pin, and discharge the LPF capacitance that has been already charged. Set the CTSUPON and CTSUCSW bits to the same value at the same time.

CTSUATUNE1 Bit (CTSU Power Supply Capacity Adjustment)

This bit sets the capacity of the CTSU power supply. Normally, the value of this bit should be set to 0.

CTSUCLK[1:0] Bits (CTSU Operating Clock Select)

These bits select the operating clock.

CTSUMD[1:0] Bits (CTSU Measurement Mode Select)

These bits set the measurement mode. For details, refer to section 47.3.2, Measurement Modes.

47.2.3 CTSU Synchronous Noise Reduction Setting Register (CTSUSDPRS)

Address(es): CTSU.CTSUSDPRS 000A 0902h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	CTSUSOFF	CTSUPRMODE[1:0]		CTSUPRRATIO[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CTSUPRRATIO[3:0]	CTSU Measurement Time and Pulse Count Adjustment	Recommended setting value: 3 (0011b)	R/W
b5, b4	CTSUPRMODE[1:0]	CTSU Base Period and Pulse Count Setting	b5 b4 0 0: 510 pulses 0 1: 126 pulses 1 0: 62 pulses (recommended setting value) 1 1: Setting prohibited	R/W
b6	CTSUSOFF	CTSU High-Pass Noise Reduction Function Off Setting	0: High-pass noise reduction function turned on 1: High-pass noise reduction function turned off	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The CTSUSDPRS register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUPRRATIO[3:0] Bits (CTSU Measurement Time and Pulse Count Adjustment)

These bits are used to determine the measurement time and the number of measurement pulses. These are calculated by the following formula. The number of base pulses is determined by setting the CTSUPRMODE[1:0] bits.

Number of measurement pulses = number of base pulses × (CTSUPRRATIO[3:0] bits + 1)

Measurement time = (number of base pulses × (CTSUPRRATIO[3:0] bits + 1) + (number of base pulses – 2) × 0.25) × base clock cycle

Note: For details on the base clock cycle, refer to section 47.2.17, CTSU Sensor Offset Register 1 (CTSUSO1).

CTSUPRMODE[1:0] Bits (CTSU Base Period and Pulse Count Setting)

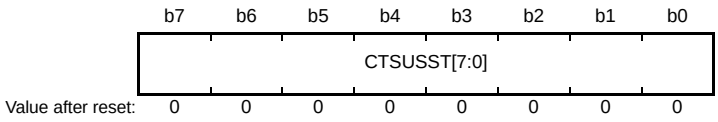
These bits select the number of base pulses during measurement.

CTSUSOFF Bit (CTSU High-Pass Noise Reduction Function Off Setting)

This bit turns on or off the function for reducing high-pass noise. Set this bit to 1 when turning off the high-pass noise reduction function.

47.2.4 CTSU Sensor Stabilization Wait Control Register (CTSUSST)

Address(es): CTSU.CTSUSST 000A 0903h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	CTSUSST[7:0]	CTSU Sensor Stabilization Wait Control	The value of these bits should be fixed to 00010000b.	R/W

The CTSUSST register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUSST[7:0] Bits (CTSU Sensor Stabilization Wait Control)

These bits set the stabilization wait time for the TSCAP pin voltage. The value of these bits should be fixed to 00010000b. If these bits are not set, the TSCAP voltage becomes unstable at the start of measurement, and the CTSU is unable to obtain correct touch measurement results.

47.2.5 CTSU Measurement Channel Register 0 (CTSUMCH0)

Address(es): CTSU.CTSUMCH0 000A 0904h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CTSUMCH0[4:0]				
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	CTSUMCH0[4:0]	CTSU Measurement Channel 0	• In self-capacitance single scan b4 b0 0 0 0 0 0: TS0 0 0 0 0 1: TS1 : : 0 1 1 1 1: TS15 1 0 0 0 0: TS16 Other than above: Starting measurement operation (CTSUCR0.CTSUSTRT bit = 1) is prohibited after these bits are set. • In other measurement modes b4 b0 0 0 0 0 0: TS0 0 0 0 0 1: TS1 : : 0 1 1 1 1: TS15 1 0 0 0 0: TS16 1 1 1 1 1: Measurement is stopped	R/W*1
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to these bits is enabled only in self-capacitance single scan mode (CTSUCR1.CTSUMD[1:0] bits = 00b).

The CTSUMCH0 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUMCH0[4:0] Bits (CTSU Measurement Channel 0)

These bits set the channel to be measured in self-capacitance single scan mode, and indicate the receive channel that is being measured in other modes.

Set only enabled channels (00000b to 10000b) when setting channels in self-capacitance single scan mode. In other modes, writing to these bits has no effect.

47.2.6 CTSU Measurement Channel Register 1 (CTSUMCH1)

Address(es): CTSU.CTSUMCH1 000A 0905h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	CTSUMCH1[4:0]				
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	CTSUMCH1[4:0]	CTSU Measurement Channel 1	b4 b0 0 0 0 0 0: TS0 0 0 0 0 1: TS1 0 0 0 1 0: TS2 0 0 0 1 1: TS3 0 0 1 0 0: TS4 0 0 1 0 1: TS5 0 0 1 1 0: TS6 0 0 1 1 1: TS7 0 1 0 0 0: TS8 0 1 0 0 1: TS9 0 1 0 1 0: TS10 0 1 0 1 1: TS11 0 1 1 0 0: TS12 0 1 1 0 1: TS13 0 1 1 1 0: TS14 0 1 1 1 1: TS15 1 0 0 0 0: TS16 1 1 1 1 1: Measurement is stopped	R
b7 to b5	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R

CTSUMCH1[4:0] Bits (CTSU Measurement Channel 1)

These bits indicate the transmit channel that is being measured in full scan mode. The value of these bits is 11111b while measurement is stopped or in self-capacitance single scan mode and multi-scan mode.

47.2.7 CTSU Channel Enable Control Register 0 (CTSUCHAC0)

Address(es): CTSU.CTSUCHAC0 000A 0906h

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUCHAC07	CTSUCHAC06	CTSUCHAC05	CTSUCHAC04	CTSUCHAC03	CTSUCHAC02	CTSUCHAC01	CTSUCHAC00
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CTSUCHAC00	CTSU Channel 0 Enable Control	0: Not measurement target 1: Measurement target	R/W
b1	CTSUCHAC01	CTSU Channel 1 Enable Control		R/W
b2	CTSUCHAC02	CTSU Channel 2 Enable Control		R/W
b3	CTSUCHAC03	CTSU Channel 3 Enable Control		R/W
b4	CTSUCHAC04	CTSU Channel 4 Enable Control		R/W
b5	CTSUCHAC05	CTSU Channel 5 Enable Control		R/W
b6	CTSUCHAC06	CTSU Channel 6 Enable Control		R/W
b7	CTSUCHAC07	CTSU Channel 7 Enable Control		R/W

The CTSUCHAC0 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHAC0j Bit (CTSU Channel m Enable Control) (j = 0 to 7; m = 0 to 7)

This bit sets the pin (for receive and transmit) whose electrostatic capacitance is to be measured.

47.2.8 CTSU Channel Enable Control Register 1 (CTSUCHAC1)

Address(es): CTSU.CTSUCHAC1 000A 0907h

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUCHAC17	CTSUCHAC16	CTSUCHAC15	CTSUCHAC14	CTSUCHAC13	CTSUCHAC12	CTSUCHAC11	CTSUCHAC10
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CTSUCHAC10	CTSU Channel 8 Enable Control	0: Not measurement target 1: Measurement target	R/W
b1	CTSUCHAC11	CTSU Channel 9 Enable Control		R/W
b2	CTSUCHAC12	CTSU Channel 10 Enable Control		R/W
b3	CTSUCHAC13	CTSU Channel 11 Enable Control		R/W
b4	CTSUCHAC14	CTSU Channel 12 Enable Control		R/W
b5	CTSUCHAC15	CTSU Channel 13 Enable Control		R/W
b6	CTSUCHAC16	CTSU Channel 14 Enable Control		R/W
b7	CTSUCHAC17	CTSU Channel 15 Enable Control		R/W

The CTSUCHAC1 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHAC1j Bit (CTSU Channel m Enable Control) (j = 0 to 7; m = 8 to 15)

This bit sets the pin (for receive and transmit) whose electrostatic capacitance is to be measured.

47.2.9 CTSU Channel Enable Control Register 2 (CTSUCHAC2)

Address(es): CTSU.CTSUCHAC2 000A 0908h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CTSUCHAC20
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CTSUCHAC20	CTSU Channel 16 Enable Control	0: Not measurement target 1: Measurement target	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The CTSUCHAC2 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHAC20 Bit (CTSU Channel 16 Enable Control)

This bit sets the pin (for receive and transmit) whose electrostatic capacitance is to be measured.

47.2.10 CTSU Channel Transmit/Receive Control Register 0 (CTSUCHTRC0)

Address(es): CTSU.CTSUCHTRC0 000A 090Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUCHTRC07	CTSUCHTRC06	CTSUCHTRC05	CTSUCHTRC04	CTSUCHTRC03	CTSUCHTRC02	CTSUCHTRC01	CTSUCHTRC00
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CTSUCHTRC00	CTSU Channel 0 Transmit/Receive Control	0: Reception 1: Transmission	R/W
b1	CTSUCHTRC01	CTSU Channel 1 Transmit/Receive Control		R/W
b2	CTSUCHTRC02	CTSU Channel 2 Transmit/Receive Control		R/W
b3	CTSUCHTRC03	CTSU Channel 3 Transmit/Receive Control		R/W
b4	CTSUCHTRC04	CTSU Channel 4 Transmit/Receive Control		R/W
b5	CTSUCHTRC05	CTSU Channel 5 Transmit/Receive Control		R/W
b6	CTSUCHTRC06	CTSU Channel 6 Transmit/Receive Control		R/W
b7	CTSUCHTRC07	CTSU Channel 7 Transmit/Receive Control		R/W

The CTSUCHTRC0 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHTRC0j Bit (CTSU Channel m Transmit/Receive Control) (j = 0 to 7; m = 0 to 7)

This bit allocates reception or transmission to the corresponding TSm pin in full scan mode. Set this bit to 0 in self-capacitance single scan mode and multi-scan mode.

47.2.11 CTSU Channel Transmit/Receive Control Register 1 (CTSUCHTRC1)

Address(es): CTSU.CTSUCHTRC1 000A 090Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUCHTRC17	CTSUCHTRC16	CTSUCHTRC15	CTSUCHTRC14	CTSUCHTRC13	CTSUCHTRC12	CTSUCHTRC11	CTSUCHTRC10
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CTSUCHTRC10	CTSU Channel 8 Transmit/Receive Control	0: Reception 1: Transmission	R/W
b1	CTSUCHTRC11	CTSU Channel 9 Transmit/Receive Control		R/W
b2	CTSUCHTRC12	CTSU Channel 10 Transmit/Receive Control		R/W
b3	CTSUCHTRC13	CTSU Channel 11 Transmit/Receive Control		R/W
b4	CTSUCHTRC14	CTSU Channel 12 Transmit/Receive Control		R/W
b5	CTSUCHTRC15	CTSU Channel 13 Transmit/Receive Control		R/W
b6	CTSUCHTRC16	CTSU Channel 14 Transmit/Receive Control		R/W
b7	CTSUCHTRC17	CTSU Channel 15 Transmit/Receive Control		R/W

The CTSUCHTRC1 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHTRC1j Bit (CTSU Channel m Transmit/Receive Control) (j = 0 to 7; m = 8 to 15)

This bit allocates reception or transmission to the corresponding TSm pin in full scan mode. Set this bit to 0 in self-capacitance single scan mode and multi-scan mode.

47.2.12 CTSU Channel Transmit/Receive Control Register 2 (CTSUCHTRC2)

Address(es): CTSU.CTSUCHTRC2 000A 090Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CTSUCHTRC20
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CTSUCHTRC20	CTSU Channel 16 Transmit/Receive Control	0: Reception 1: Transmission	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The CTSUCHTRC2 register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUCHTRC20 Bit (CTSU Channel 16 Transmit/Receive Control)

This bit allocates reception or transmission to the corresponding TS16 pin in full scan mode. Set this bit to 0 in self-capacitance single scan mode and multi-scan mode.

47.2.13 CTSU High-Pass Noise Reduction Control Register (CTSUDCLKC)

Address(es): CTSU.CTSUDCLKC 000A 0910h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CTSUSSMOD[1:0]	CTSU Diffusion Clock Mode Select	These bits should be set to 00b.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	CTSUSSCNT[1:0]	CTSU Diffusion Clock Control	These bits should be set to 11b.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The CTSUDCLKC register should be set when the CTSUCR0.CTSUSTRT bit is 0.

CTSUSSMOD[1:0] Bits (CTSU Diffusion Clock Mode Select)

These bits set the mode of the spectrum diffusion clock for high-pass noise reduction. When using the high-pass function, the value of these bits should be fixed to 00b. If these bits are not set, the effect of high-pass noise reduction cannot be correctly obtained.

CTSUSSCNT[1:0] Bits (CTSU Diffusion Clock Control)

These bits adjust the spectrum diffusion amount to reduce high-pass noise. When using the high-pass noise reduction function, the value of these bits should be fixed to 11b. If these bits are not set, touch measurement may not be correctly performed.

47.2.14 CTSU Status Register (CTSUST)

Address(es): CTSU.CTSUST 000A 0911h

	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUPS	CTSUROVF	CTSUSOVF	CTSUDTSR	—	CTSUSTC[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CTSUSTC[2:0]	CTSU Measurement Status Counter	b2 b0 0 0 0: Status 0 0 0 1: Status 1 0 1 0: Status 2 0 1 1: Status 3 1 0 0: Status 4 1 0 1: Status 5	R
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	CTSUDTSR	CTSU Data Transfer Status Flag	0: Measurement result has been read 1: Measurement result has not been read	R
b5	CTSUSOVF	CTSU Sensor Counter Overflow Flag	0: No overflow 1: An overflow	R/W
b6	CTSUROVF	CTSU Reference Counter Overflow Flag	0: No overflow 1: An overflow	R/W
b7	CTSUPS	CTSU Mutual Capacitance Status Flag	0: First measurement 1: Second measurement	R

When using the CTSUCR0.CTSUINIT bit to clear an overflow flag, make sure that the CTSUCR0.CTSUSTRT bit is 0.

CTSUSTC[2:0] Flags (CTSU Measurement Status Counter)

These counters indicate the current measurement status. For details on each status, refer to section 47.3.2.2, Status Counter.

CTSUDTSR Flag (CTSU Data Transfer Status Flag)

This flag indicates whether the measurement result stored in the sensor counter and the reference counter has been read. This flag is set to 1 when measurement is completed; 0 when the reference counter is read by software or the DTC. This flag is also cleared using the CTSUCR0.CTSUINIT bit.

CTSUSOVF Flag (CTSU Sensor Counter Overflow Flag)

This flag indicates whether the sensor counter has overflowed. FFFFh can be read as the measurement result (CTSUSC counter) when an overflow has occurred.

Even if an overflow occurs, measurement processing is continued until the set period.

No interrupt is generated even when an overflow occurs. To determine the channel on which the overflow has occurred, read the measurement result of each channel after measurement is completed (after a measurement end interrupt is generated).

This flag is cleared when 0 is written after 1 is read by software. This flag is also cleared using the CTSUCR0.CTSUINIT bit.

CTSUROVF Flag (CTSU Reference Counter Overflow Flag)

This flag indicates whether the reference counter has overflowed. FFFFh can be read as the measurement result (CTSURC counter) when an overflow has occurred.

Even if an overflow occurs, measurement processing is continued until the set period.

No interrupt is generated even when an overflow occurs. To determine the channel on which the overflow has occurred, read the measurement result of each channel after measurement is completed (after a measurement end interrupt is generated).
This flag is cleared when 0 is written after 1 is read by software. This flag is also cleared using the CTSUCR0.CTSUINIT bit.

CTSUPS Flag (CTSU Mutual Capacitance Status Flag)

This flag indicates whether the measurement is the first or second of two measurements for each channel in mutual capacitance full scan mode (CTSUCR1.CTSUMD[1:0] bits = 11b).
This flag indicates 0 while measurement is stopped or in other measurement modes.

47.2.15 CTSU High-Pass Noise Reduction Spectrum Diffusion Control Register (CTSUSSC)

Address(es): CTSU.CTSUSSC 000A 0912h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CTSUSSDIV[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	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	CTSUSSDIV[3:0]	CTSU Spectrum Diffusion Frequency Division Setting	These bits specify the spectrum diffusion frequency division setting according to the base clock frequency division setting.	R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CTSUSSDIV[3:0] Bits (CTSU Spectrum Diffusion Frequency Division Setting)

These bits specify the spectrum diffusion frequency division setting according to the base clock frequency division setting. See the relationship between base clock frequencies and the CTSUSSDIV[3:0] bits settings in Table 47.6, for setting the value of these bits.

Table 47.6 Relationship between Base Clock Frequencies and the CTSUSSDIV[3:0] Bits Settings

Base Clock Frequency fb (MHz)	CTSUSSDIV[3:0] Bits Setting
4.00 ≤ fb	0000b
2.00 ≤ fb < 4.00	0001b
1.33 ≤ fb < 2.00	0010b
1.00 ≤ fb < 1.33	0011b
0.80 ≤ fb < 1.00	0100b
0.67 ≤ fb < 0.80	0101b
0.57 ≤ fb < 0.67	0110b
0.50 ≤ fb < 0.57	0111b
0.44 ≤ fb < 0.50	1000b
0.40 ≤ fb < 0.44	1001b
0.36 ≤ fb < 0.40	1010b
0.33 ≤ fb < 0.36	1011b
0.31 ≤ fb < 0.33	1100b
0.29 ≤ fb < 0.31	1101b
0.27 ≤ fb < 0.29	1110b
fb < 0.27	1111b

47.2.16 CTSU Sensor Offset Register 0 (CTSUSO0)

Address(es): CTSU.CTSUSO0 000A 0914h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUSNUM[5:0]						CTSUSO[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b9 to b0	CTSUSO[9:0]	CTSU Sensor Offset Adjustment	b9 b0 0 0 0 0 0 0 0 0 0 0: Current offset amount is 0 0 0 0 0 0 0 0 0 0 1: Current offset amount is 1 0 0 0 0 0 0 0 0 1 0: Current offset amount is 2 : : 1 1 1 1 1 1 1 1 1 0: Current offset amount is 1022 1 1 1 1 1 1 1 1 1 1: Current offset amount is maximum	R/W
b15 to b10	CTSUSNUM[5:0]	CTSU Measurement Count Setting	These bits set the number of measurements.	R/W

CTSUSO[9:0] Bits (CTSU Sensor Offset Adjustment)

These control bits adjust the input current offset of the sensor ICO. These bits are used to offset the sensor ICO input current generated from electrostatic capacitance while the electrode is not being touched during touch measurement, thus preventing overflow of the CTSU sensor counter.

Make settings for the TS pin that is to be measured next after a CTSUWR interrupt is generated.

CTSUSNUM[5:0] Bits (CTSU Measurement Count Setting)

These bits set how many times the number of measurement pulses specified by the CTSUSDPRS.CTSUPRRATIO[3:0] and CTSUSDPRS.CTSUPRMODE[1:0] bits is repeated in the measurement time. The number of measurement pulses is repeated (CTSUSNUM[5:0] bits + 1) times.

Make settings for the TS pin that is to be measured next after a CTSUWR interrupt is generated.

47.2.17 CTSU Sensor Offset Register 1 (CTSUSO1)

Address(es): CTSU.CTSUSO1 000A 0916h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	CTSUICOG[1:0]		CTSUSDPA[4:0]					CTSURICOA[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	CTSURICOA[7:0]	CTSU Reference ICO Current Adjustment	b7 b0 0 0 0 0 0 0 0 0: Input current amount 0 0 0 0 0 0 0 0 1: Input current amount 1 0 0 0 0 0 0 1 0: Input current amount 2 : : 1 1 1 1 1 1 1 0: Input current amount 254 1 1 1 1 1 1 1 1: Input current amount maximum	R/W
b12 to b8	CTSUSDPA[4:0]	CTSU Base Clock Setting	b12 b8 0 0 0 0 0: Operating clock divided by 2*1 0 0 0 0 1: Operating clock divided by 4 0 0 0 1 0: Operating clock divided by 6 0 0 0 1 1: Operating clock divided by 8 0 0 1 0 0: Operating clock divided by 10 0 0 1 0 1: Operating clock divided by 12 0 0 1 1 0: Operating clock divided by 14 0 0 1 1 1: Operating clock divided by 16 0 1 0 0 0: Operating clock divided by 18 0 1 0 0 1: Operating clock divided by 20 0 1 0 1 0: Operating clock divided by 22 0 1 0 1 1: Operating clock divided by 24 0 1 1 0 0: Operating clock divided by 26 0 1 1 0 1: Operating clock divided by 28 0 1 1 1 0: Operating clock divided by 30 0 1 1 1 1: Operating clock divided by 32 1 0 0 0 0: Operating clock divided by 34 1 0 0 0 1: Operating clock divided by 36 1 0 0 1 0: Operating clock divided by 38 1 0 0 1 1: Operating clock divided by 40 1 0 1 0 0: Operating clock divided by 42 1 0 1 0 1: Operating clock divided by 44 1 0 1 1 0: Operating clock divided by 46 1 0 1 1 1: Operating clock divided by 48 1 1 0 0 0: Operating clock divided by 50 1 1 0 0 1: Operating clock divided by 52 1 1 0 1 0: Operating clock divided by 54 1 1 0 1 1: Operating clock divided by 56 1 1 1 0 0: Operating clock divided by 58 1 1 1 0 1: Operating clock divided by 60 1 1 1 1 0: Operating clock divided by 62 1 1 1 1 1: Operating clock divided by 64	R/W
b14, b13	CTSUICOG[1:0]	CTSU ICO Gain Adjustment	b14 b13 0 0: 100% gain 0 1: 66% gain 1 0: 50% gain 1 1: 40% gain	R/W
b15	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. The CTSUSDPA[4:0] bits should not be set to 00000b while the high-pass noise reduction function is turned off (CTSUSDPRS.CTSUSOFF bit = 1) in mutual capacitance full scan mode (CTSUCR1.CTSUMD[1:0] bits = 11b).

Write first to the CTSUSSC register, then CTSUSO0 register, and then CTSUSO1 register after a CTSUWR interrupt is generated. Write operation to the CTSUSO1 register causes a transition to Status 3. Thus, set all the bits in a single setting when writing to the CTSUSO1 register.

CTSURICOA[7:0] Bits (CTSU Reference ICO Current Adjustment)

These bits adjust the oscillation frequency using the input current of the reference ICO.

CTSUSDPA[4:0] Bits (CTSU Base Clock Setting)

These bits are used to generate a base clock used as the source for the sensor drive pulse by dividing the operating clock. For details on the setting procedure, refer to section 47.3.2.1, Initial Setting Flowchart.

CTSUICOG[1:0] Bits (CTSU ICO Gain Adjustment)

These bits adjust the output frequency gain of the sensor ICO and the reference ICO. If changes in the capacitance between when the electrode is touched and when it is not touched greatly exceed the dynamic range of the sensor ICO, set the gain adjustment bits to adjust the gain appropriately.

47.2.18 CTSU Sensor Counter (CTSUSC)

Address(es): CTSU.CTSUSC 000A 0918h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUSC[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	CTSUSC[15:0]	CTSU Sensor Counter	These bits indicate FFFFh when an overflow occurs.	R

Read first from the CTSUSC counter and then the CTSURC counter after a CTSURD interrupt is generated.

CTSUSC[15:0] Bits (CTSU Sensor Counter)

These bits are configured as an increment counter that counts the sensor ICO clock.
Read these bits after a CTSURD interrupt is generated. After the CTSURC counter is read, these bits are cleared immediately before the CTSU measurement status counter value changes to Status 4 (the CTSUST.CTSUSTC[2:0] flags changes to 100b) in the next measurement. These bits are also cleared using the CTSUCR0.CTSUINIT bit.

47.2.19 CTSU Reference Counter (CTSURC)

Address(es): CTSU.CTSURC 000A 091Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CTSURC[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	CTSURC[15:0]	CTSU Reference Counter	These bits indicate FFFFh when an overflow occurs.	R

Read first from the CTSUSC counter and then the CTSURC counter after a CTSURD interrupt is generated.
Even when the stabilization time specified for Status 3 has elapsed, if the CTSURC counter is not read, Status 3 continues until the counter is read.

CTSURC[15:0] Bits (CTSU Reference Counter)

These bits are configured as an increment counter that counts the reference ICO clock.
The reference ICO is used to optimize touch measurement performed using the sensor ICO. There is some deviation depending on the internal sensor ICO and the reference ICO in the CTSU, but both ICOs have almost the same characteristics, and the dynamic range and the current to frequency characteristics are almost the same. The range of current amount that can be set by the reference ICO current adjustment bits is about the same as the range of both ICOs, and the current amount input to the sensor ICO must be within this dynamic range. First, use the reference ICO to check the differences between the ICOs and measure the current to oscillation frequency characteristics. Since the reference ICO oscillation frequency can be obtained from the reference ICO counter, the ICO oscillation frequency (counter value/ measurement time) for the input current amount can be measured by setting the value in the reference ICO current adjustment bits and measuring the reference ICO counter. The reference ICO counter value measured using the maximum value of the reference ICO current adjustment bits is the maximum value of the ICO dynamic range. Therefore, the current amount of the sensor ICO needs to be offset by setting the offset adjustment bits so that the sensor ICO counter value does not exceed this value.
Read the CTSURC[15:0] bits after a CTSURD interrupt is generated. After these bits are read, they are cleared immediately before the CTSU measurement status counter value changes to Status 4 (the CTSUST.CTSUSTC[2:0] flags changes to 100b) in the next measurement. These bits are also cleared using the CTSUCR0.CTSUINIT bit.

47.2.20 CTSU Error Status Register (CTSUERRS)

Address(es): CTSU.CTSUERRS 000A 091Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CTSUICOMP	—	—	—	—	—	—	—	CTSUTSOC	—	—	—	CTSUDRV	CTSUTSOD	CTSUSPMD[1:0]	
Value after reset:	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CTSUSPMD[1:0]	Calibration Mode	b1 b0 0 0: Capacitance measurement mode 0 1: Setting prohibited 1 0: Calibration mode 1 1: Setting prohibited	R/W
b2	CTSUTSOD	TS Pin Fixed Output	0: Capacitance measurement mode 1: TS pins are forced to be high or low	R/W
b3	CTSUDRV	Calibration Setting 1	0: Capacitance measurement mode 1: Calibration setting 1	R/W
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	CTSUTSOC	Calibration Setting 2	0: Capacitance measurement mode 1: Calibration setting 2	R/W
b14 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	CTSUICOMP	TSCAP Voltage Error Monitor	0: Normal TSCAP voltage 1: Abnormal TSCAP voltage	R

CTSUSPMD[1:0] Bit (Calibration Mode)

These bits are used to calibrate the CTSU.
When measuring the capacitance, set these bits to 00b.

CTSUTSOD Bit (TS Pin Fixed Output)

This bit is used to calibrate the CTSU. When setting this bit to 1, the TS pins are forced to the logic level specified by the CTSUCR0.CTSUIOC bit.
When measuring the capacitance, set this bit to 0.

CTSUDRV Bit (Calibration Setting 1)

This bit is used to calibrate the CTSU.
When measuring the capacitance, set this bit to 0.

CTSUTSOC Bit (Calibration Setting 2)

This bit is used to calibrate the CTSU.
When measuring the capacitance, set this bit to 0.

CTSUICOMP Bit (TSCAP Voltage Error Monitor)

If the offset current amount set by the CTSUSO0 register exceeds the sensor ICO input current during touch measurement, the TSCAP voltage becomes abnormal and touch measurement cannot be correctly performed. This bit monitors the TSCAP voltage and it is set to 1 if the voltage becomes abnormal. If the TSCAP voltage becomes abnormal, the sensor ICO counter value will be undefined, but touch measurement is normally completed, so it difficult to detect an abnormality by reading the sensor ICO counter value. If the CTSU reference ICO current adjustment bits (CTSURICOA[7:0]) in the CTSUSO1 register are set to a value other than 0, check this bit when touch measurement is completed.

This bit is cleared by writing 0 to the CTSUCR1.CTSUPON bit and turning off the power supply.

47.2.21 CTSU Reference Current Calibration Register (CTSUTRMR)

Address(es): CTSU.CTSUTRMR 007F B0ECh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	CAL[7:0]							
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

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

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b23 to b16	CAL[7:0]	Reference Current Calibration	These bits calibrate the reference current used as the reference for measurement.	R/W
b31 to b24	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

The CTSUTRMR register stores a reference current value calibrated under the specified condition for each chip at factory shipment.
When rewriting this register, set the CTSUERRS.CTSUSPMD[1:0] bits to 10b (calibration mode). When resetting the MCU, the value returns to the factory setting value.
Do not rewrite this register when the CTSUSPMD[1:0] bits are 00b (capacitance measurement mode).

47.3 Operation

47.3.1 Principles of Measurement Operation

Figure 47.4 shows the measurement circuit.

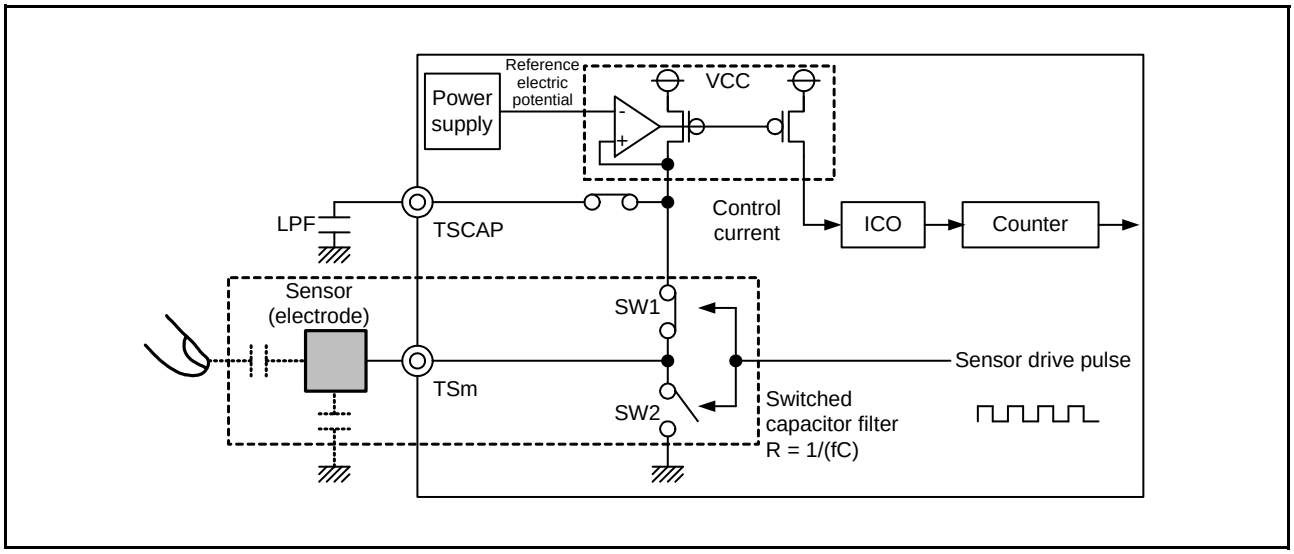


Figure 47.4 Measurement Circuit (m = 0 to 16)

The electrostatic capacitance measurement operation principles of the CTSU current frequency conversion method are explained using Figure 47.5 to Figure 47.7.

(1) The electrostatic capacitance of the electrode is charged by turning SW1 on and SW2 off (Figure 47.5).

(2) The charged capacitance is discharged by turning SW1 off and SW2 on (Figure 47.6).

Current flows to the switched capacitor filter by switching between charging and discharging in steps (1) and (2). At this time, the value of electrostatic capacitance varies depending on whether a finger is in close proximity, so the flowing current changes. A clock is generated by supplying the control current, which is proportional to the amount of the current flowing through the switched capacitor filter, from the circuit that generates the TSCAP power supply to the ICO. The counter is used to measure the clock frequency which changes depending on whether a finger is in close proximity, and the value read from the counter is used by software to determine contact with a finger (Figure 47.7).

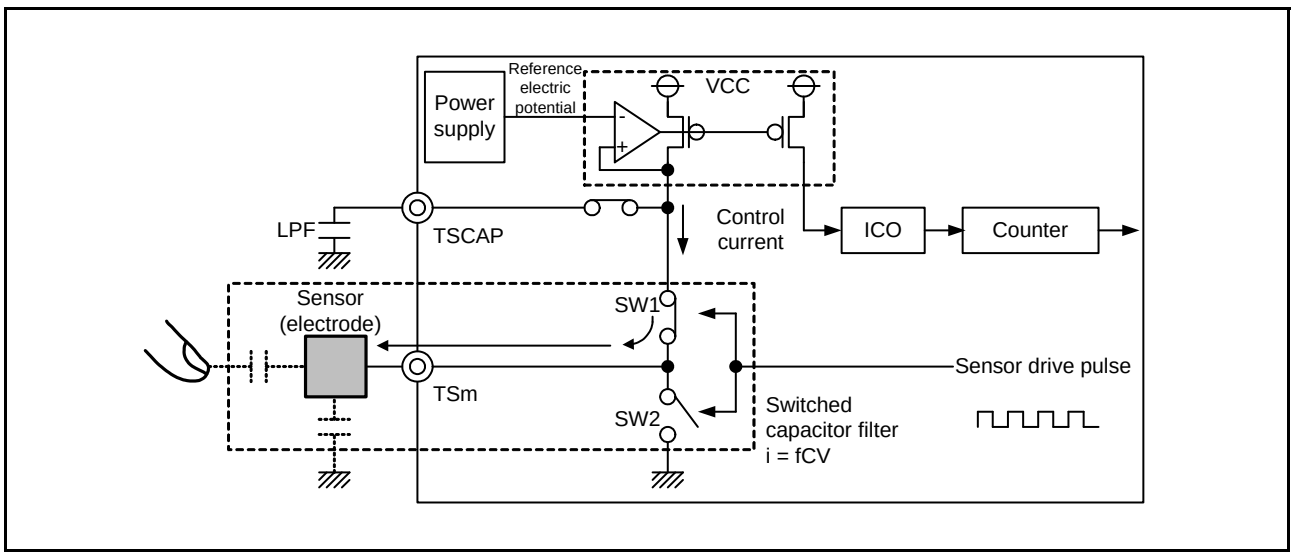


Figure 47.5 Charging Operation (m = 0 to 16)

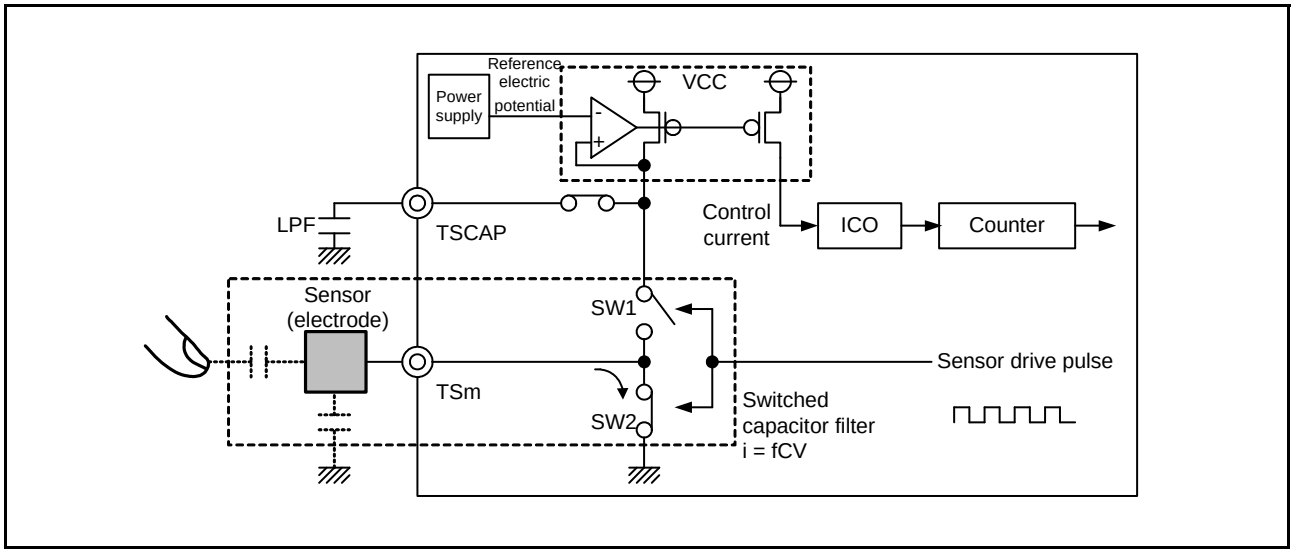


Figure 47.6 Discharging Operation (m = 0 to 16)

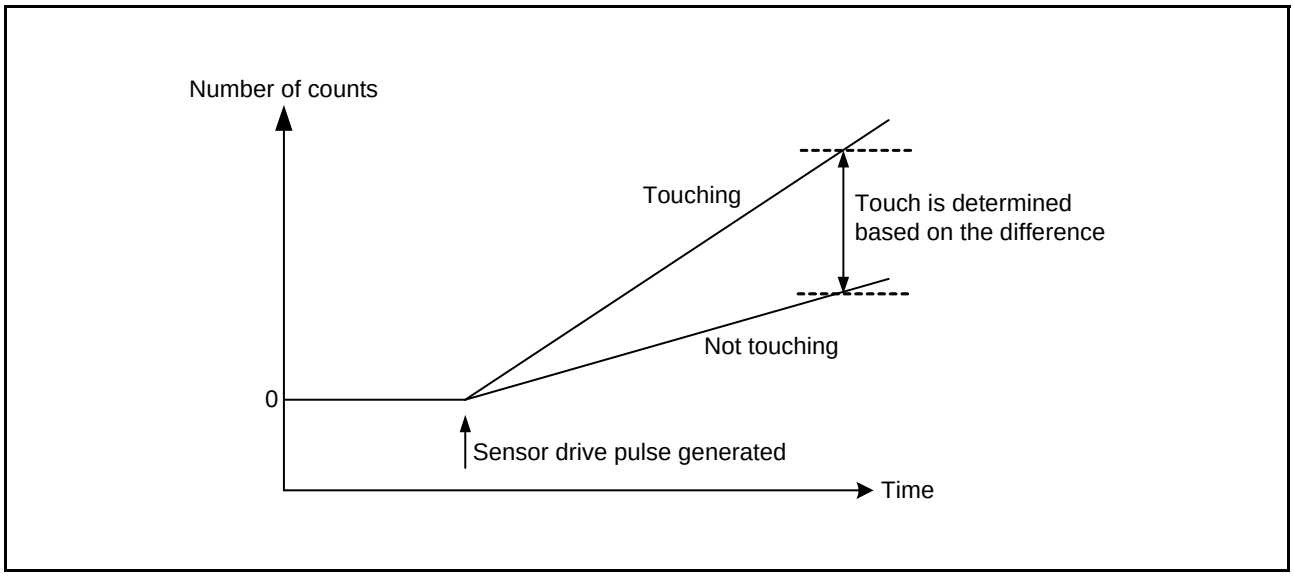


Figure 47.7 Change in Measured Value When Finger is Touching and Not Touching

47.3.2 Measurement Modes

The CTSU supports self-capacitance and mutual capacitance methods. Figure 47.8 illustrates these methods.

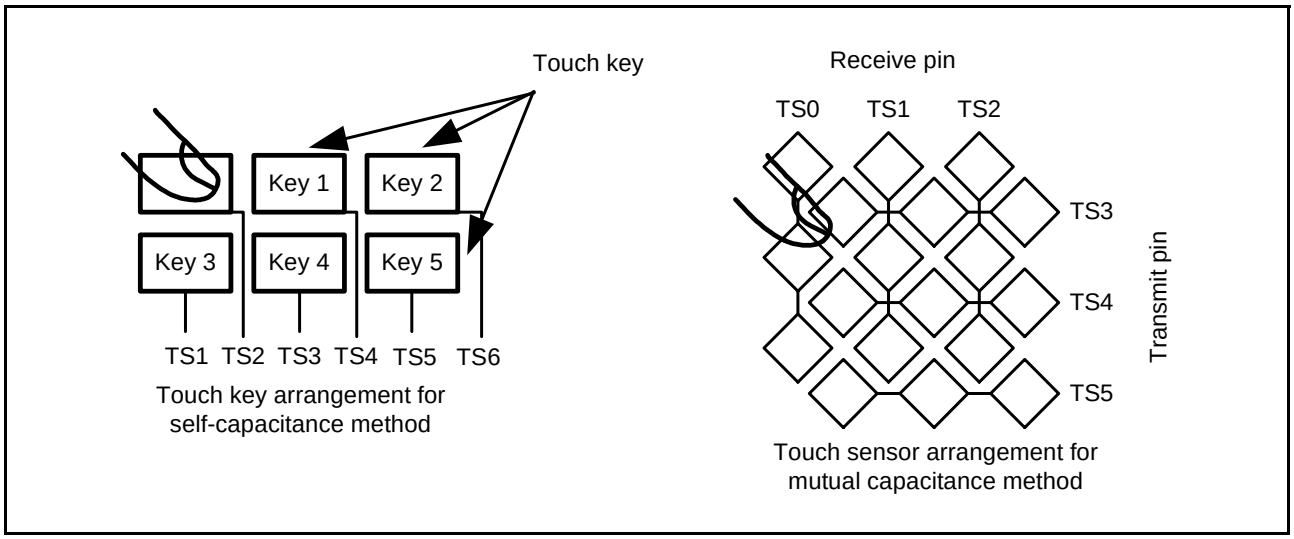


Figure 47.8 Overview of Self-Capacitance Method and Mutual Capacitance Method

In the self-capacitance method, a single touch pin is allocated to a single touch key to measure individual electrostatic capacitance when a finger is in close proximity. In this method, single scan and multi-scan can be used as measurement modes.

In the mutual capacitance method, the capacitance between two opposite electrodes (transmit and receive pins) is measured.

47.3.2.1 Initial Setting Flowchart

Figure 47.9 shows the flowchart for CTSU initial setting.

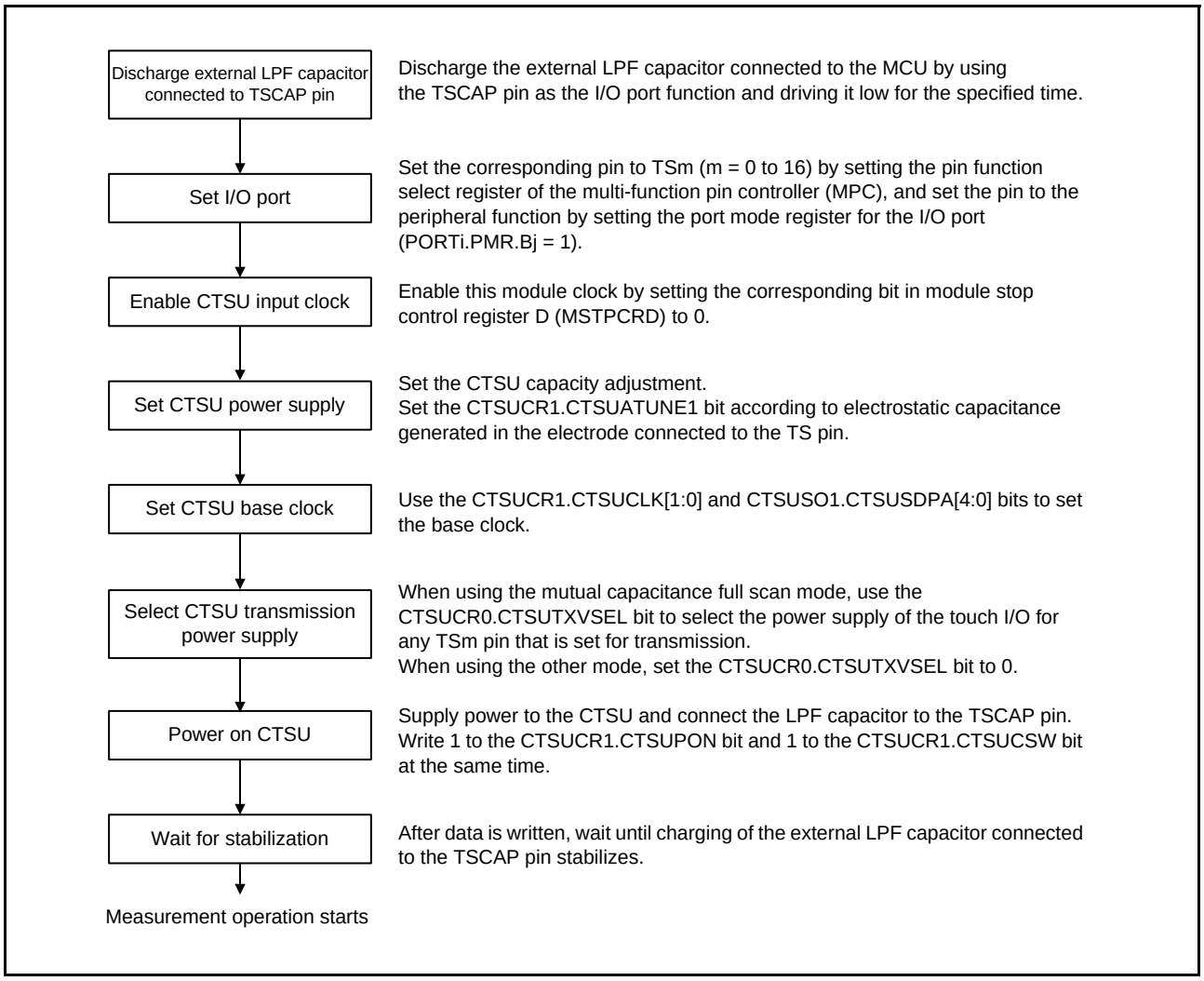


Figure 47.9 CTSU Initial Setting Flowchart

Figure 47.10 shows the flowchart for stopping CTSU operation and setting to the standby state.

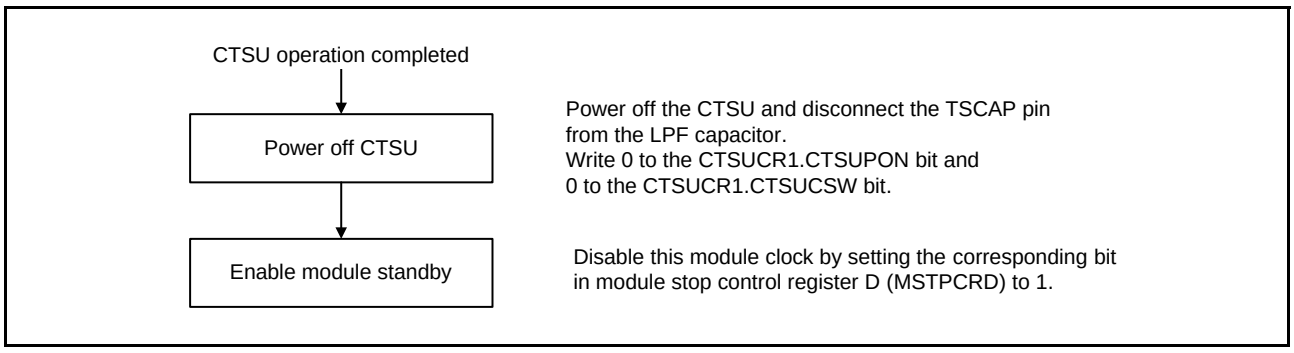


Figure 47.10 CTSU Stopping Flowchart

When restarting operation after it has been stopped, follow the initial setting flowchart shown in Figure 47.9.

47.3.2.2 Status Counter

The measurement status counter of the CTSU status register (CTSUST) indicates the current measurement status. The measurement status is common to all four modes. Figure 47.11 shows status operation transitions.

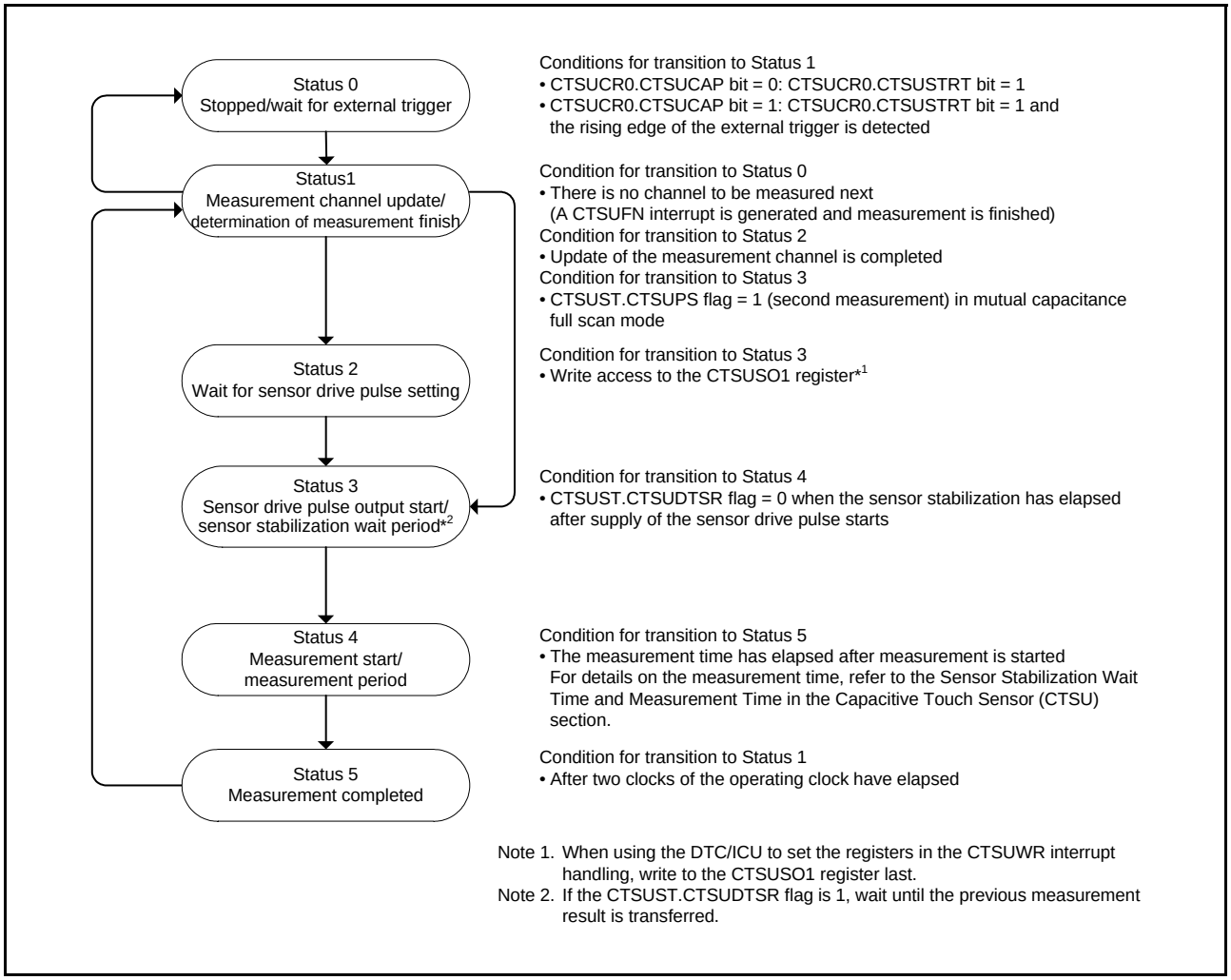


Figure 47.11 Status Operation Transitions

The status of the status counter transitions to Status 0 when all of the specified measurement channels are measured. The CTSUCR0.CTSUSTRT bit is cleared to 0 by hardware when a software trigger is used. When an external trigger is used, the value 1 is retained, and the CTSU waits for the next trigger.

When operation is forcibly stopped (by writing 0 to the CTSUCR0.CTSUSTRT bit and 1 to the CTSUCR0.CTSUINIT bit at the same time) during measurement or the wait state for the trigger, the status transitions to Status 0 and measurement is stopped forcibly.

If there is no channel to be measured by setting the CTSUMCH0, CTSUCHACn, and CTSUCHTRCn registers (n = 0 to 2), a CTSUFN interrupt is generated immediately after a transition to Status 1, and then the status transitions to Status 0. The following are the cases when there is no channel to be measured.

- A measurement target channel is not specified by the CTSUCHACn registers.
- In self-capacitance single scan mode, the channel specified in the CTSUMCH0 register is not a measurement target in the CTSUCHACn registers.
- In full scan mode, there is no transmit channel or receive channel to be measured by combining the CTSUCHACn and CTSUCHTRCn registers.

47.3.2.3 Self-Capacitance Single Scan Mode Operation

In self-capacitance single scan mode, electrostatic capacitance on a channel is measured. Figure 47.12 shows the software flowchart and an operation example, and Figure 47.13 shows the timing chart.

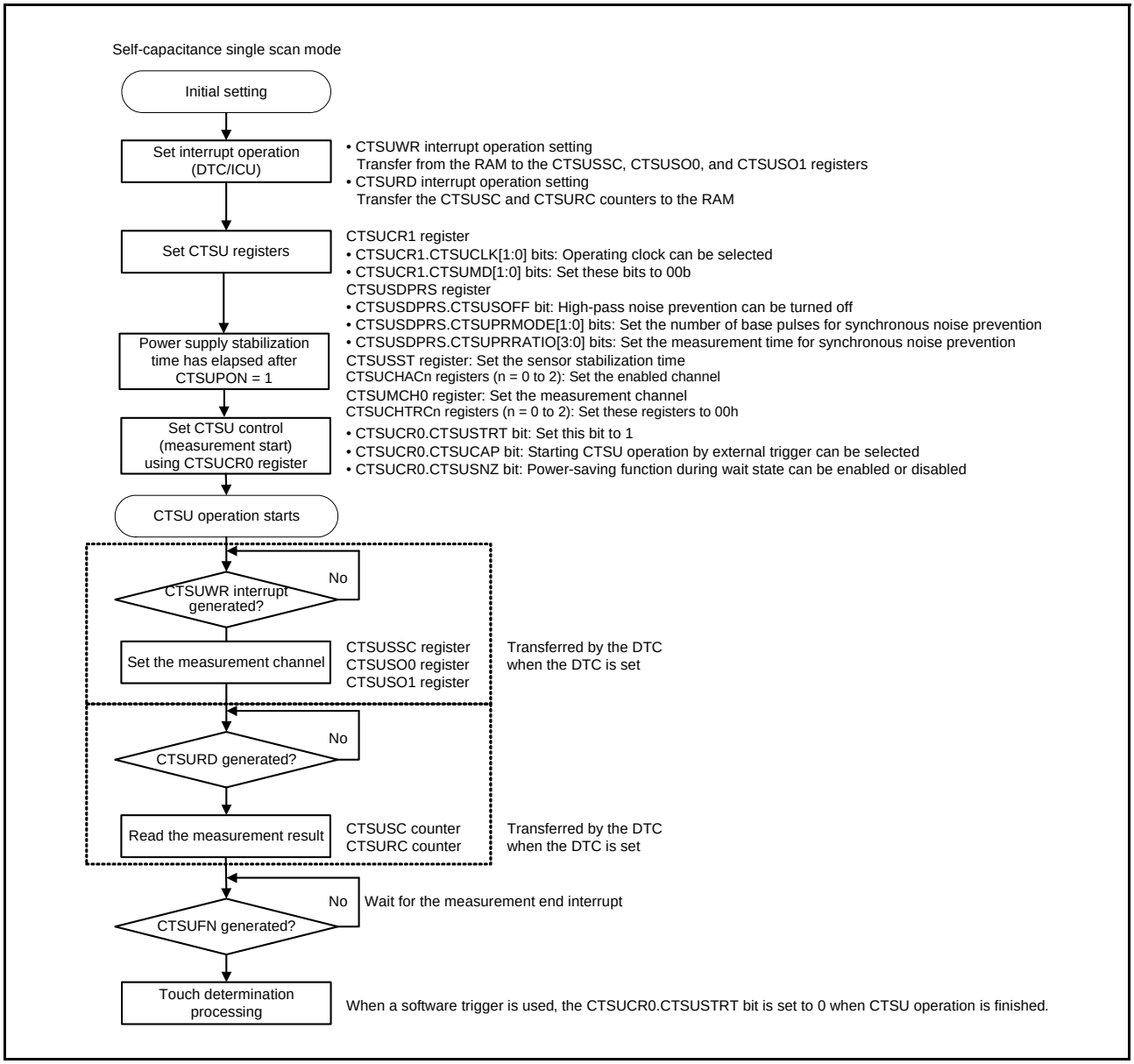


Figure 47.12 Software Flowchart and Operation Example of Self-Capacitance Single Scan Mode

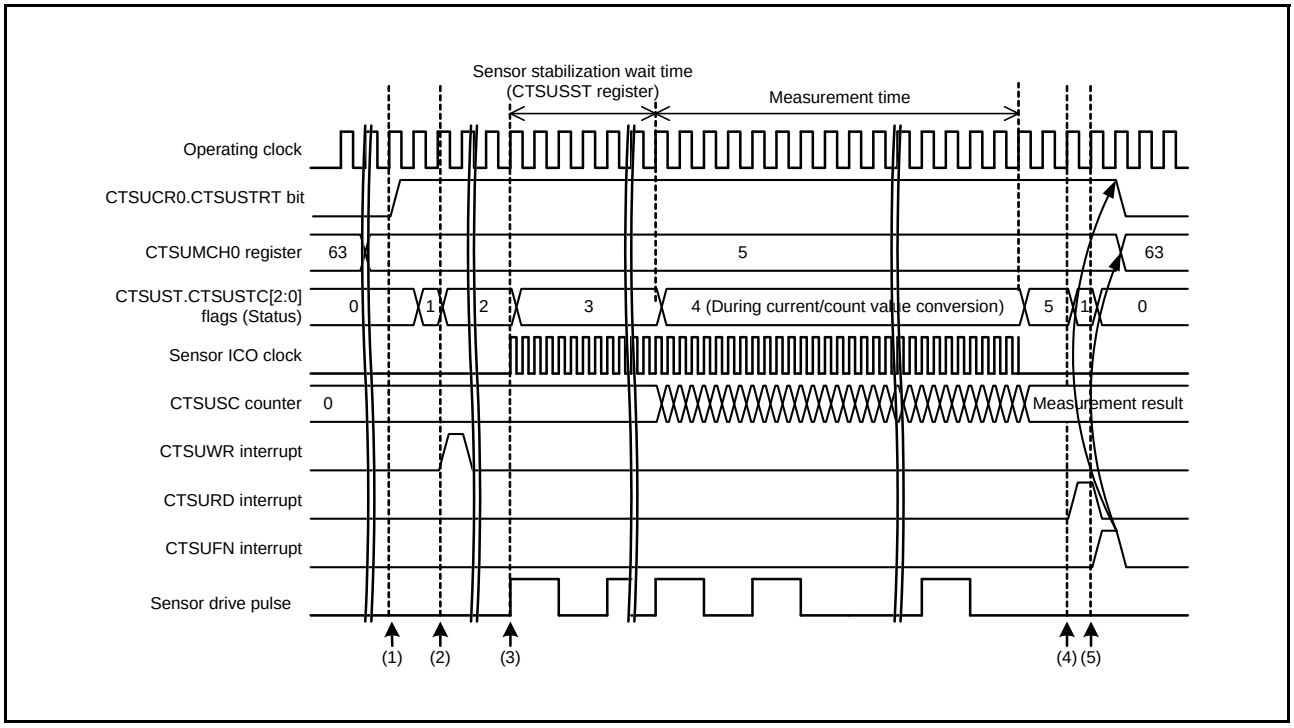


Figure 47.13 Timing Chart of Self-Capacitance Single Scan Mode (Measurement Start Condition is Software Trigger)

The following describes operation shown in the timing chart in Figure 47.13.

(1) After various settings are made, operation is started by writing 1 to the CTSUCR0.CTSUSTRT bit.

(2) After a channel to be measured is determined according to the preset conditions, a request for setting the corresponding channel (CTSUWR) is output.

(3) Upon completion of writing the measurement channel settings (CTSUSSC, CTSUSO0, and CTSUSO1 registers), the sensor drive pulse is output and the sensor ICO clock and the reference ICO clock operate.

(4) After the sensor stabilization wait time and the measurement time have elapsed and measurement is finished, a measurement result read request (CTSURD) is output.

(5) A measurement end interrupt (CTSUFN) is output and measurement is finished (transition to Status 0).

Table 47.7 lists the touch pin states in self-capacitance single scan mode.

Table 47.7 Touch Pin States in Self-Capacitance Single Scan Mode

	Touch Pin	
Status	**Measurement Channel**	**Non-Measurement Channel**
0	Low	Low
1	Low	Low
2	Low	Low
3	Pulse	Low
4	Pulse	Low
5	Low	Low

47.3.2.4 Self-Capacitance Multi-Scan Mode Operation

In self-capacitance multi-scan mode, electrostatic capacitance on all channels that are specified as measurement targets by setting the CTSUCHACn registers (n = 0 to 2) are measured sequentially in ascending order. Figure 47.14 shows the software flowchart and an operation example, and Figure 47.15 shows the timing chart.

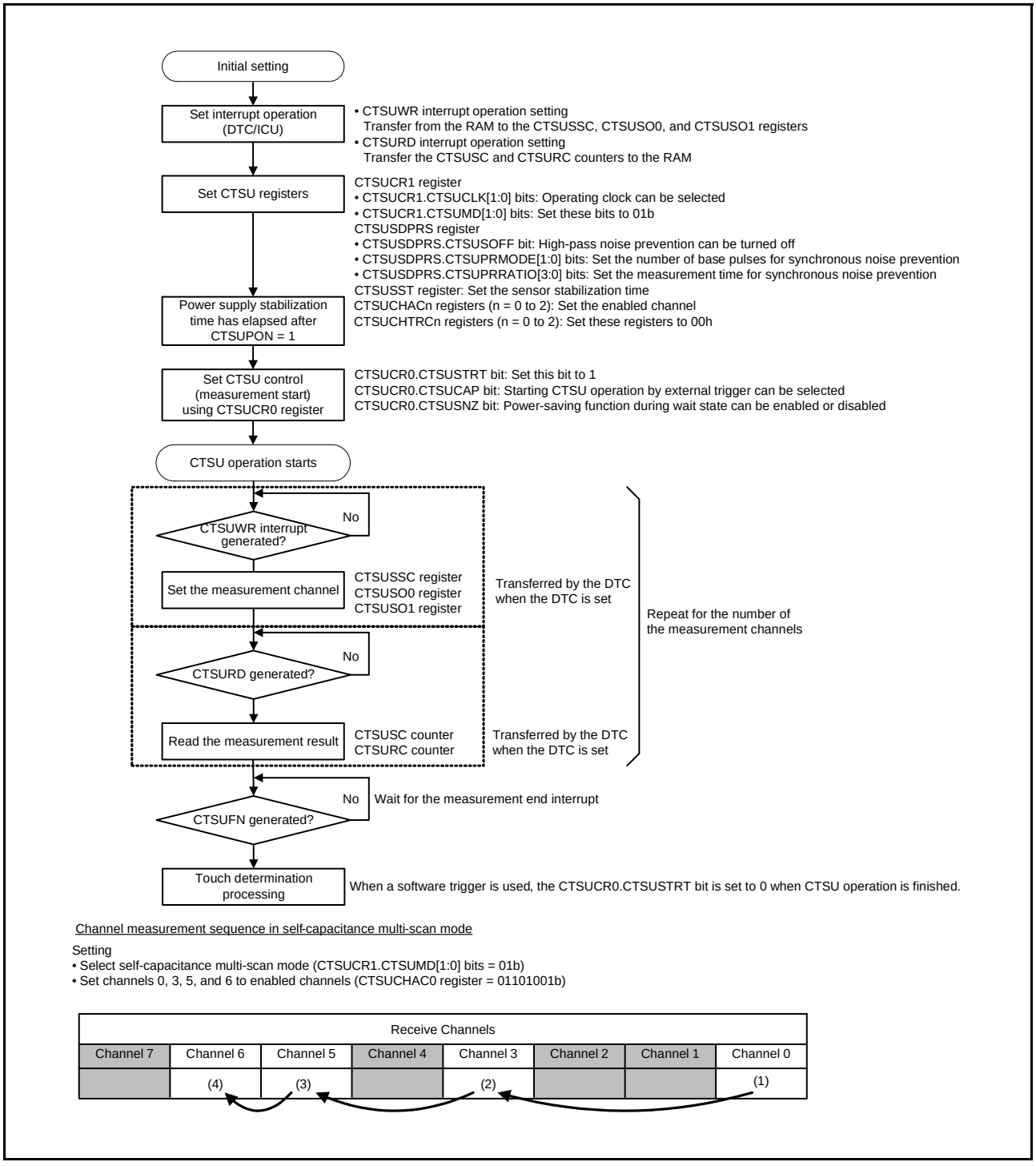


Figure 47.14 Software Flow and Operation Example of Self-Capacitance Multi-Scan Mode

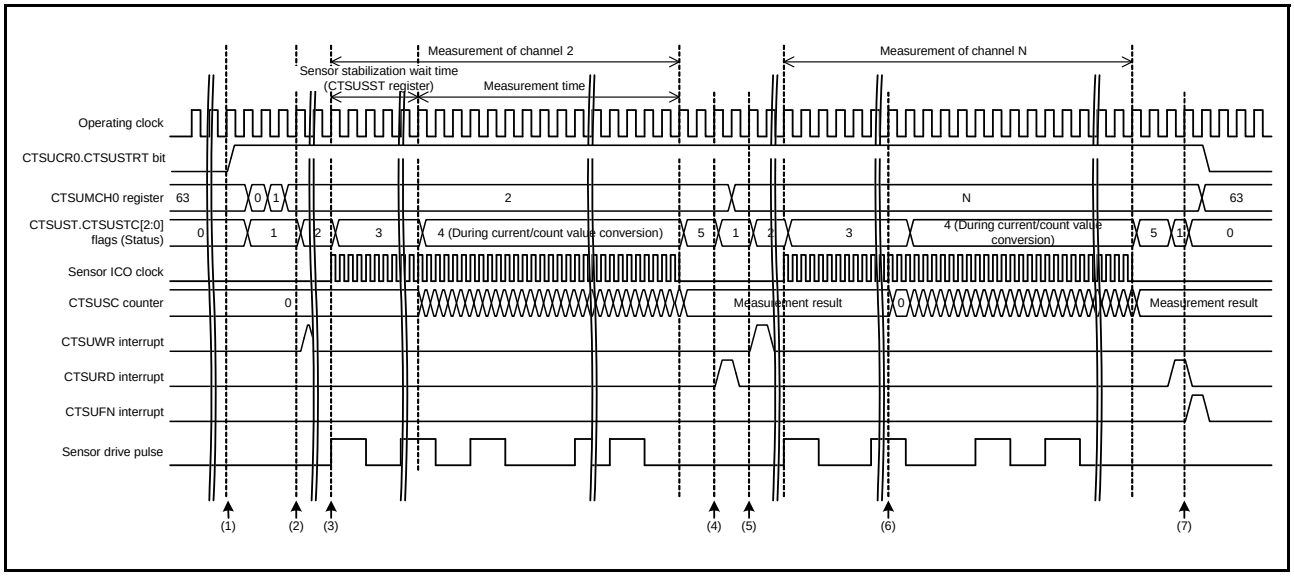


Figure 47.15 Timing Chart of Self-Capacitance Multi-Scan Mode (Measurement Start Condition is Software Trigger)

The following describes operation shown in the timing chart in Figure 47.15.

(1) After various settings are made, operation is started by writing 1 to the CTSUCR0.CTSUSTRT bit.

(2) After a channel to be measured is determined according to the preset conditions, a request for setting the corresponding channel (CTSUWR) is output.

(3) Upon completion of writing the measurement channel settings (CTSUSSC, CTSUSO0, and CTSUSO1 registers), the sensor drive pulse is output and the sensor ICO clock and the reference ICO clock operate.

(4) After the sensor stabilization wait time and the measurement time have elapsed and measurement is finished, a measurement result read request (CTSURD) is output.

(5) After a channel to be measured next is determined, a measurement channel setting request (CTSUWR) is output.

(6) After the stabilization wait time has elapsed and when the previous measurement is read, the result is cleared and measurement is started.

(7) Upon completion of all measurement channels, a measurement end interrupt (CTSUFN) is output and measurement is finished (transition to Status 0).

Table 47.8 lists the touch pin states in self-capacitance multi-scan mode.

Table 47.8 Touch Pin States in Self-Capacitance Multi-Scan Mode

	Touch Pin	
Status	**Measurement Channel**	**Non-Measurement Channel**
0	Low	Low
1	Low	Low
2	Low	Low
3	Pulse	Low
4	Pulse	Low
5	Low	Low

47.3.2.5 Mutual Capacitance Full Scan Mode Operation

In mutual capacitance full scan mode, measurement is performed during the high-level period of the sensor drive pulse on the receive channel by applying the edge to the target transmit channel to be measured. A single measurement target is measured twice, at the rising and falling edges. The difference between the data of these two measurements is used to determine whether or not the electrode is touched, thus achieving higher touch sensitivity.
Electrostatic capacitance is measured sequentially on channels set to transmission or reception specified by the CTSUCHTRCn registers (n = 0 to 2), and measurement targets specified by the CTSUCHACn registers. Electrostatic capacitance is measured by combining signals from the measurement target pins that are allocated to transmission or reception. Figure 47.16 shows the software flowchart and an operation example, and Figure 47.17 shows the timing chart.

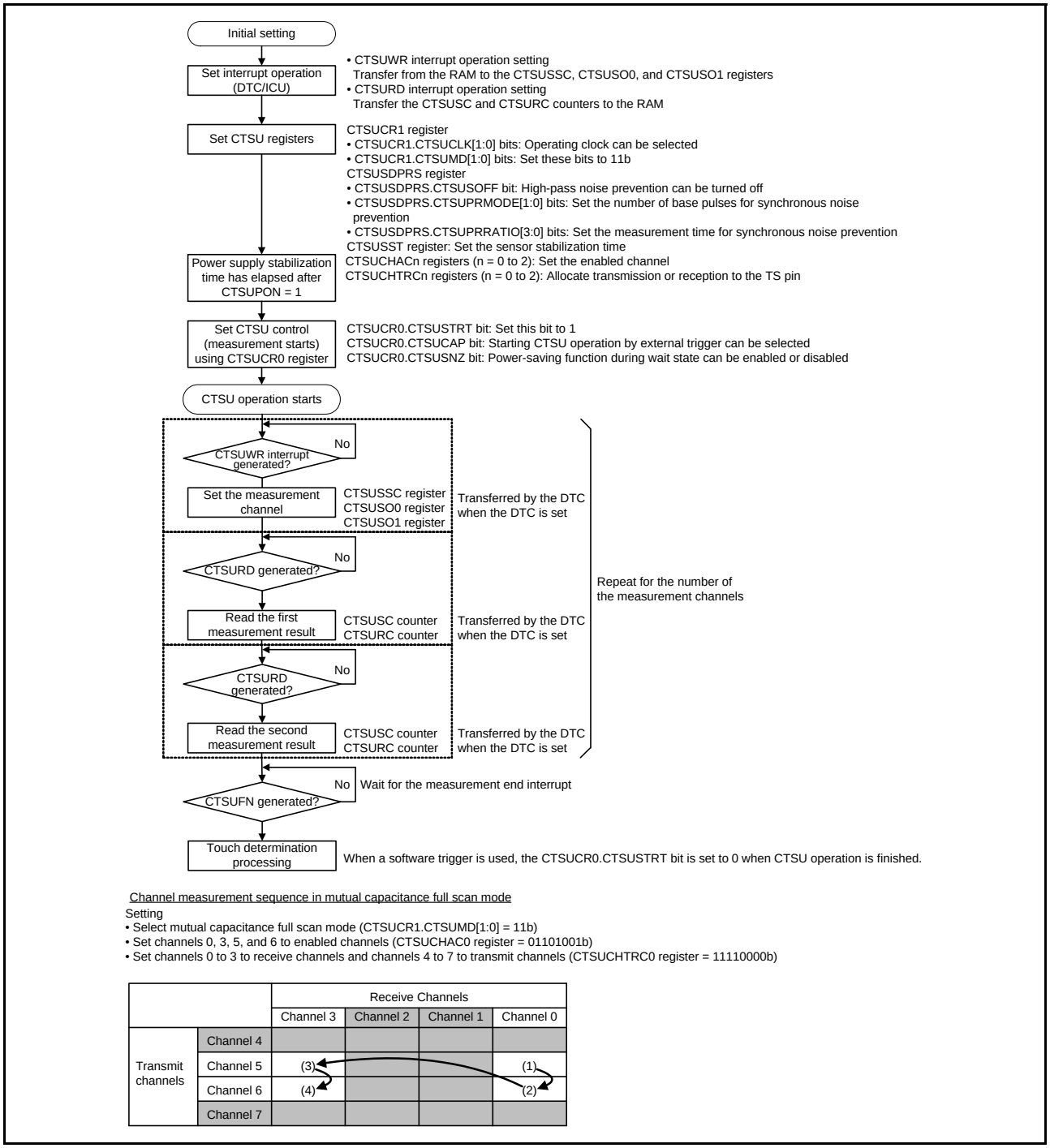


Figure 47.16 Software Flowchart and Operation Example of Mutual Capacitance Full Scan Mode

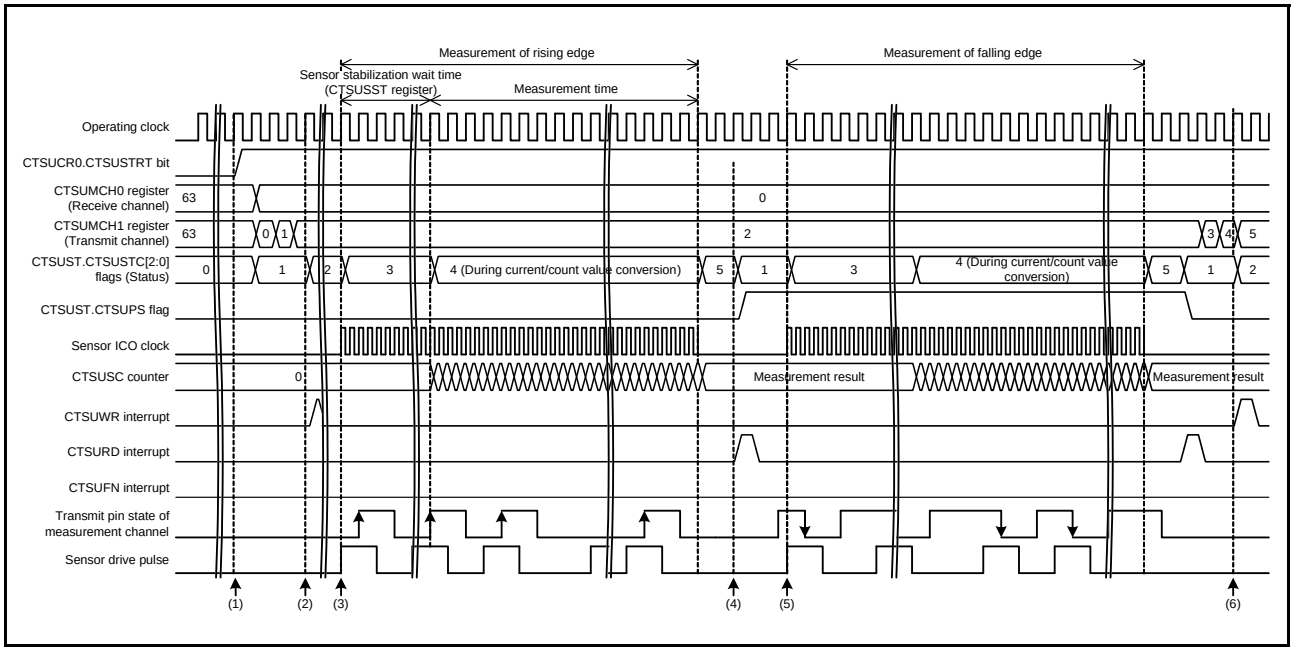


Figure 47.17 Timing Chart of Mutual Capacitance Full Scan Mode (Measurement Start Condition is Software Trigger)

The following describes operation shown in the timing chart in Figure 47.17.

(1) After various settings are made, operation is started by writing 1 to the CTSUCR0.CTSUSTRT bit.

(2) After a channel to be measured is determined according to the preset conditions, a request for setting the corresponding channel (CTSUWR) is output.

(3) Upon completion of writing the measurement channel settings (CTSUSSC, CTSUSO0, and CTSUSO1 registers), the sensor drive pulse is output and the sensor ICO clock and the reference ICO clock operate.
At the same time, a pulse which is handled as the rising edge is output to the transmit pin on the measurement channel during the high-level period of the sensor drive pulse.

(4) After the sensor stabilization wait time and the measurement time have elapsed and measurement is finished, a measurement result read request (CTSURD) is output.

(5) The same channel is measured by outputting a pulse that is handled as the falling edge during the high-level period of the sensor drive pulse.

(6) After the same channel is measured twice, a channel to be measured next is determined and measured in the similar way.

(7) Upon completion of all measurement channels, a measurement end interrupt (CTSUFN) is output and measurement is finished (transition to Status 0).

The mutual capacitance measurement status flag (CTSUST.CTSUPS flag) is changed when Status 5 transitions to Status 1.

Table 47.9 lists the touch pin states in mutual capacitance full scan mode.

Table 47.9 Touch Pin States in Mutual Capacitance Full Scan Mode

	Touch Pin of Receive Channel		Touch Pin of Transmit Channel		
Status	Measurement Channel	Non-Measurement Channel	Measurement Channel	Non-Measurement Channel	Remarks
0	Low	Low	Low	Low	—
1	Low	Low	Low/High	Low	—
2	Low	Low	Low	Low	—
3	Pulse	Low	Pulse	Low	Pulse of the phase same as that of the receive channel at the first measurement Pulse of the phase opposite to that of the receive channel at the second measurement
4	Pulse	Low	Pulse	Low	—
5	Low	Low	Low	Low	—

47.3.3 Items Common to Multiple Modes

47.3.3.1 Sensor Stabilization Wait Time and Measurement Time

Figure 47.18 shows the timing chart of the sensor stabilization wait time and measurement time.

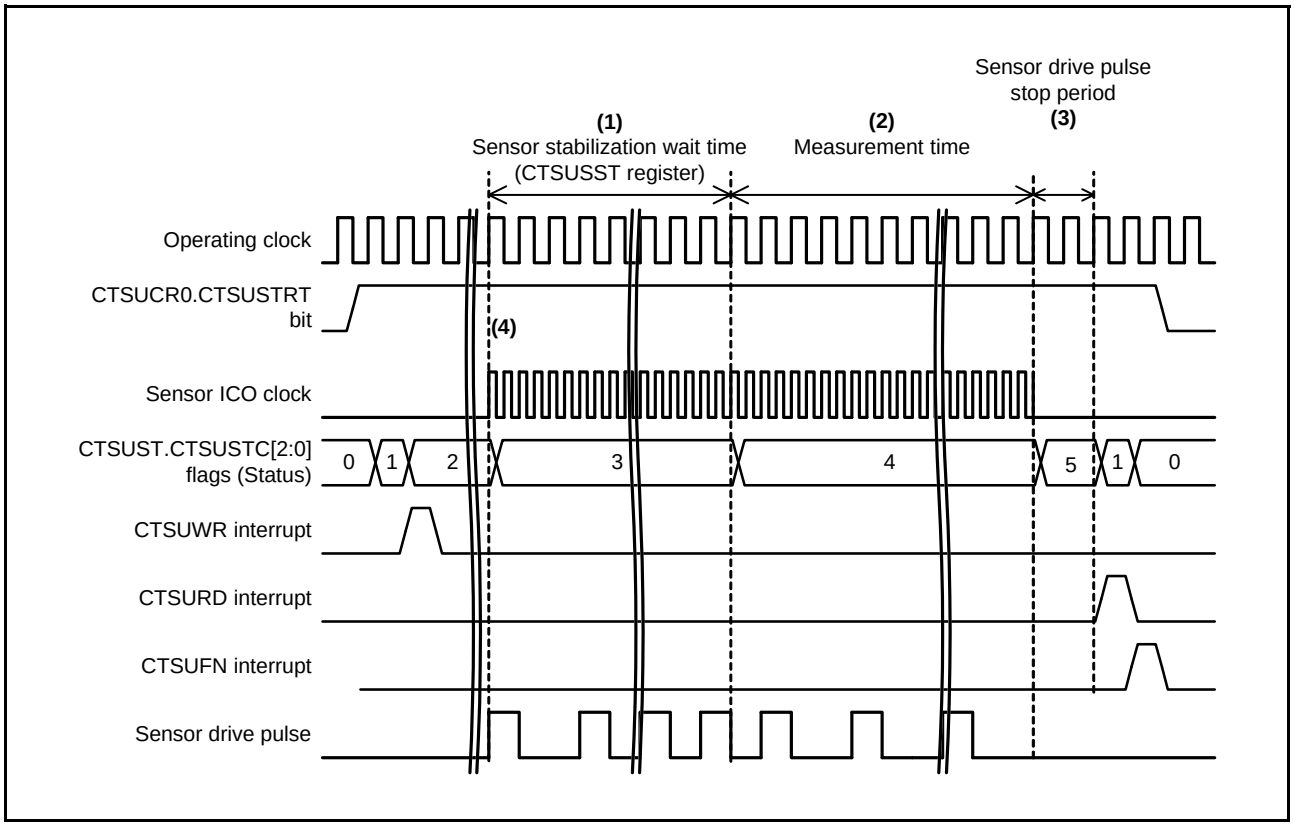


Figure 47.18 Sensor Stabilization Wait Time and Measurement Time

(1) In response to the CTSUWR interrupt request, output of the sensor drive pulse is started by write access to the CTSUSO1 register. Then, wait for the stabilization time set in the CTSUSST register.

(2) When the sensor stabilization time has elapsed and the CTSUST.CTSUDTSR flag is set to 0, measurement is started at transition to Status 4. The measurement time is determined by setting the base clock cycle and the CTSUSDPRS.CTSUPRMODE[1:0], CTSUPRRATIO[3:0], and CTSUSO0.CTSUSNUM[5:0] bits. When the measurement time has elapsed, measurement of the corresponding channel is finished.

(3) After the measurement time has elapsed, the status transitions to Status 1 after two operating clock cycles and a CTSURD interrupt is generated, so read the data from the CTSUSC and CTSURC counters.
At this time, the sensor drive pulse is output at the low level. When measurement of all specified channels is completed, the CTSUCR0.CTSUSTRT bit becomes 0.

(4) The sensor ICO clock oscillates while the CTSUSTC[2:0] flags are 011b (Status 3) or 100b (Status 4).

47.3.3.2 Interrupts

There are three types of interrupts for the CTSU:

- Write request interrupt for setting registers for each channel (CTSUWR)
- Measurement data transfer request interrupt (CTSURD)
- Measurement end interrupt (CTSUFN)

(1) Write request interrupt for setting registers for each channel (CTSUWR)

Store the setting data for each measurement channel in the RAM, and set the DTC or ICU transfer corresponding to the CTSUWR interrupt in advance. The CTSUWR interrupt is output when Status 1 transitions to Status 2. Write the setting data of the corresponding channel from the RAM to the CTSUSSC, CTSUSO0, and CTSUSO1 registers (Figure 47.19). Since write access to the CTSUSO1 register controls a transition to the next status, be sure to set this register last.

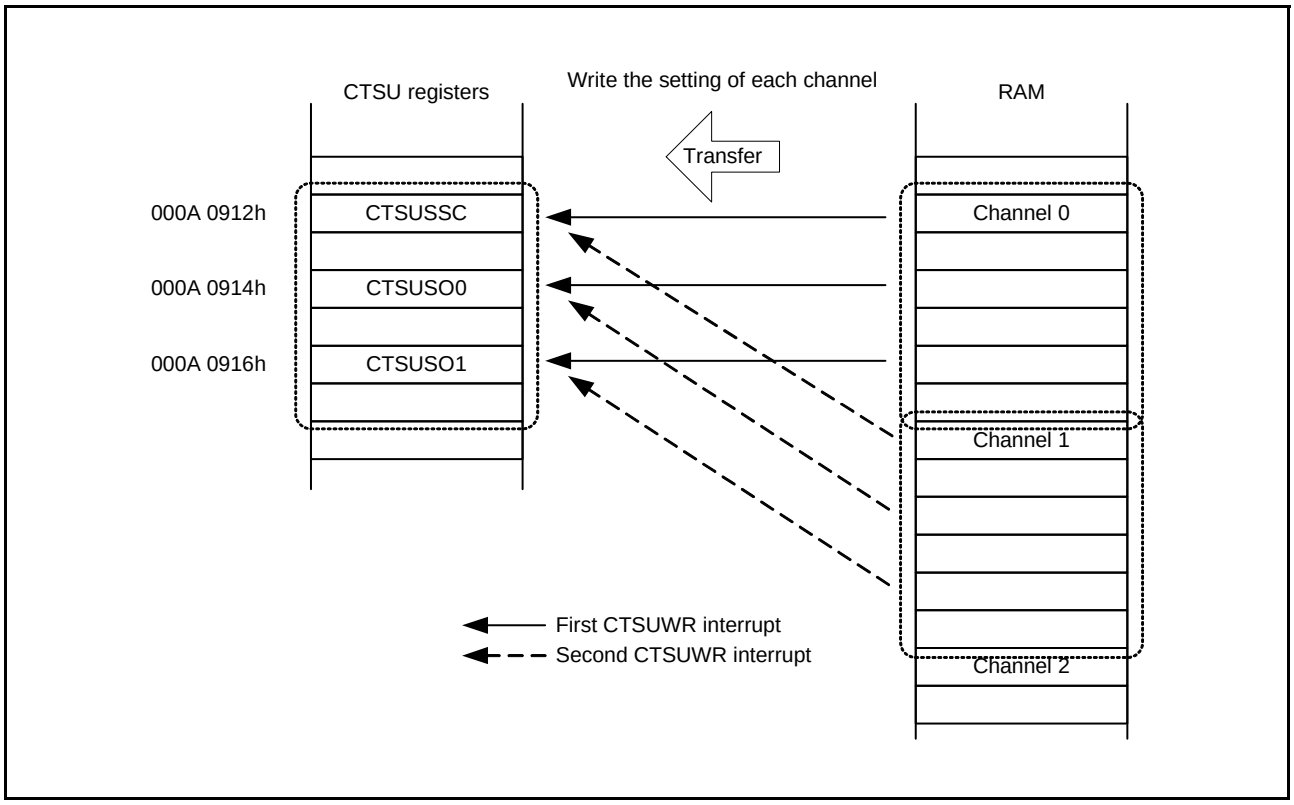


Figure 47.19 Example of DTC Transfer Operation Using CTSUWR Interrupt

The registers (CTSUSSC, CTSUSO0, and CTSUSO1 registers) to be set are allocated at sequential addresses. Set the operation at interrupt generation as shown below:

- Transfer destination address: Address of the CTSUSSC register
- Handling at the transfer destination address: Transfer 2-byte data three times by a single interrupt. (The address of the start byte is fixed.)
- Transfer source address: CTSUSSC register data storage address for the minimum channel in the setting data stored in the RAM
- Handling at the transfer source address: Transfer 2-byte data three times by a single interrupt. (The address of the first byte is continued from the previous interrupt handling.)
- Number of transfers by an interrupt: Specify the number of measurements.

(2) Measurement data transfer request interrupt (CTSURD)

Set DTC or ICU transfer corresponding to the CTSURD interrupt in advance. The CTSURD interrupt is output when Status 5 transitions to Status 1. Read the measurement result from the CTSUSC and CTSURC counters (Figure 47.20).

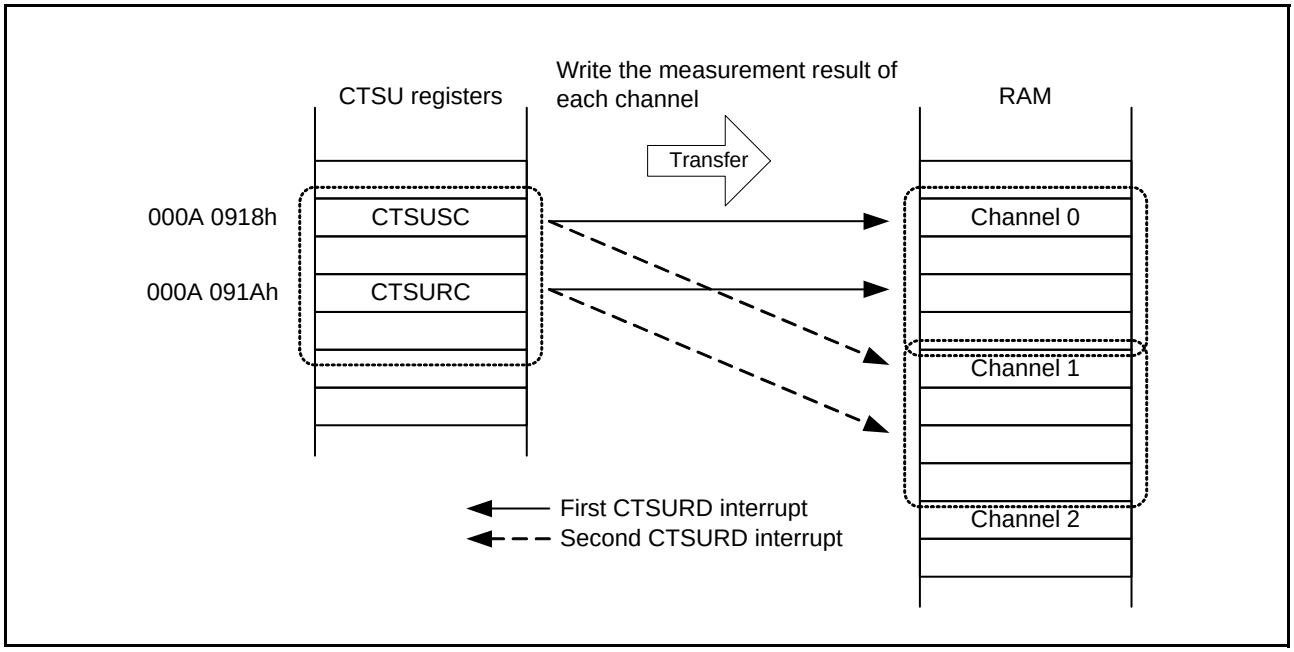


Figure 47.20 Example of DTC Transfer Operation Using CTSURD Interrupt

The measurement result registers (CTSUSC and CTSURC counters) used as transfer sources are allocated at sequential addresses. Set the operation at interrupt generation as shown below:

- Transfer source address: Address of the CTSUSC counter
- Handling at the transfer source address: Transfer 2-byte data twice by a single interrupt. (The start address is fixed.)
- Transfer destination address: CTSUSC counter data storage address for the minimum channel in the setting data stored in the RAM.
- Handling at the transfer destination address: Transfer 2-byte data twice by a single interrupt. (The start address is continued from the previous interrupt handling.)
- Number of transfers by an interrupt: Specify the number of measurements.

(3) Measurement end interrupt (CTSUFN)

When all channels are measured, an interrupt is generated when Status 1 transitions to Status 0. Use software to confirm the overflow flags (CTSUST.CTSUSOVF and CTSUROVF flags) and read the measurement results to determine whether or not the electrode is touched.

Interrupt requests are accepted or disabled in the interrupt control block.

47.4 Usage Notes

47.4.1 Measurement Result Data (CTSUSC and CTSURC Counters)

Read access during measurement is prohibited. If the measurement result data is accessed, an incorrect value may be read due to asynchronous operation.

47.4.2 Software Trigger

When 10b (PCLK/4) is selected by the CTSUCR1.CTSUCLK[1:0] bits, to restart measurement by writing 1 to the CTSUCR0.CTSUSTRT bit after measurement has been completed, wait for at least three cycles to elapse after an interrupt is generated, and then write to the CTSUCR0.CTSUSTRT bit.

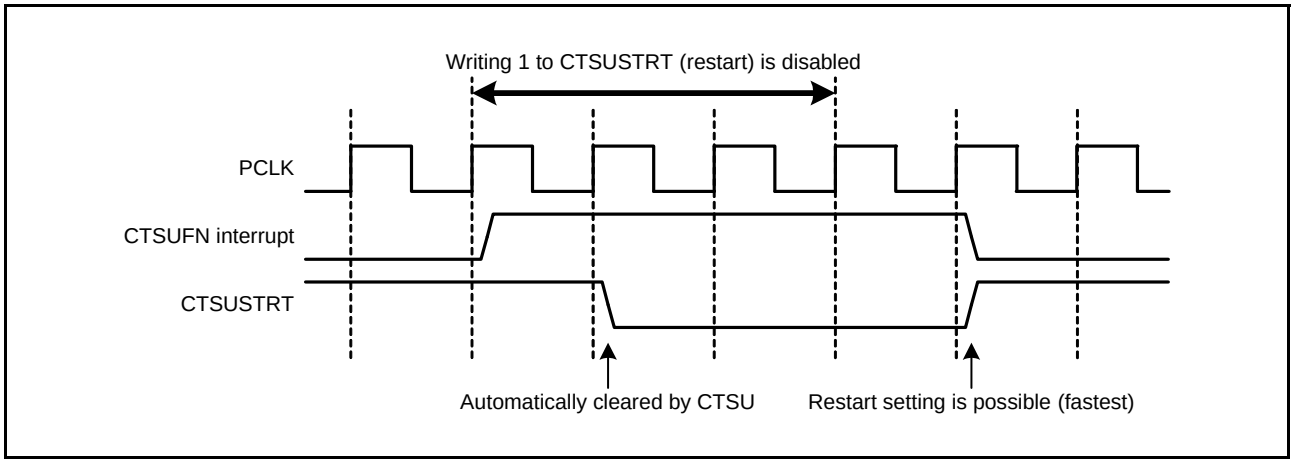


Figure 47.21 Notes on Restarting Measurement

47.4.3 External Trigger

- If an external trigger is input during the measurement time, measurement is not started. The next external event is enabled after one cycle of the operating clock when a CTSUFN interrupt is generated.
- To stop external trigger mode, write 0 to the CTSUCR0.CTSUSTRT bit and 1 to the CTSUCR0.CTSUINIT bit at the same time (forced stop).

47.4.4 Notes on Forcibly Stopping Operation

To forcibly stop the current operation, write 0 to the CTSUCR0.CTSUSTRT bit and 1 to the CTSUCR0.CTSUINIT bit at the same time. After this setting, the operation is stopped and the internal control registers are initialized.
When the CTSUCR0.CTSUINIT bit is used for initialization, the following registers are initialized in addition to the initialization of the internal measurement state.

- CTSUMCH0 register
- CTSUMCH1 register
- CTSUST register
- CTSUSC counter
- CTSURC counter

If operation is forcibly stopped, an interrupt request may be generated depending on the internal state. After operation is forcibly stopped, perform the processing for stopping/disabling the DTC or ICU.

If DTC transfer is stopped in the mounted system for some reason, also perform the processing for forcibly stopping and initializing the CTSU.

47.4.5 TSCAP Pin

The TSCAP pin requires an external decoupling capacitor to stabilize CTSU internal voltage. The traces between the TSCAP pin and the capacitor, and the capacitor and ground should be as short and wide as physically possible.
The capacitor connected to the TSCAP pin should be fully discharged using I/O port control to output a low level, before turning on the switch (CTSUCR1.CTSUCSW bit = 1) to establish a connection.

47.4.6 Notes during Measurement Operation (CTSUCR0.CTSUSTRT Bit = 1)

During measurement operation (CTSUCR0.CTSUSTRT bit = 1), do not use settings such as “stop the peripheral module clock” or “change the port settings related to the touch pins (TS and TSCAP pins)” in the higher layers of the system.
If control settings non-compliant to these restrictions are made, after operation is forcibly stopped (CTSUCR0.CTSUSTRT bit = 0 and CTSUCR0.CTSUINIT bit = 1), write 0 to the CTSUCR1.CTSUPON bit and 0 to the CTSUCR1.CTSUCSW bit at the same time, and set the CTSUCR0.CTSUSNZ bit to 0. Then, restart from the initial setting flow shown in Figure 47.9.

48. Boundary Scan

This MCU has boundary scan function.
The boundary scan is a serial I/O interface based on the JTAG (Joint Test Action Group, IEEE Std.1149.1 and IEEE Standard Test Access Port and Boundary-Scan Architecture).

48.1 Overview

Table 48.1 lists the specifications of boundary scan.
Figure 48.1 shows a block diagram of the boundary scan function.

Table 48.1 Specifications of Boundary Scan

Item	Description
Boundary scan enabled/disabled	Boundary scan is enabled when the RES# pin and the BSCANP pin are driven high and the EMLE pin is driven low.
Dedicated boundary scan pins	The following pins are dedicated for the JTAG, when boundary scan function is enabled (TDO/TCK/TDI/TMS/TRST#). 145-pin TFLGA/64-pin TFBGA: P26/P27/P30/P31/P34
Six test modes	• BYPASS mode • EXTEST mode • SAMPLE/PRELOAD mode • CLAMP mode • HIGHZ mode • IDCODE mode

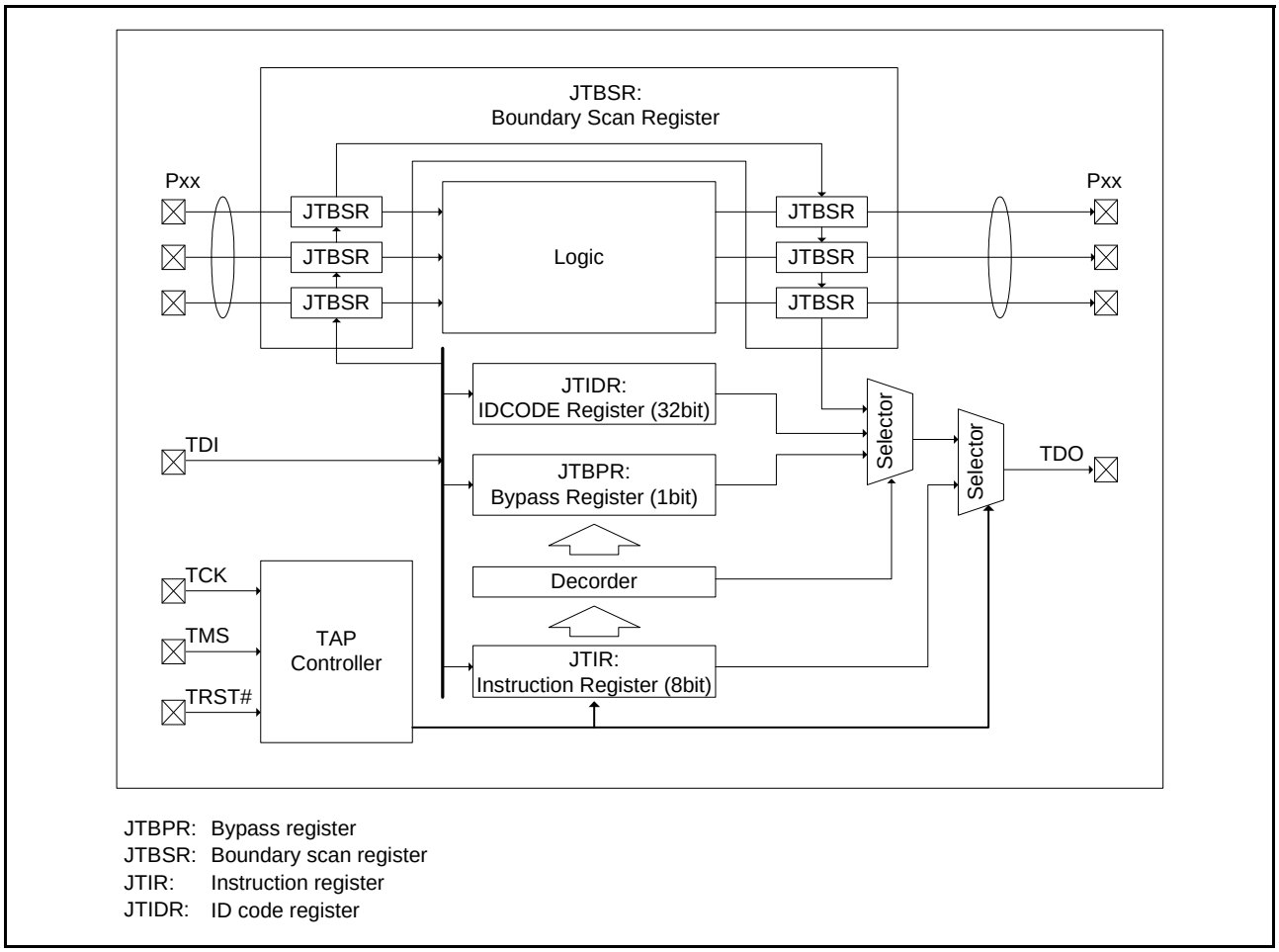


Figure 48.1 Block Diagram of JTAG

Table 48.2 shows the I/O pins used in the boundary scan function.

Table 48.2 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

48.2 Register Descriptions

Table 48.3 lists the boundary scan registers.

Table 48.3 List of Boundary Scan Registers

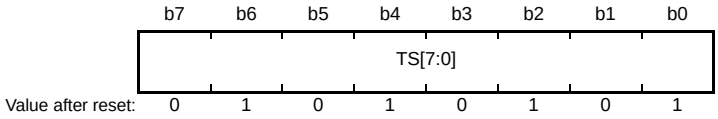
Register Name	Symbol	Value after Reset
Instruction register	JTIR	55h
ID code register	JTIDR	0841 8447h
Bypass register	JTBPR	Undefined
Boundary scan register	JTBSR	Undefined

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 48.6 to Table 48.8, 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 48.4 shows the availability of serial transfer for the registers.

Table 48.4 Serial Transfer for the Registers

Register Name	Serial Input	Serial Output
Instruction register (JTIR)	Available	Available
ID code register (JTIDR)	Available	Available
Bypass register (JTBPR)	Available	Available
Boundary scan register (JTBSR)	Available	Available

48.2.1 Instruction Register (JTIR)



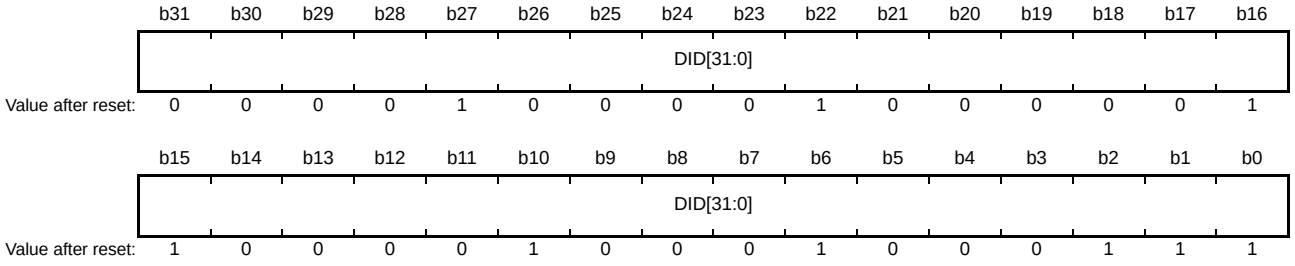
Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TS[7:0]	Test Bit Set	The command configuration is as shown in Table 48.5.	—

Table 48.5 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.

48.2.2 ID Code Register (JTIDR)



Bit	Symbol	Bit Name	Description	R/W
b31 to b0	DID[31:0]	Reserved	JTIDR is a register with the fixed value that indicates the device IDCODE.	—

JTIDR data is output from the TDO pin when the IDCODE instruction has been executed.

48.2.3 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.

48.2.4 Boundary Scan Register (JTBSR)

The JTBSR register is a shift register to control the external input and output pins of this MCU 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 48.6 to Table 48.8 show the correspondence between the JTBSR bits and the pins of this MCU.
The value after reset is undefined.

Table 48.6 Boundary Scan Register 145-Pin TFLGA (0.65-mm pitch) (1/8)

Pin No.	Pin Name	Input/Output	Bit Name
From TDI			
B3	P05	Output	339
		Output enable	338
		Input	337
D3	P03	Output	336
		Output enable	335
		Input	334
C2	P02	Output	333
		Output enable	332
		Input	331
D4	P01	Output	330
		Output enable	329
		Input	328
D1	P00	Output	327
		Output enable	326
		Input	325
D2	PF5	Output	324
		Output enable	323
		Input	322
E3	PJ5	Output	321
		Output enable	320
		Input	319
F3	PJ3	Output	318
		Output enable	317
		Input	316
G3	MD	Output	315
		Output enable	314
		Input	313
H4	P35	Input	309
J2	P33	Output	305
		Output enable	304
		Input	303
J3	P32	Output	302
		Output enable	301
		Input	300
L1	P25	Output	287
		Output enable	286
		Input	285
L4	P24	Output	284
		Output enable	283
		Input	282
L2	P23	Output	281
		Output enable	280
		Input	279

Table 48.6 Boundary Scan Register 145-Pin TFLGA (0.65-mm pitch) (2/8)

Pin No.	Pin Name	Input/Output	Bit Name
M1	P22	Output	278
		Output enable	277
		Input	276
N1	P21	Output	275
		Output enable	274
		Input	273
N2	P20	Output	272
		Output enable	271
		Input	270
M2	P17	Output	269
		Output enable	268
		Input	267
N3	P87	Output	266
		Output enable	265
		Input	264
L3	P16	Output	263
		Output enable	262
		Input	261
M3	P86	Output	260
		Output enable	259
		Input	258
K4	P15	Output	257
		Output enable	256
		Input	255
N4	P14	Output	254
		Output enable	253
		Input	252
L5	P13	Output	251
		Output enable	250
		Input	249
M4	P12	Output	248
		Output enable	247
		Input	246
N5	PH2/ USB0_DM	Output	245
		Output enable	244
		Input	243
N6	PH1/ USB0_DP	Output	242
		Output enable	241
		Input	240
L6	P56	Output	239
		Output enable	238
		Input	237
N7	P55	Output	236
		Output enable	235
		Input	234

Table 48.6 Boundary Scan Register 145-Pin TFLGA (0.65-mm pitch) (3/8)

Pin No.	Pin Name	Input/Output	Bit Name
K5	P54	Output	233
		Output enable	232
		Input	231
K6	P53	Output	230
		Output enable	229
		Input	228
L7	P52	Output	227
		Output enable	226
		Input	225
K7	P51	Output	224
		Output enable	223
		Input	222
M7	P50	Output	221
		Output enable	220
		Input	219
L8	P83	Output	218
		Output enable	217
		Input	216
N9	PC7	Output	215
		Output enable	214
		Input	213
M8	PC6	Output	212
		Output enable	211
		Input	210
L9	PC5	Output	209
		Output enable	208
		Input	207
N10	P82	Output	206
		Output enable	205
		Input	204
M9	P81	Output	203
		Output enable	202
		Input	201
K9	P80	Output	200
		Output enable	199
		Input	198
L10	PC4	Output	197
		Output enable	196
		Input	195
N11	PC3	Output	194
		Output enable	193
		Input	192
M10	P77	Output	191
		Output enable	190
		Input	189

Table 48.6 Boundary Scan Register 145-Pin TFLGA (0.65-mm pitch) (4/8)

Pin No.	Pin Name	Input/Output	Bit Name
K10	P76	Output	188
		Output enable	187
		Input	186
L11	PC2	Output	185
		Output enable	184
		Input	183
N12	P75	Output	182
		Output enable	181
		Input	180
N13	P74	Output	179
		Output enable	178
		Input	177
M12	PC1	Output	176
		Output enable	175
		Input	174
M11	PC0	Output	173
		Output enable	172
		Input	171
L12	P73	Output	170
		Output enable	169
		Input	168
K11	PB7	Output	167
		Output enable	166
		Input	165
K12	PB6	Output	164
		Output enable	163
		Input	162
K13	PB5	Output	161
		Output enable	160
		Input	159
J11	PB4	Output	158
		Output enable	157
		Input	156
J10	PB3	Output	155
		Output enable	154
		Input	153
J12	PB2	Output	152
		Output enable	151
		Input	150
J13	PB1	Output	149
		Output enable	148
		Input	147
H10	PB0	Output	140
		Output enable	139
		Input	138

Table 48.6 Boundary Scan Register 145-Pin TFLGA (0.65-mm pitch) (5/8)

Pin No.	Pin Name	Input/Output	Bit Name
H11	PA7	Output	137
		Output enable	136
		Input	135
G11	PA6	Output	134
		Output enable	133
		Input	132
G10	PA5	Output	131
		Output enable	130
		Input	129
F11	PA4	Output	128
		Output enable	127
		Input	126
F10	PA3	Output	125
		Output enable	124
		Input	123
F13	PA2	Output	122
		Output enable	121
		Input	120
F12	PA1	Output	119
		Output enable	118
		Input	117
E10	PA0	Output	116
		Output enable	115
		Input	114
E13	P67	Output	113
		Output enable	112
		Input	111
E11	P66	Output	110
		Output enable	109
		Input	108
E12	P65	Output	107
		Output enable	106
		Input	105
D10	PE7	Output	104
		Output enable	103
		Input	102
D13	PE6	Output	101
		Output enable	100
		Input	99
C12	P70	Output	98
		Output enable	97
		Input	96
D12	PE5	Output	95
		Output enable	94
		Input	93

Table 48.6 Boundary Scan Register 145-Pin TFLGA (0.65-mm pitch) (6/8)

Pin No.	Pin Name	Input/Output	Bit Name
B13	PE4	Output	92
		Output enable	91
		Input	90
A13	PE3	Output	89
		Output enable	88
		Input	87
B12	PE2	Output	86
		Output enable	85
		Input	84
A12	PE1	Output	83
		Output enable	82
		Input	81
C11	PE0	Output	80
		Output enable	79
		Input	78
D9	P64	Output	77
		Output enable	76
		Input	75
C10	P63	Output	74
		Output enable	73
		Input	72
A11	P62	Output	71
		Output enable	70
		Input	69
B11	P61	Output	68
		Output enable	67
		Input	66
D8	P60	Output	65
		Output enable	64
		Input	63
C9	PD7	Output	62
		Output enable	61
		Input	60
A9	PD6	Output	59
		Output enable	58
		Input	57
D7	PD5	Output	56
		Output enable	55
		Input	54
B9	PD4	Output	53
		Output enable	52
		Input	51
C8	PD3	Output	50
		Output enable	49
		Input	48

Table 48.6 Boundary Scan Register 145-Pin TFLGA (0.65-mm pitch) (7/8)

Pin No.	Pin Name	Input/Output	Bit Name
A8	PD2	Output	47
		Output enable	46
		Input	45
C7	PD1	Output	44
		Output enable	43
		Input	42
B8	PD0	Output	41
		Output enable	40
		Input	39
D6	P93	Output	38
		Output enable	37
		Input	36
A7	P92	Output	35
		Output enable	34
		Input	33
B7	P91	Output	32
		Output enable	31
		Input	30
A6	P90	Output	29
		Output enable	28
		Input	27
B6	P47	Output	26
		Output enable	25
		Input	24
C5	P46	Output	23
		Output enable	22
		Input	21
A5	P45	Output	20
		Output enable	19
		Input	18
E5	P44	Output	17
		Output enable	16
		Input	15
B5	P43	Output	14
		Output enable	13
		Input	12
A4	P42	Output	11
		Output enable	10
		Input	9
C4	P41	Output	8
		Output enable	7
		Input	6
A3	P40	Output	5
		Output enable	4
		Input	3

Table 48.6 Boundary Scan Register 145-Pin TFLGA (0.65-mm pitch) (8/8)

Pin No.	Pin Name	Input/Output	Bit Name
A2	P07	Output	2
		Output enable	1
		Input	0
To TDO			

Table 48.7 Boundary Scan Register 145-Pin TFLGA (0.50-mm pitch) (1/8)

Pin No.	Pin Name	Input/Output	Bit Name
From TDI			
B3	P05	Output	339
		Output enable	338
		Input	337
D3	P03	Output	336
		Output enable	335
		Input	334
C2	P02	Output	333
		Output enable	332
		Input	331
D4	P01	Output	330
		Output enable	329
		Input	328
D1	P00	Output	327
		Output enable	326
		Input	325
D2	PF5	Output	324
		Output enable	323
		Input	322
E3	PJ5	Output	321
		Output enable	320
		Input	319
F3	PJ3	Output	318
		Output enable	317
		Input	316
G3	MD	Output	315
		Output enable	314
		Input	313
H4	P35	Input	309
J2	P33	Output	305
		Output enable	304
		Input	303
J3	P32	Output	302
		Output enable	301
		Input	300
L1	P25	Output	287
		Output enable	286
		Input	285
L4	P24	Output	284
		Output enable	283
		Input	282
L2	P23	Output	281
		Output enable	280
		Input	279

Table 48.7 Boundary Scan Register 145-Pin TFLGA (0.50-mm pitch) (2/8)

Pin No.	Pin Name	Input/Output	Bit Name
M1	P22	Output	278
		Output enable	277
		Input	276
N1	P21	Output	275
		Output enable	274
		Input	273
N2	P20	Output	272
		Output enable	271
		Input	270
M2	P17	Output	269
		Output enable	268
		Input	267
N3	P87	Output	266
		Output enable	265
		Input	264
L3	P16	Output	263
		Output enable	262
		Input	261
M3	P86	Output	260
		Output enable	259
		Input	258
K4	P15	Output	257
		Output enable	256
		Input	255
N4	P14	Output	254
		Output enable	253
		Input	252
L5	P13	Output	251
		Output enable	250
		Input	249
M4	P12	Output	248
		Output enable	247
		Input	246
N5	PH2/ USB0_DM	Output	245
		Output enable	244
		Input	243
N6	PH1/ USB0_DP	Output	242
		Output enable	241
		Input	240
L6	P56	Output	239
		Output enable	238
		Input	237
N7	P55	Output	236
		Output enable	235
		Input	234

Table 48.7 Boundary Scan Register 145-Pin TFLGA (0.50-mm pitch) (3/8)

Pin No.	Pin Name	Input/Output	Bit Name
K5	P54	Output	233
		Output enable	232
		Input	231
K6	P53	Output	230
		Output enable	229
		Input	228
L7	P52	Output	227
		Output enable	226
		Input	225
K7	P51	Output	224
		Output enable	223
		Input	222
M7	P50	Output	221
		Output enable	220
		Input	219
L8	P83	Output	218
		Output enable	217
		Input	216
N9	PC7	Output	215
		Output enable	214
		Input	213
M8	PC6	Output	212
		Output enable	211
		Input	210
L9	PC5	Output	209
		Output enable	208
		Input	207
N10	P82	Output	206
		Output enable	205
		Input	204
M9	P81	Output	203
		Output enable	202
		Input	201
K9	P80	Output	200
		Output enable	199
		Input	198
L10	PC4	Output	197
		Output enable	196
		Input	195
N11	PC3	Output	194
		Output enable	193
		Input	192
M10	P77	Output	191
		Output enable	190
		Input	189

Table 48.7 Boundary Scan Register 145-Pin TFLGA (0.50-mm pitch) (4/8)

Pin No.	Pin Name	Input/Output	Bit Name
K10	P76	Output	188
		Output enable	187
		Input	186
L11	PC2	Output	185
		Output enable	184
		Input	183
N12	P75	Output	182
		Output enable	181
		Input	180
N13	P74	Output	179
		Output enable	178
		Input	177
M12	PC1	Output	176
		Output enable	175
		Input	174
M11	PC0	Output	173
		Output enable	172
		Input	171
L12	P73	Output	170
		Output enable	169
		Input	168
K11	PB7	Output	167
		Output enable	166
		Input	165
K12	PB6	Output	164
		Output enable	163
		Input	162
K13	PB5	Output	161
		Output enable	160
		Input	159
J11	PB4	Output	158
		Output enable	157
		Input	156
J10	PB3	Output	155
		Output enable	154
		Input	153
J12	PB2	Output	152
		Output enable	151
		Input	150
J13	PB1	Output	149
		Output enable	148
		Input	147
H10	P72	Output	146
		Output enable	145
		Input	144

Table 48.7 Boundary Scan Register 145-Pin TFLGA (0.50-mm pitch) (5/8)

Pin No.	Pin Name	Input/Output	Bit Name
H11	P71	Output	143
		Output enable	142
		Input	141
H12	PB0	Output	140
		Output enable	139
		Input	138
H13	PA7	Output	137
		Output enable	136
		Input	135
G11	PA6	Output	134
		Output enable	133
		Input	132
G10	PA5	Output	131
		Output enable	130
		Input	129
G13	PA4	Output	128
		Output enable	127
		Input	126
F10	PA3	Output	125
		Output enable	124
		Input	123
F13	PA2	Output	122
		Output enable	121
		Input	120
F12	PA1	Output	119
		Output enable	118
		Input	117
E10	PA0	Output	116
		Output enable	115
		Input	114
E13	P67	Output	113
		Output enable	112
		Input	111
E11	P66	Output	110
		Output enable	109
		Input	108
E12	P65	Output	107
		Output enable	106
		Input	105
D10	PE7	Output	104
		Output enable	103
		Input	102
D13	PE6	Output	101
		Output enable	100
		Input	99

Table 48.7 Boundary Scan Register 145-Pin TFLGA (0.50-mm pitch) (6/8)

Pin No.	Pin Name	Input/Output	Bit Name
C12	P70	Output	98
		Output enable	97
		Input	96
D12	PE5	Output	95
		Output enable	94
		Input	93
B13	PE4	Output	92
		Output enable	91
		Input	90
A13	PE3	Output	89
		Output enable	88
		Input	87
B12	PE2	Output	86
		Output enable	85
		Input	84
A12	PE1	Output	83
		Output enable	82
		Input	81
C11	PE0	Output	80
		Output enable	79
		Input	78
D9	P64	Output	77
		Output enable	76
		Input	75
C10	P63	Output	74
		Output enable	73
		Input	72
A11	P62	Output	71
		Output enable	70
		Input	69
B11	P61	Output	68
		Output enable	67
		Input	66
D8	P60	Output	65
		Output enable	64
		Input	63
C9	PD7	Output	62
		Output enable	61
		Input	60
A9	PD6	Output	59
		Output enable	58
		Input	57
D7	PD5	Output	56
		Output enable	55
		Input	54

Table 48.7 Boundary Scan Register 145-Pin TFLGA (0.50-mm pitch) (7/8)

Pin No.	Pin Name	Input/Output	Bit Name
B9	PD4	Output	53
		Output enable	52
		Input	51
C8	PD3	Output	50
		Output enable	49
		Input	48
A8	PD2	Output	47
		Output enable	46
		Input	45
C7	PD1	Output	44
		Output enable	43
		Input	42
B8	PD0	Output	41
		Output enable	40
		Input	39
D6	P93	Output	38
		Output enable	37
		Input	36
A7	P92	Output	35
		Output enable	34
		Input	33
B7	P91	Output	32
		Output enable	31
		Input	30
A6	P90	Output	29
		Output enable	28
		Input	27
B6	P47	Output	26
		Output enable	25
		Input	24
C5	P46	Output	23
		Output enable	22
		Input	21
A5	P45	Output	20
		Output enable	19
		Input	18
E5	P44	Output	17
		Output enable	16
		Input	15
B5	P43	Output	14
		Output enable	13
		Input	12
A4	P42	Output	11
		Output enable	10
		Input	9

Table 48.7 Boundary Scan Register 145-Pin TFLGA (0.50-mm pitch) (8/8)

Pin No.	Pin Name	Input/Output	Bit Name
C4	P41	Output	8
		Output enable	7
		Input	6
A3	P40	Output	5
		Output enable	4
		Input	3
A2	P07	Output	2
		Output enable	1
		Input	0
To TDO			

Table 48.8 Boundary Scan Register 64-Pin TFBGA (1/3)

Pin No.	Pin Name	Input/Output	Bit Name
From TDI			
C3	MD	Output	315
		Output enable	314
		Input	313
F3	P35	Input	309
H2	P17	Output	269
		Output enable	268
		Input	267
H3	P16	Output	263
		Output enable	262
		Input	261
E4	P13	Output	251
		Output enable	250
		Input	249
F4	P12	Output	248
		Output enable	247
		Input	246
H4	PH2/ USB0_DM	Output	245
		Output enable	244
		Input	243
H5	PH1/ USB0_DP	Output	242
		Output enable	241
		Input	240
F5	P53	Output	230
		Output enable	229
		Input	228
G6	PC7	Output	215
		Output enable	214
		Input	213
H6	PC6	Output	212
		Output enable	211
		Input	210
G7	PC5	Output	209
		Output enable	208
		Input	207
H7	PC4	Output	197
		Output enable	196
		Input	195
H8	PC1	Output	176
		Output enable	175
		Input	174
G8	PC0	Output	173
		Output enable	172
		Input	171

Table 48.8 Boundary Scan Register 64-Pin TFBGA (2/3)

Pin No.	Pin Name	Input/Output	Bit Name
F6	PB7	Output	167
		Output enable	166
		Input	165
F7	PB6	Output	164
		Output enable	163
		Input	162
F8	PB5	Output	161
		Output enable	160
		Input	159
E6	PA7	Output	137
		Output enable	136
		Input	135
D6	PA6	Output	134
		Output enable	133
		Input	132
D8	PA4	Output	128
		Output enable	127
		Input	126
D7	PA2	Output	122
		Output enable	121
		Input	120
C7	PA1	Output	119
		Output enable	118
		Input	117
C8	PE7	Output	104
		Output enable	103
		Input	102
B8	PE6	Output	101
		Output enable	100
		Input	99
A8	PE2	Output	86
		Output enable	85
		Input	84
B7	PE1	Output	83
		Output enable	82
		Input	81
A7	PE0	Output	80
		Output enable	79
		Input	78
A6	PD7	Output	62
		Output enable	61
		Input	60
B6	PD6	Output	59
		Output enable	58
		Input	57

Table 48.8 Boundary Scan Register 64-Pin TFBGA (3/3)

Pin No.	Pin Name	Input/Output	Bit Name
C6	PD5	Output	56
		Output enable	55
		Input	54
C5	PD4	Output	53
		Output enable	52
		Input	51
B5	PD3	Output	50
		Output enable	49
		Input	48
A5	PD2	Output	47
		Output enable	46
		Input	45
D5	P43	Output	14
		Output enable	13
		Input	12
B4	P42	Output	11
		Output enable	10
		Input	9
C4	P41	Output	8
		Output enable	7
		Input	6
D4	P40	Output	5
		Output enable	4
		Input	3
To TDO			

48.3 Operations

The boundary scan functionality is valid when the RES# pin is driven high, the EMLE pin is driven low, and the BSCANP pin is driven high.

48.3.1 TAP Controller

Figure 48.2 shows the state transition diagram of the TAP controller.

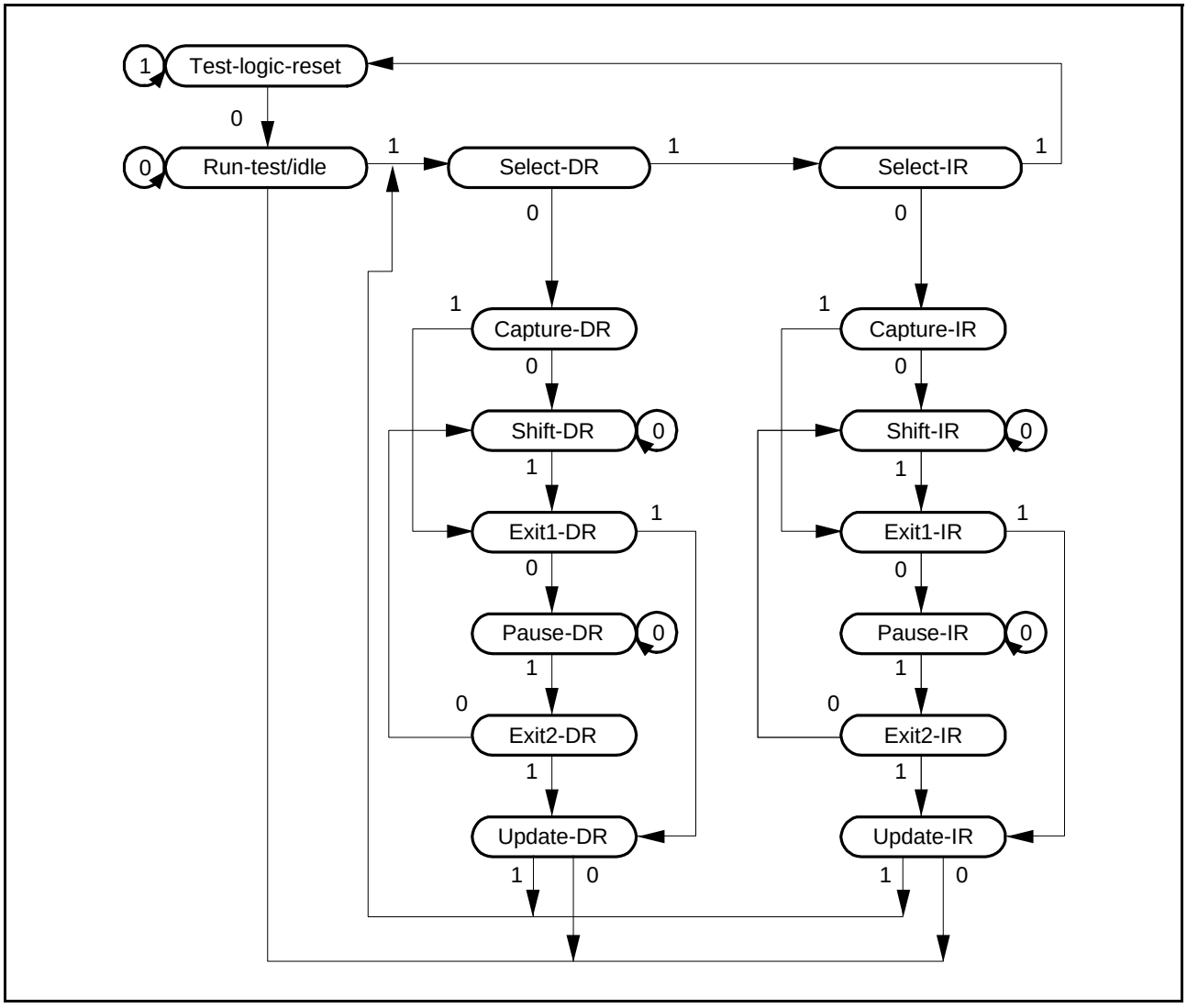


Figure 48.2 State Transition Diagram of TAP Controller

48.3.2 List of Commands

(1) BYPASS [Instruction Code: 1111 1111b]

The BYPASS instruction is an instruction that drives the bypass register (JTBPR). This instruction shortens the shift path, facilitating the transfer of serial data to other LSIs on a printed-circuit board at higher speeds. While this instruction is being executed, the test circuit has no effect on the system circuits.
The bypass register (JTBPR) is connected between the TDI and TDO pins. Bypass operation is initiated from shift-DR operation. The TDO is low in the first clock cycle in the shift-DR state; in the subsequent clock cycles, the TDI signal is output on the TDO pin.

(2) EXTEST [Instruction Code: 0000 0000b]

The EXTEST instruction is used to test external circuits when this LSI is installed on the printed circuit board. If this instruction is executed, output pins are used to output test data (specified by the SAMPLE/PRELOAD instruction) from the boundary scan register to the print circuit board, and input pins are used to input test result.

(3) SAMPLE/PRELOAD [Instruction Code: 0100 0000b]

The SAMPLE/PRELOAD instruction is used to input data from the LSI internal circuits to the boundary scan register, output data from scan path, and reload the data to the scan path. While this instruction is executed, input signals are directly input to the LSI and output signals are also directly output to the external circuits. The LSI system circuit is not affected by this instruction.
In SAMPLE operation, the boundary scan register latches the snap shot of data transferred from input pins to internal circuit or data transferred from internal circuit to output pins. The latched data is read from the scan path. The scan register latches the snap data at the rising edge of the TCK in Capture-DR state. The scan register latches snap shot without affecting the LSI normal operation.
In PRELOAD operation, initial value is written from the scan path to the parallel output latch of the boundary scan register prior to the EXTEST instruction execution. If the EXTEST is executed without executing this PRELOAD operation, undefined values are output from the beginning to the end (transfer to the output latch) of the EXTEST sequence. (In EXTEST instruction, output parallel latches are always output to the output pins.)

(4) IDCODE [Instruction Code: 0101 0101b]

When the IDCODE instruction is selected, IDCODE register value is output to the TDO in Shift-DR state of the TAP controller. In this case, IDCODE register value is output from the LSB. During this instruction execution, test circuit does not affect the system circuit. JTIR is initialized by the IDCODE instruction in Test-Logic-Reset state of the TAP controller.

(5) CLAMP [Instruction Code: 1101 0000b]

When the CLAMP instruction is selected, output pins output the boundary scan register value which was specified by the SAMPLE/PRELOAD instruction in advance. While the CLAMP instruction is selected, the status of boundary scan register is maintained regardless of the TAP controller state.
BYPASS is connected between TDI and TDO, the same operation as BYPASS instruction can be achieved.

(6) HIGHZ [Instruction Code: 1000 0000b]

When the HIGHZ instruction is selected, all output pins enter high-impedance state. While the HIGHZ instruction is selected, the status of boundary scan register is maintained regardless of the state of the TAP controller.
BYPASS is connected between TDI and TDO pins, leading to the same operation as when the BYPASS instruction has been selected.

48.4 Usage Notes

(1) Pin serial transfer, data are input or output in LSB order (refer to Figure 48.3).

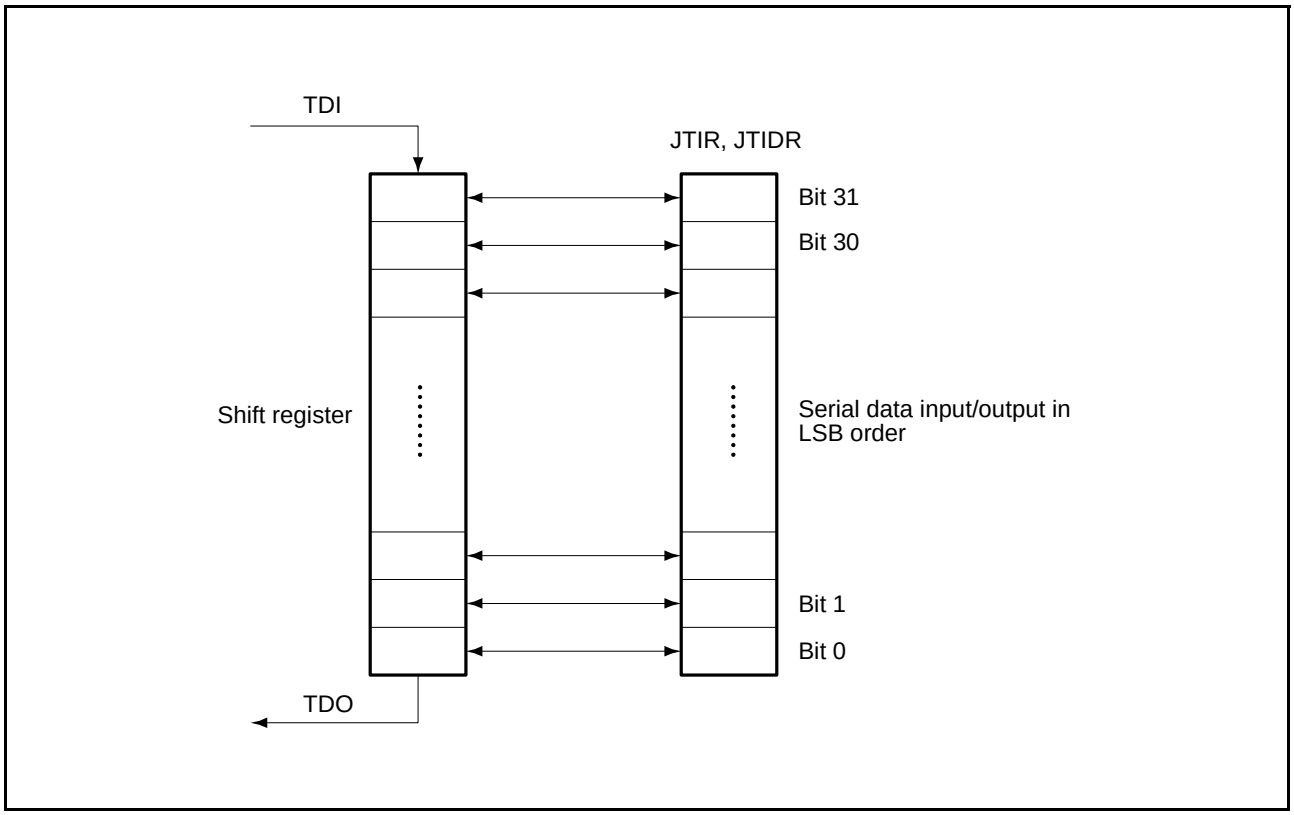


Figure 48.3 Serial Data Input/Output

(2) Pins of the boundary scan (TCK, TDI, TMS, and TRST#) have to be pulled up by pull-up resistors.
However, handle the TRST# pin in the way described in the manual for the given on-chip emulator if an on-chip emulator is in use.
If the TRST# pin is pulled down but a boundary scan is to proceed, ensure that the TRST# pin is also controllable.

(3) Power supply pins (VCC, VCL, VSS, AVCC0, AVCC1, AVSS0, AVSS1, VCC_USB, VSS_USB, VBATT) cannot be boundary-scanned.

(4) Analog reference pins (VREFH0, VREFL0) cannot be boundary-scanned.

(5) Clock pins (EXTAL, XTAL, XCIN, and XCOUT) cannot be boundary-scanned.

(6) Reset signal (RES#) cannot be boundary-scanned.

(7) USB dedicated pins (USB1_DP, USB1_DM) cannot be boundary-scanned.

(8) The on-chip emulator enable pin (EMLE) cannot be boundary-scanned.

(9) The boundary-scan pin (BSCANP) cannot be boundary-scanned.

(10) The boundary-scan pins (TCK, TMS, TRST#, TDI, and TDO) cannot be boundary-scanned.

(11) The boundary-scan facility is not available when the chip is in the states below.

- Reset state
- Software standby or deep software standby

(12) For a pin that incorporates open-drain functionality and for which the open-drain function is enabled, if the boundary-scan function sets the corresponding bit in the output scan register and output enable register to 1, executing an EXTEST, CLAMP, or SAMPLE/PRELOAD instruction makes the pin output the high level rather than placing it in the high-impedance state.

(13) Be sure to satisfy the standards for the boundary scan function when multiplex ports are used. Figure 48.4 (1) shows the pin configuration of the port pins on which the RIIC and TSCAP pin functions (P16, P17, P20, P21, and

PC4) are multiplexed. When the boundary scan function is to be used while pins P16, P17, P20, and P21 are in use as the RIIC pins (SCL1, SCL2, SDA1, and SDA2) or the PC4 pin is in use as a CTSU pin (TSCAP), contention with the open-drain outputs or sneak current might be generated.

(14) Figure 48.4 (2) shows the pin configuration of the pins P00 to P02, P40 to P47, P90, PD0 to PD7, PE0, and PE1. When the boundary scan function is used with pins P00 to P02, P40 to P47, P90, PD0 to PD7, PE0, and PE1 to be used as AD input pins (AN000 to AN007, ANEX0, ANEX1, and AN100 to AN111), the conflict with the AD input or sneak current might be generated.

(15) As the TS pins are multiplexed with port pin functions, make sure the standards for the boundary scan function are satisfied. Figure 48.4 (3) shows the pin configuration of the port pins on which the TS pin functions are multiplexed (P2[7:0], P14, P15, P33, P34, P53, PC0, PC1, PC5, and PC6). When the boundary scan function is to be used while pins P2[7:0], P14, P15, P33, P34, P53, PC0, PC1, PC5 and PC6 are in use as the TS pins (TS0 to TS16), the conflict with the TS outputs or sneak current might be generated.

(16) As the USB0 pins are multiplexed with port pin functions, make sure the standards for the boundary scan function are satisfied. Figure 48.4 (4) shows the pin configuration of the port pins on which the USB0 pin functions are multiplexed (PH1 and PH2). When the boundary scan function is to be used while pins PH1 and PH2 are in use as the USB0 pins (USB0_DP and USB0_DM), the conflict with the USB0 output signals or sneak current might be generated.

(17) The HIGHZ mode option of the MD pin is not available.

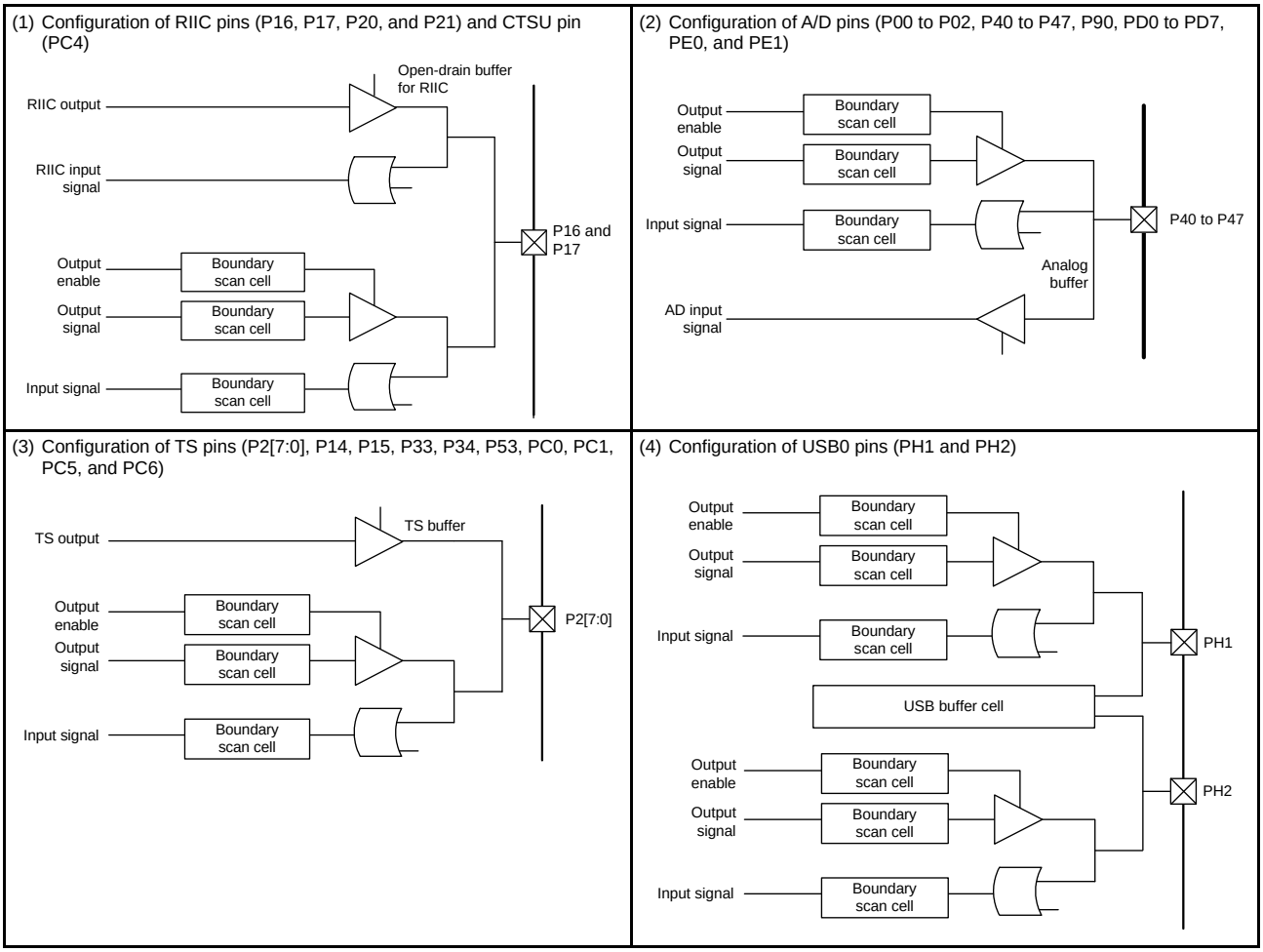


Figure 48.4 Pin Configuration

49. Trusted Secure IP (TSIP)

This MCU incorporates a Trusted Secure IP module to provide security functions. The module consists of an access management circuit, encryption engine, and random number generator. In combination with the Trusted Secure IP library, the Trusted Secure IP can prevent eavesdropping (confidentiality), falsification of information (integrity), and impersonation (authenticity).
Key information to be used in encrypting and decrypting data is only stored within the Trusted Secure IP, and any external access can be shut out to obtain a system with strong security.

49.1 Overview

Table 49.1 summarizes the specifications of the Trusted Secure IP. Figure 49.1 shows a block diagram of the Trusted Secure IP.

Table 49.1 Specifications of Trusted Secure IP (1/2)

Item	Description
Access control	Access management circuit • In case of irregular access to the Trusted Secure IP due to a falsified program or runaway execution of a program, this circuit blocks all subsequent access and stops the output of data from the Trusted Secure IP.
Encryption engine	AES: Compliant with NIST FIPS PUB 197 algorithm • Key sizes: 128, 192, or 256 bits • Block sizes: 128 bits • Block cipher mode of operation ECB, CBC, CTR: Compliant with NIST SP 800-38A CMAC: Compliant with NIST SP 800-38B CCM: Compliant with NIST SP 800-38C GCM: Compliant with NIST SP 800-38D XTS: Compliant with NIST SP 800-38E GCTR • Number of cycles for execution*1 ECB, CBC, CTR, CMAC, GCTR, XTS: 11 cycles of PCLKB for 128-bit keys, 13 cycles of PCLKB for 192-bit keys, 15 cycles of PCLKB for 256-bit keys CCM: 22 cycles of PCLKB for 128-bit keys, 26 cycles of PCLKB for 192-bit keys, 30 cycles of PCLKB for 256-bit keys AES-GCM • AES-GCM is realized by combining AES-GCTR and GHASH. RSA • Key sizes: Up to 2048 bits • Block sizes: Up to 2048 bits • Number of cycles for execution: Approximately 1,300,000 cycles of PCLKB when the CRT is used*1 TDES • Key sizes: 56 bits, 2 × 56 bits, or 3 × 56 bits • Block sizes: 64 bits • Block cipher mode of operation: ECB, CBC • Number of cycles for execution*1 16 cycles of PCLKB for 56-bit keys, 32 cycles of PCLKB for 2 × 56-bit keys, 48 cycles of PCLKB for 3 × 56-bit keys ARC4 • Key sizes: 2048 bits • Block sizes: 128 bits • Number of cycles for execution: 16 cycles of PCLKB*1 ECC • Key sizes: Up to 256 bits • Block sizes: 256 bits HASH Support for SHA1, SHA224/SHA256/MD5, GHASH • Block sizes: 512bits • Number of cycles for execution*1 SHA1: 80 cycles of PCLKB SHA224/SHA256/MD5: 64 cycles of PCLKB GHASH: 9 cycles of PCLKB

Table 49.1 Specifications of Trusted Secure IP (2/2)

Item	Description
Encryption engine	Key management • Keys are only valid within the Trusted Secure IP. • Only key generation information is output from the Trusted Secure IP. • Keys can be regenerated by the input of key generation information to the Trusted Secure IP. Endian • Big or little
Generation of random numbers	32-bit true random number generator • The Trusted Secure IP library can assemble 32-bit true random numbers to generate 128- or 256-bit true random numbers. • The generated 128-bit and 256-bit true random numbers are used as keys in encrypting and decrypting data.
Protection against illicit key copying	• An ID unique to the MCU (unique ID) is accessible from the access management circuit through the dedicated bus. • Combining the unique ID with the key generation information prevents the illicit copying of the key to another MCU.
Supervisor mode	• The supervisor mode signal is connected to the access management circuit and is used to limit control of the Trusted Secure IP module to supervisor mode only.
Interrupt sources	11
Low power consumption	Setting of the module stop state is possible.

Note 1. This does not include the overhead for calling functions of the Trusted Secure IP library.

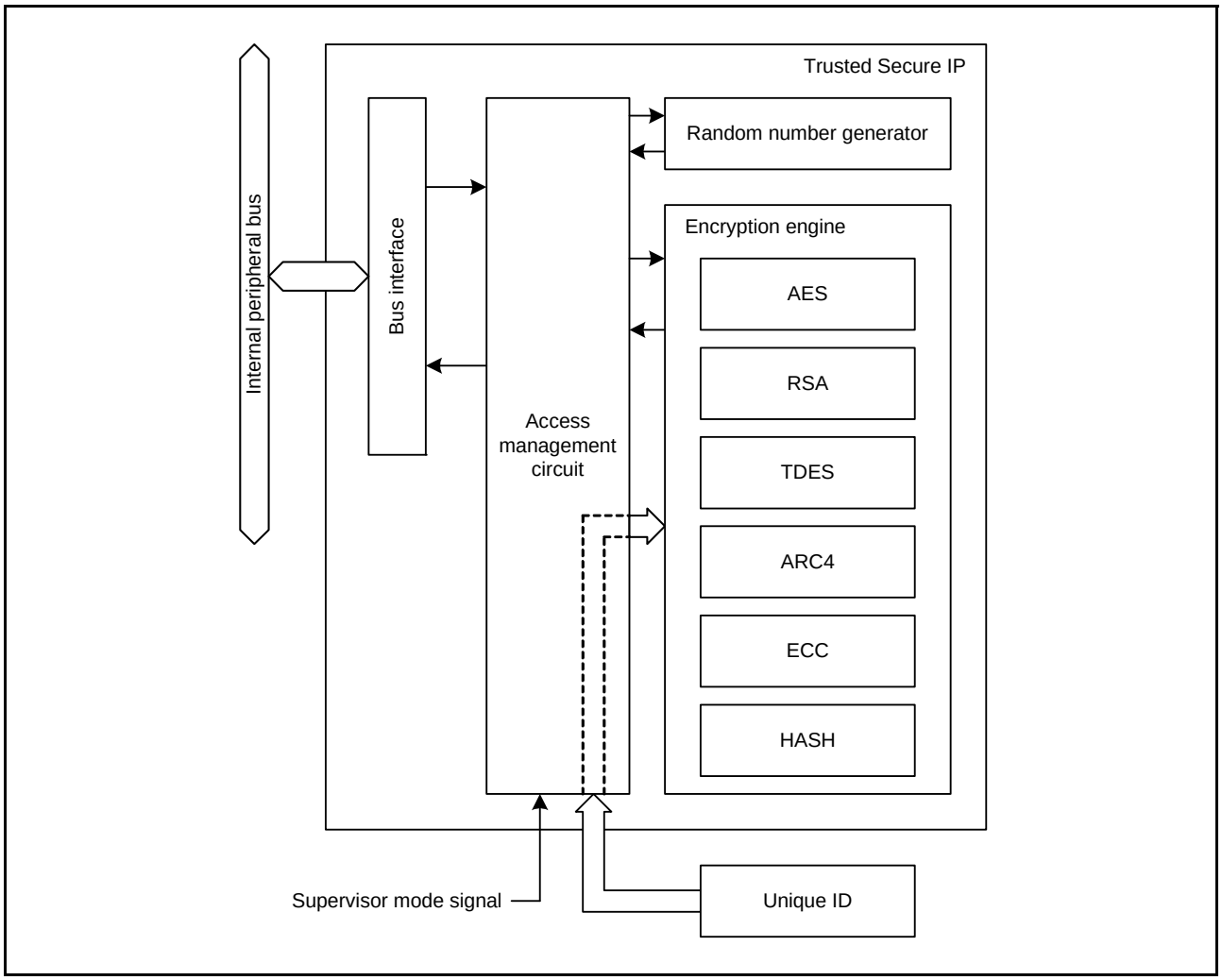


Figure 49.1 Trusted Secure IP Block Diagram

49.2 Operation

49.2.1 Operating Modes and State Transitions

Figure 49.2 shows the state transitions of the Trusted Secure IP.
Use of the Trusted Secure IP security functions is only possible through use of the Trusted Secure IP library provided by Renesas Electronics, in accordance with the state transitions as shown in the figure below.
When irregular access to the Trusted Secure IP (access that violates the defined procedure) due to a falsified program or a program entering runaway execution, etc. is attempted, the access management circuit does not accept any subsequent access and stops the output of any data from the Trusted Secure IP.

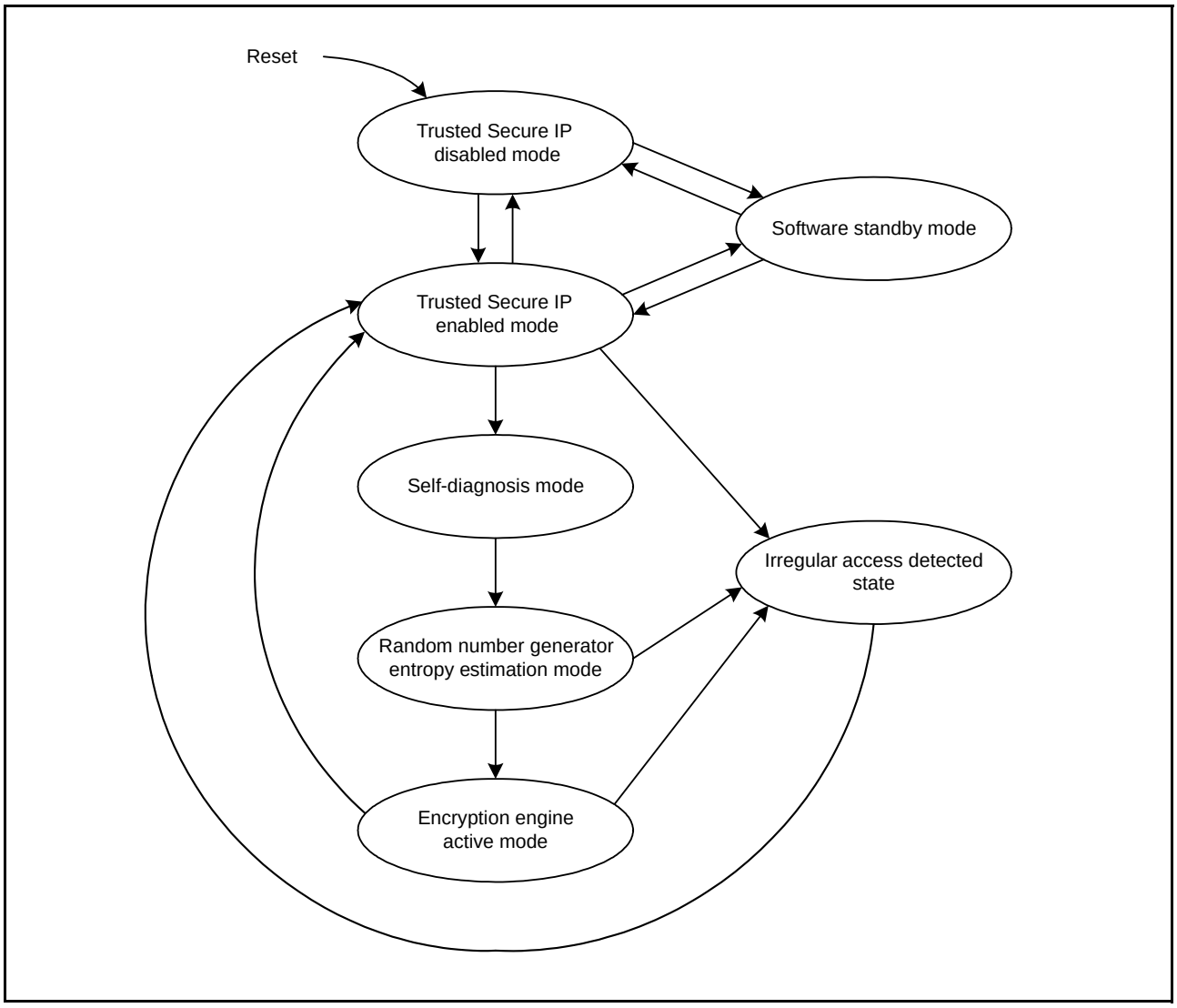


Figure 49.2 Trusted Secure IP Operating Modes and State Transitions

Many of the security functions that the Trusted Secure IP offers are applicable only in the encryption engine active mode. The operations that can be performed in this mode are given below.

(1) Key Installation
(2) Encryption and decryption
(3) Key generation
(4) Random number generation

49.2.2 Encryption Engine

Figure 49.3 shows processes of the encryption engine integrated in the Trusted Secure IP.
The encryption engine, using the key generation information, performs plaintext to ciphertext encryption and ciphertext to plaintext decryption by hardware.
In no part of the encryption or decryption process, is key data or intermediate data ever exposed outside of the Trusted Secure IP.

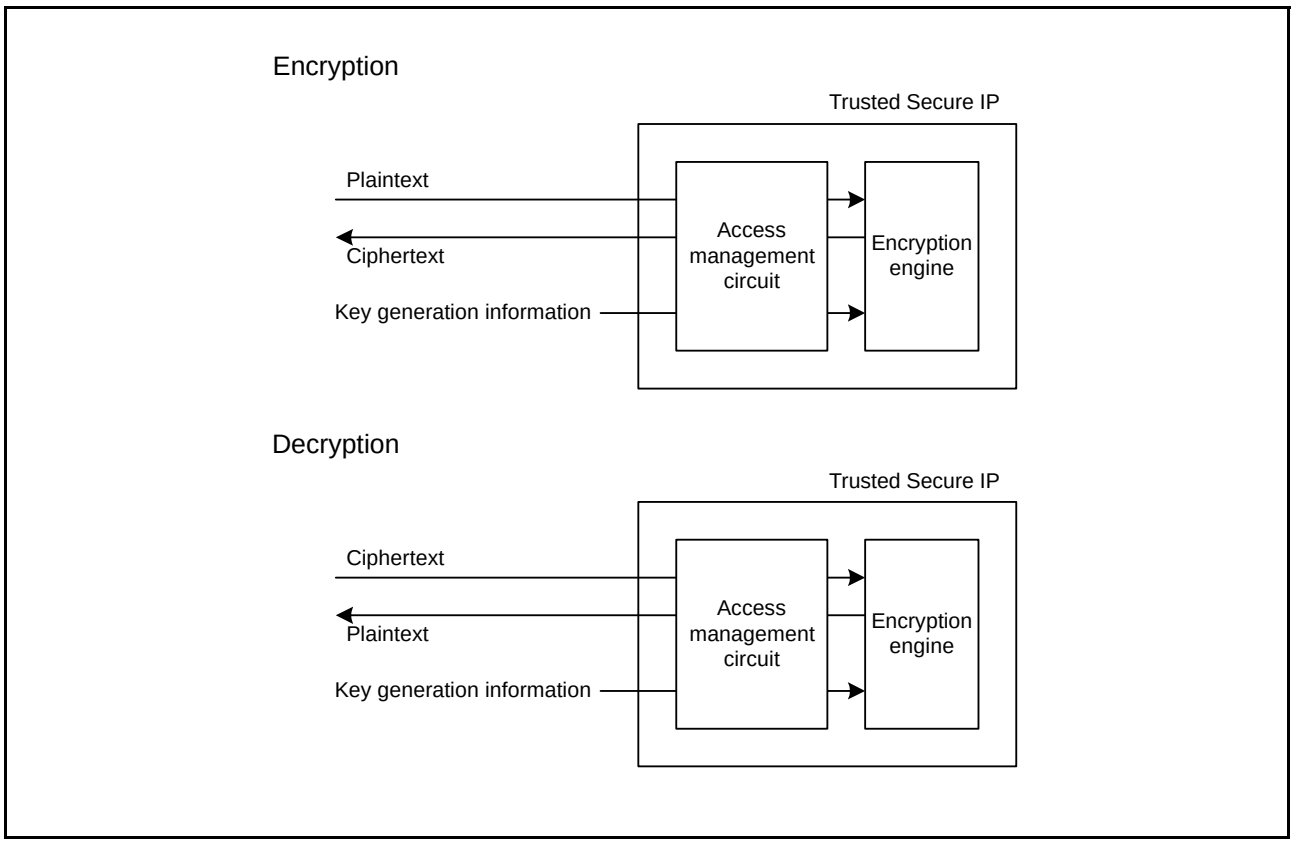


Figure 49.3 Encryption and Decryption processes by Encryption Engine

49.2.3 Key Installation

The key installation is the operation that safely converts the user key to the key generation information and stores it in flash memory. The procedure for installing the key data are given below.

(1) The user uses the key (Key-2) used for encrypting the user key to encrypt the user key (Key-1) producing eKey-1.
(2) The user sends the encrypted user key (eKey-1) to the Trusted Secure IP over the serial interface.
(3) The key generation information of the Key-2 (Index-2) contained in the Trusted Secure IP library is used to recover the Key-2, which is then used to decrypt the user key.
(4) The user key is converted to user key generation information (Index-1) using the unique ID and a random number, and stored in flash memory.

The installation process and flow chart are given in Figure 49.4 and Figure 49.5, respectively.
Once the key data is installed, the user key generation information (Index-1) can then be used to perform encryption or decryption.

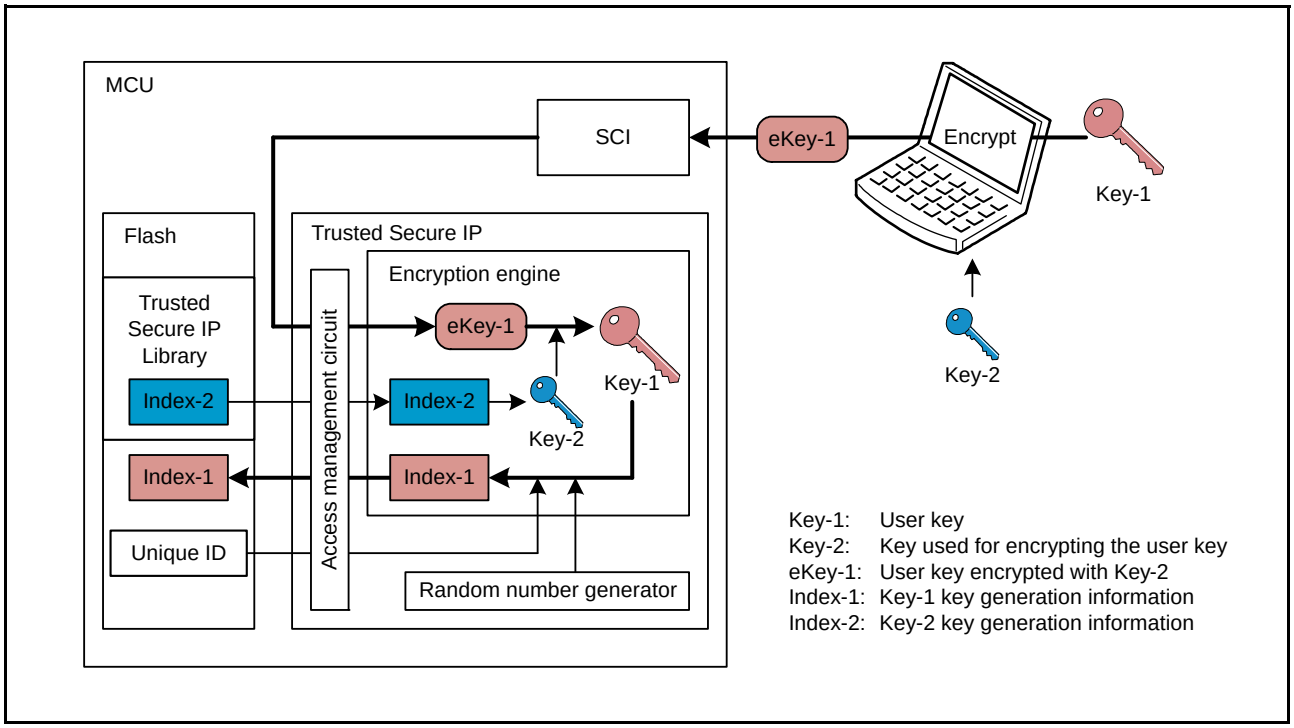


Figure 49.4 Key Installation Process

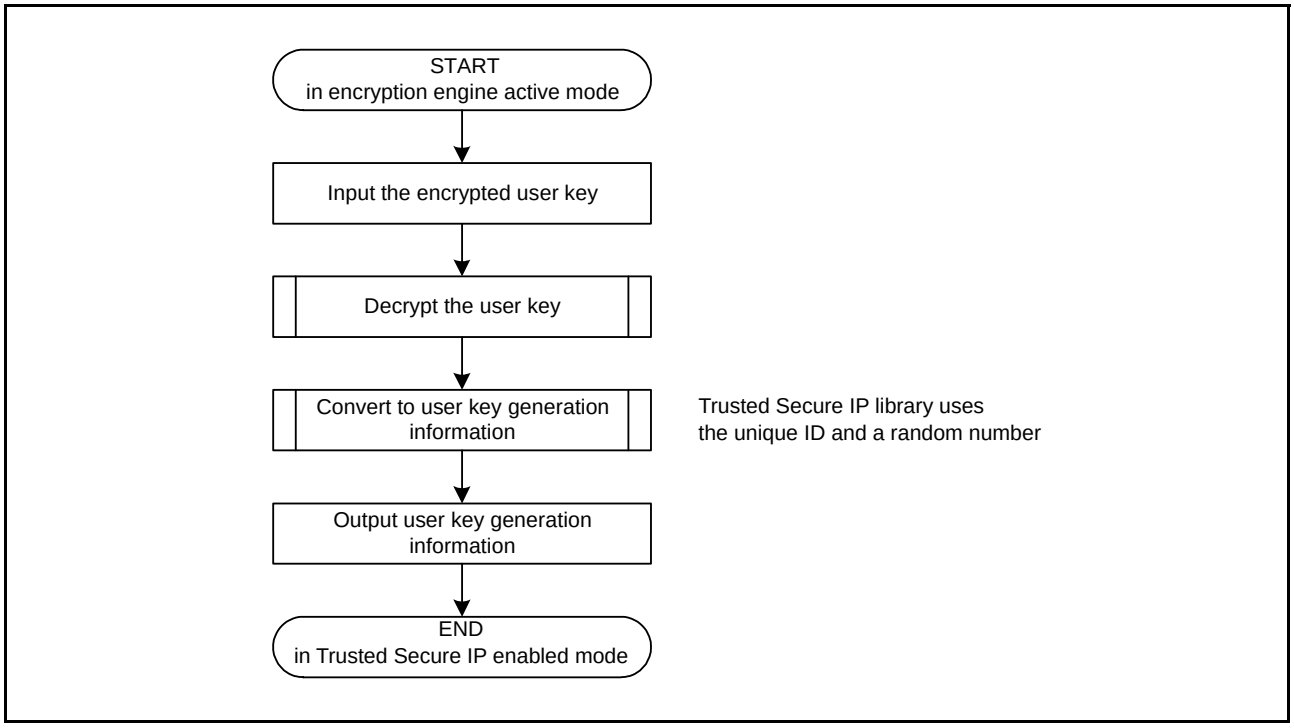


Figure 49.5 Key Installation Flow Chart

49.2.4 Encryption and Decryption

The procedures for encrypting and decrypting data are given below.

(1) Input the key generation information into the Trusted Secure IP, and recover the key data.
(2) Input the data to encrypt or decrypt into the Trusted Secure IP. This converts plaintext into ciphertext, and ciphertext into plaintext.
(3) Read the converted data.

The encryption engine has an input buffer and an output buffer, enabling encryption/decryption to proceed in parallel with data input/output.
Figure 49.6, Figure 49.7, and Figure 49.8 show the timing diagram, encryption flow, and decryption flow, respectively.

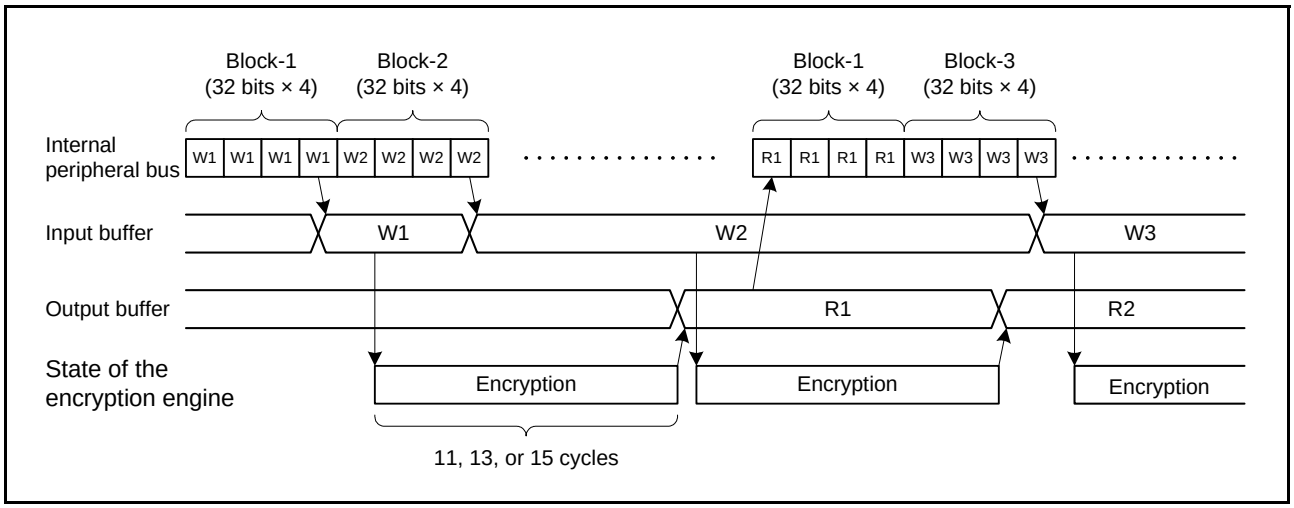


Figure 49.6 Encryption and Decryption Timing Diagram

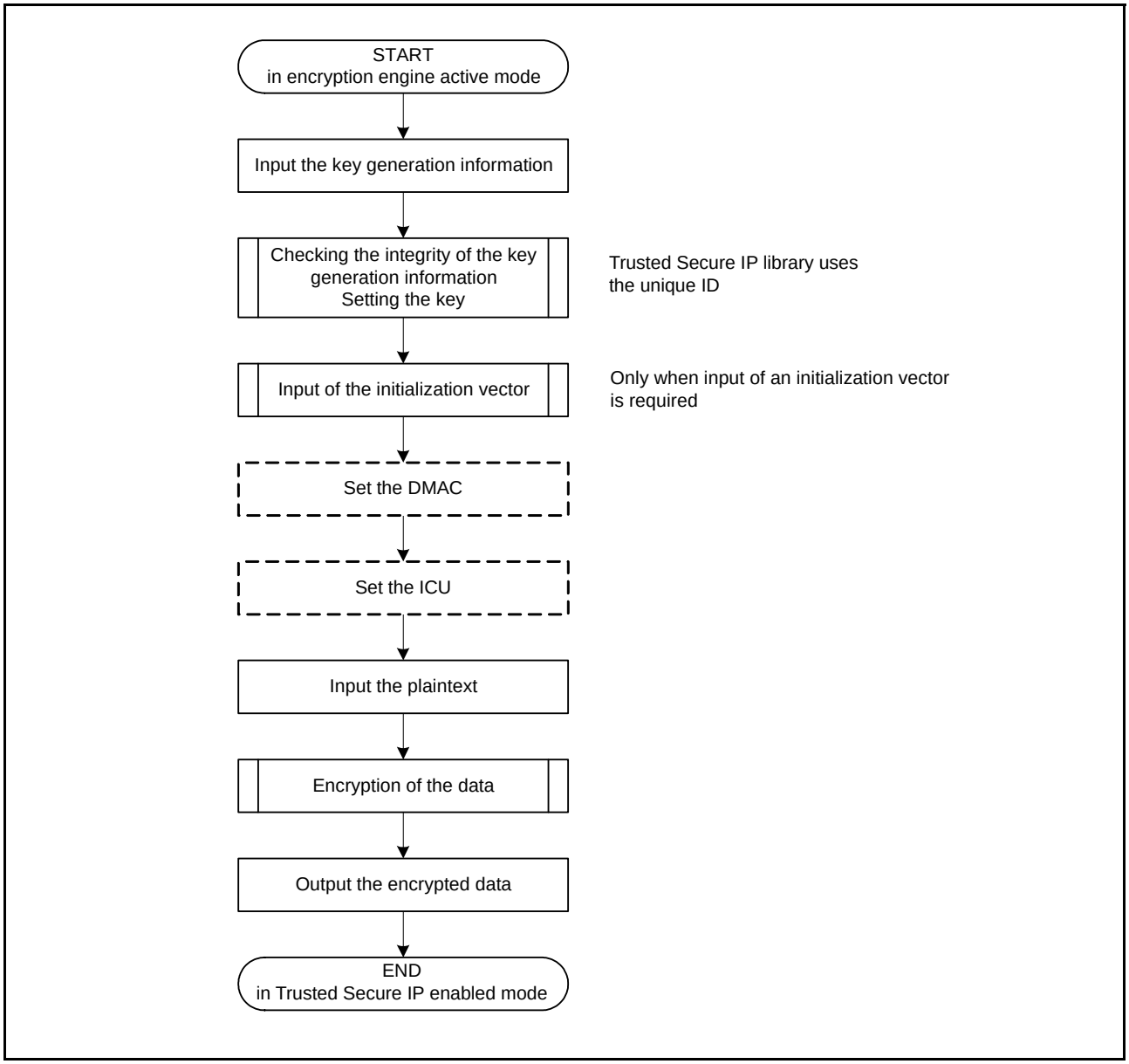


Figure 49.7 Encryption Flow Chart

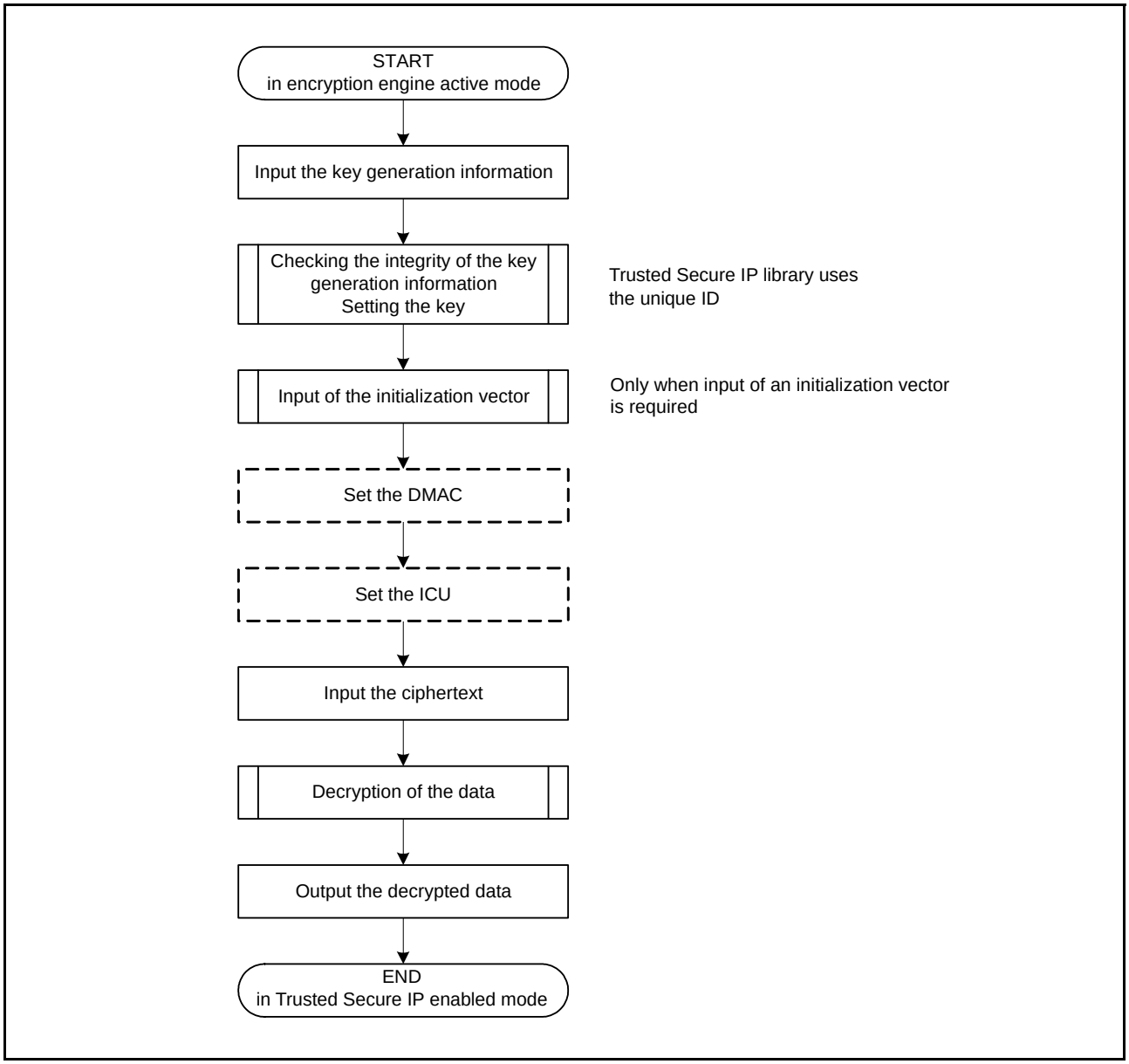


Figure 49.8 Decryption Flow Chart

49.2.5 Generating Key Generation Information (by Using Random Numbers)

Figure 49.9 shows the generating flow for the key generation information by using random numbers.

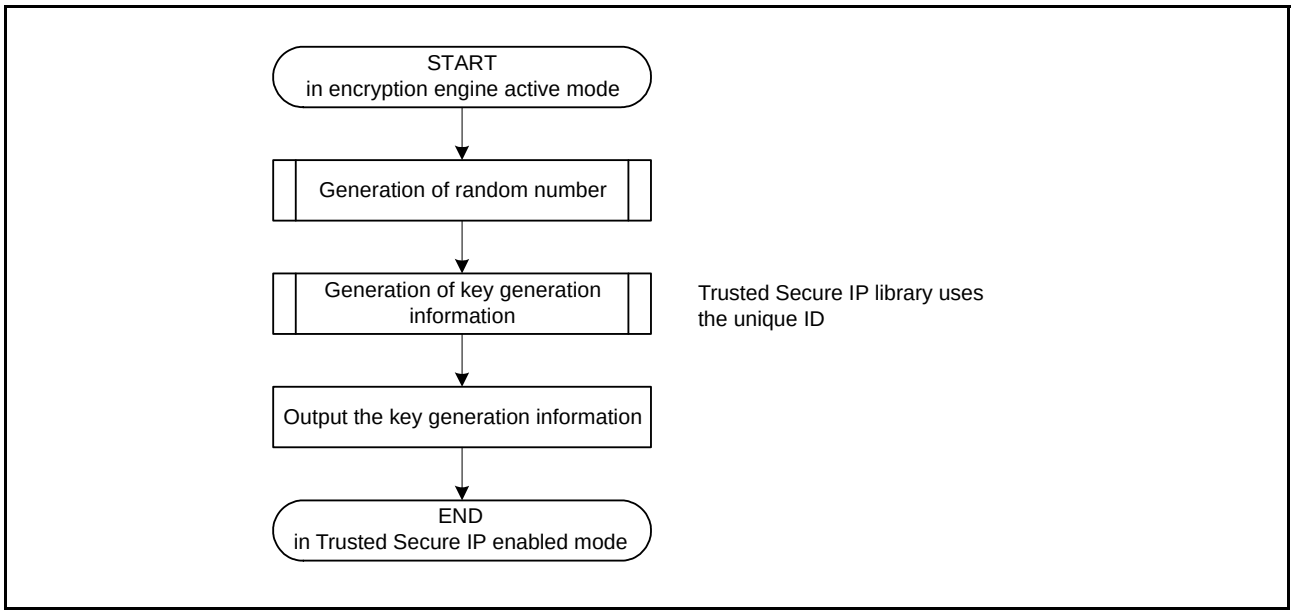


Figure 49.9 Key Generation Information Generating Flow Chart (Using Random Numbers)

49.2.6 Random Number Generation

Figure 49.10 shows the random number generation flow.

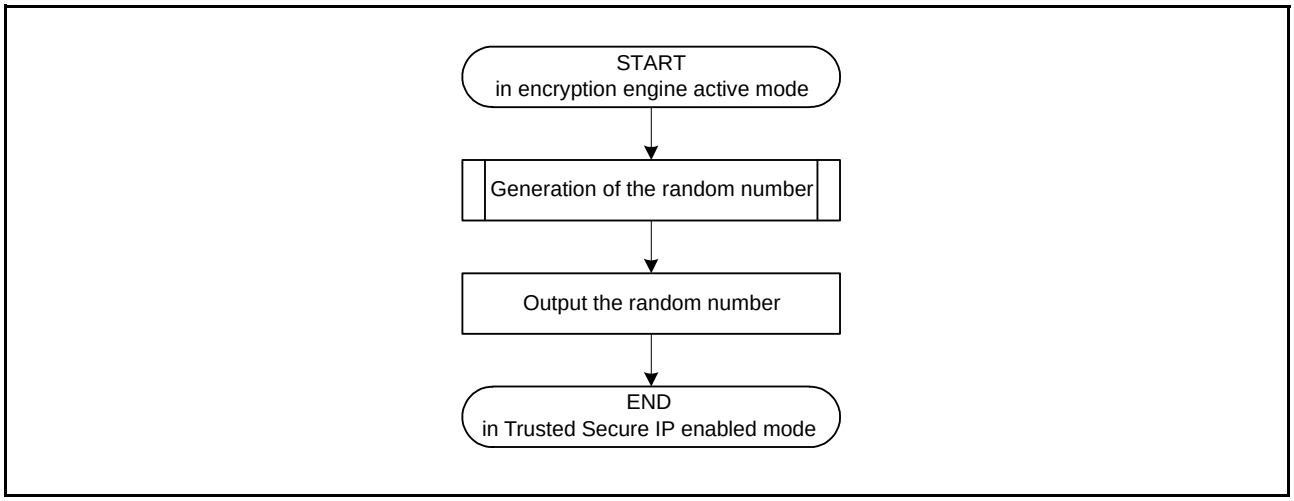


Figure 49.10 Random Number Generation Flow Chart

49.3 Interrupt

Table 49.2 lists the interrupt sources.
Trusted Secure IP library uses interrupts caused by these interrupt sources. Do not set the ICU.IERm.IENj bits corresponding to these interrupt sources to 0.

Table 49.2 Trusted Secure IP Interrupt Sources

Name	Interrupt Source
PROC_BUSY	Procedure completion interrupt
ROMOK	Falsification detection interrupt
LONG_PLG	Calculation completion interrupt
TEST_BUSY	Test busy
WRRDY0	Write ready 0
WRRDY1	Write ready 1
WRRDY4	Write ready 4
RDRDY0	Read ready 0
RDRDY1	Read ready 1
INTEGRATE_WRRDY	Integration write ready
INTEGRATE_RDRDY	Integration read ready

49.4 Usage Notes

49.4.1 Standby Mode

When standby mode is entered while the encryption engine is in processing, proper processing cannot be resumed after standby mode is exited. Standby mode should therefore be entered only after first entering Trusted Secure IP disabled mode or Trusted Secure IP enabled mode.

49.4.2 Setting the Module Stop Function

The module stop control register D (MSTPCRD) enables or disables operation of the Trusted Secure IP. After a reset, the Trusted Secure IP is stopped. After exiting the module stop state, the Trusted Secure IP can be accessed. Refer to section 11, Low Power Consumption for details.

49.4.3 Trusted Secure IP Library

Use of the Trusted Secure IP requires the Trusted Secure IP library provided by Renesas Electronics. Please contact our sales office for information regarding the Trusted Secure IP library.

50. 12-Bit A/D Converter (S12ADFa)

In this section, “PCLK” is used to refer to PCLKB.

50.1 Overview

This MCU incorporates two units of a 12-bit successive approximation A/D converter. Unit 0 for high-speed conversion can select analog input of up to 8 channels. Unit 1 for middle-speed conversion can select analog input of up to 12 channels, temperature sensor output, and internal reference voltage.

The 12-bit A/D converter converts analog input of up to 8 selected channels (unit 0) or up to 12 channels (unit 1), temperature sensor output, or internal reference voltage into a 12-bit digital value through successive approximation.

The A/D converter has three operating modes: single scan mode in which the analog inputs of eight arbitrarily selected channels (unit 0) and 12 channels (unit 1) are converted in ascending channel order; continuous scan mode in which the analog inputs of eight arbitrarily selected channels (unit 0) and 12 channels (unit 1) are continuously converted in ascending channel order; and group scan mode in which eight arbitrarily selected channels (unit 0) and 12 channels (unit 1) are arbitrarily divided into two groups (groups A and B) or three groups (groups A, B, and C) and converted in ascending channel order in each group.

In group scan mode, either two groups (groups A and B) or three groups (groups A, B, and C) is selected.

The conditions for scanning start of each group (A and B or A, B, and C) (synchronous trigger) can be independently selected, thus allowing A/D conversion of each group to be started independently.

During group priority operation, in addition to the above-mentioned operation, a trigger to start scanning for the priority group is accepted during scan for the low-priority group, and scan for the priority group is started after scan for the low-priority group was discontinued. The priority order is group A > group B > group C. Accordingly, as priority operation, when a trigger to start scanning for group B is accepted during scan for group C, group C scan is discontinued, and scan for group B is started. Likewise, when a trigger to start scanning for group A is accepted during scan for group C, group C scan is discontinued and scan for group A is started. In the same way, when a trigger to start scanning for group A is accepted during scan for group B, group B scan is discontinued and scan for group A is started.

The discontinued scan operation can be restarted after the scanning of the priority group is completed.

In double trigger mode, one analog input channel arbitrarily selected is converted in single scan mode or group scan mode (group A), and the resulting data of A/D conversion started by the first and second synchronous triggers are stored into different registers (duplication of A/D conversion data).

Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages internally generated in the 12-bit A/D converter is converted.

The temperature sensor output and internal reference voltage can be selected with analog input of the channels, and A/D conversion is performed in the order of the analog input of the channels, temperature sensor output, and internal reference voltage. A/D conversion of the extended analog input is independently performed.

The MCU incorporates the comparison function (with windows A and B). In addition, the value of A/D conversion and the reference value of the low side can be compared by a comparator.

Table 50.1 lists the specifications of the 12-bit A/D converter and Table 50.2 lists the functions of the 12-bit A/D converter. Figure 50.1 and Figure 50.2 show block diagrams of the 12-bit A/D converter.

Table 50.1 Specifications of 12-Bit A/D Converter (1/2)

Item	Description
Number of units	Two units (S12AD and S12AD1)
Input channels	Eight channels for S12AD and 12 channels for S12AD1 + 1 extension
Extended analog function	Temperature sensor output, internal reference voltage
A/D conversion method	Successive approximation method
Resolution	12 bits
Conversion time	0.48 µs per channel (12-bit conversion mode) 0.45 µs per channel (10-bit conversion mode) 0.42 µs per channel (8-bit conversion mode) (A/D conversion clock: when ADCLK operates at 60 MHz)
A/D conversion clock	Peripheral module clock PCLK*1 and A/D conversion clock ADCLK*1 can be set so that the frequency ratio should be one of the following. PCLK to ADCLK frequency ratio = 1:1, 2:1, 4:1, 8:1 ADCLK is set using the clock generation circuit.
Data registers	• 20 registers for analog input (eight for S12AD and 12 for S12AD1), 1 for A/D-converted data duplication in double trigger mode per unit, and 2 for A/D-converted data duplication during extended operation in double trigger mode per unit. • One register for temperature sensor (S12AD1) • One register for internal reference (S12AD1) • One register for self-diagnosis per unit • The results of A/D conversion are stored in 12-bit A/D data registers. • 8-, 10-, and 12-bit accuracy output for the results of A/D conversion • The value obtained by adding up A/D-converted results is stored as a value in the number of bit for conversion accuracy + 2 bits/4 bits*2 in the A/D data registers in A/D-converted value addition mode. • Double trigger mode (selectable in single scan and group scan modes): The first piece of A/D-converted analog-input data on one selected channel is stored in the data register for the channel, and the second piece is stored in the duplication register. • Extended operation in double trigger mode (available for specific triggers): A/D-converted analog-input data on one selected channel is stored in the duplication register that is prepared for each type of trigger.
Operating modes	Operating modes can be set independently for two units. • Single scan mode: A/D conversion is performed only once on the analog inputs arbitrarily selected. A/D conversion is performed only once on the temperature sensor output (S12AD1). A/D conversion is performed only once on the internal reference voltage (S12AD1). A/D conversion is performed only once on the extended analog input (S12AD1). • Continuous scan mode: A/D conversion is performed repeatedly on the analog input, temperature sensor output (S12AD1), and internal reference voltage (S12AD1) of the arbitrarily selected channel. A/D conversion is performed repeatedly on the extended analog input (S12AD1). • Group scan mode: Two (groups A and B) or three (groups A, B, and C) can be selected as the number of the groups to be used. Only the combination of groups A and B can be selected when the number of the groups is two. Analog inputs, temperature sensor output (S12AD1), and internal reference voltage (S12AD1) that are arbitrarily selected are divided into two groups (group A and B) or three groups (group A, B, and C), and A/D conversion of the analog input selected on a group basis is performed only once. The conditions for scanning start of groups A, B, and C (synchronous trigger) can be independently selected, thus allowing A/D conversion of each group to be started independently. • Group scan mode (when group priority control selected): If a priority-group trigger is input during scanning of the low-priority group, scan of the low-priority group is stopped and scan of the priority group is started. The priority order is group A (highest) > group B > group C (lowest). Whether or not to restart scanning of the low-priority group after processing for the high-priority group completes, is selectable. Rescan can also be set to start either from the beginning of the selected channel or the channel on which A/D conversion is not completed.

Table 50.1 Specifications of 12-Bit A/D Converter (2/2)

Item	Description
Conditions for A/D conversion start	• Software trigger • Synchronous trigger Trigger by the multi-function timer pulse unit (MTU), 8-bit timer (TMR), 16-bit timer pulse unit (TPU), or event link controller (ELC). • Asynchronous trigger A/D conversion can be triggered by the external trigger ADTRG0# (S12AD) or ADTRG1# (S12AD1) pin (independently for two units).
Functions	• Variable sampling state count (settable for each channel) • Self-diagnosis of 12-bit A/D converter • Selectable A/D-converted value addition mode or average mode • Analog input disconnection detection assist function (discharge function/precharge function) • Double trigger mode (duplication of A/D conversion data) • 12-/10-/8-bit conversion switching • Automatic clear function of A/D data registers • Extended analog input • Comparison function (windows A and B)
Interrupt sources	• In the modes except double trigger mode and group scan mode, a scan end interrupt request (S12ADI or S12ADI1) can be generated on completion of single scan (independently for two units). • In double trigger mode, a scan end interrupt request (S12ADI or S12ADI1) can be generated on completion of double scan (independently for two units). • In group scan mode, a scan end interrupt request (S12ADI or S12ADI1) can be generated on completion of group A scan, whereas a scan end interrupt request (S12GBADI or S12GBADI1) for group B can be generated on completion of group B scan, and a group C scan end interrupt request (S12GCADI or S12GCADI1) can be generated on completion of group C scan. • When double trigger mode is selected in group scan mode, an A/D scan end interrupt request (S12ADI or S12ADI1) can be generated on completion of double scan of group A, and the corresponding scan end interrupt request (S12GBADI/S12GCADI or S12GBADI1/S12GCADI1) can be generated on completion of group B and group C scan. • A compare interrupt request (S12CMPAI, S12CMPAI1, S12CMPBI, or S12CMPBI1) can be generated upon a match with the comparison condition for the digital compare function. • The S12ADI/S12ADI1, S12GBADI/S12GBADI1, and S12GCADI/S12GCADI1 interrupts can activate the DMA controller (DMAC) and data transfer controller (DTC).
Event link	• An ELC event is generated upon completion of all scans • Able to start scanning by a trigger from the ELC
Low power consumption function	• Module stop state can be set.*3, *4

Note 1. The frequency of PCLK, the peripheral module clock, becomes that set in the SCKCR.PCKB[3:0] bits, and that of ADCLK, the A/D conversion clock, becomes the frequency set in the SCKCR.PCKC[3:0] bits for unit 0 (S12AD), and that set in the SCKCR.PCKD[3:0] bits for unit 1 (S12AD1).
Note 2. The number of extended bits during addition differs depending on the addition count.
2-bit extension: 1-time to 4-time conversion (add zero to three times)
4-bit extension: 16-time conversion (add 15 times)
Note 3. See section 11, Low Power Consumption for details.
Note 4. Wait for 1 µs or longer to start A/D conversion after release from the module stop state.

Table 50.2 Functions of 12-Bit A/D Converter

Item			Pin Name, Abbreviation	
			Unit 0 (S12AD)	Unit 1 (S12AD1)
Analog input channels			AN000 to AN007	AN100 to AN111, internal reference voltage, temperature sensor output, extended input
Conditions for A/D conversion start	Software	Software trigger	Enabled	
	Asynchronous trigger	Trigger input pin	ADTRG0#	ADTRG1#
	Synchronous trigger	Compare match/input capture from MTU0.TGRA	TRGA0N	
		Compare match/input capture from MTU1.TGRA	TRGA1N	
		Compare match/input capture from MTU2.TGRA	TRGA2N	
		Compare match/input capture from MTU3.TGRA	TRGA3N	
		Compare match/input capture from MTU4.TGRA or underflow (trough) of MTU4.TCNT in complementary PWM mode	TRGA4N	
		Compare match/input capture from MTU6.TGRA	TRGA6N	
		Compare match/input capture from MTU7.TGRA or underflow (trough) of MTU7.TCNT in complementary PWM mode	TRGA7N	
		Compare match from MTU0.TGRE	TRG0N	
		Compare match between MTU4.TADCORA and MTU4.TCNT	TRG4AN	
		Compare match between MTU4.TADCORB and MTU4.TCNT	TRG4BN	
		Compare match between MTU4.TADCORA and MTU4.TCNT, or compare match between MTU4.TADCORB and MTU4.TCNT	TRG4AN or TRG4BN	
		Compare match between MTU4.TADCORA and MTU4.TCNT, and compare match between MTU4.TADCORB and MTU4.TCNT (when interrupt skipping function 2 is used)	TRG4ABN	
		Compare match between MTU7.TADCORA and MTU7.TCNT	TRG7AN	
		Compare match between MTU7.TADCORB and MTU7.TCNT	TRG7BN	
		Compare match between MTU7.TADCORA and MTU7.TCNT, or compare match between MTU7.TADCORB and MTU7.TCNT	TRG7AN or TRG7BN	
		Compare match between MTU7.TADCORA and MTU7.TCNT, and compare match between MTU7.TADCORB and MTU7.TCNT (when interrupt skipping function 2 is used)	TRG7ABN	
		Compare match between TMR0.TCORA and TMR0.TCNT	TMTRG0AN_0	
		Compare match between TMR2.TCORA and TMR2.TCNT	TMTRG0AN_1	
		Compare match/input capture from TPU0.TGRA or compare match/input capture from TPU1.TGRA or compare match/input capture from TPU2.TGRA or compare match/input capture from TPU3.TGRA or compare match/input capture from TPU4.TGRA	TPTRGAN	
		Compare match/input capture from TPU0.TGRA0	TPTRG0AN	
		ELC trigger	ELCTRG0N	ELCTRG1N
Interrupts			S12ADI, S12GBADI, S12GCADI, S12CMPAI, S12CMPBI interrupt	S12ADI1, S12GBADI1, S12GCADI1, S12CMPAI1, S12CMPBI1 interrupt
Setting of module stop function*1, *2			MSTPCRA.MSTPA17 bit	MSTPCRA.MSTPA16 bit

Note: When setting an A/D conversion start trigger to ADTRG0# or ADTRG1#, set the pin mode control bit in the port mode register for the corresponding pin to 1 (peripheral functions), and set the pin function select bit in the pin function control register to ADTRG0# or ADTRG1#. See section 22, I/O Ports for details.

Note 1. See section 11, Low Power Consumption for details.

Note 2. Wait for 1 µs or longer to start A/D conversion after release from the module stop state.

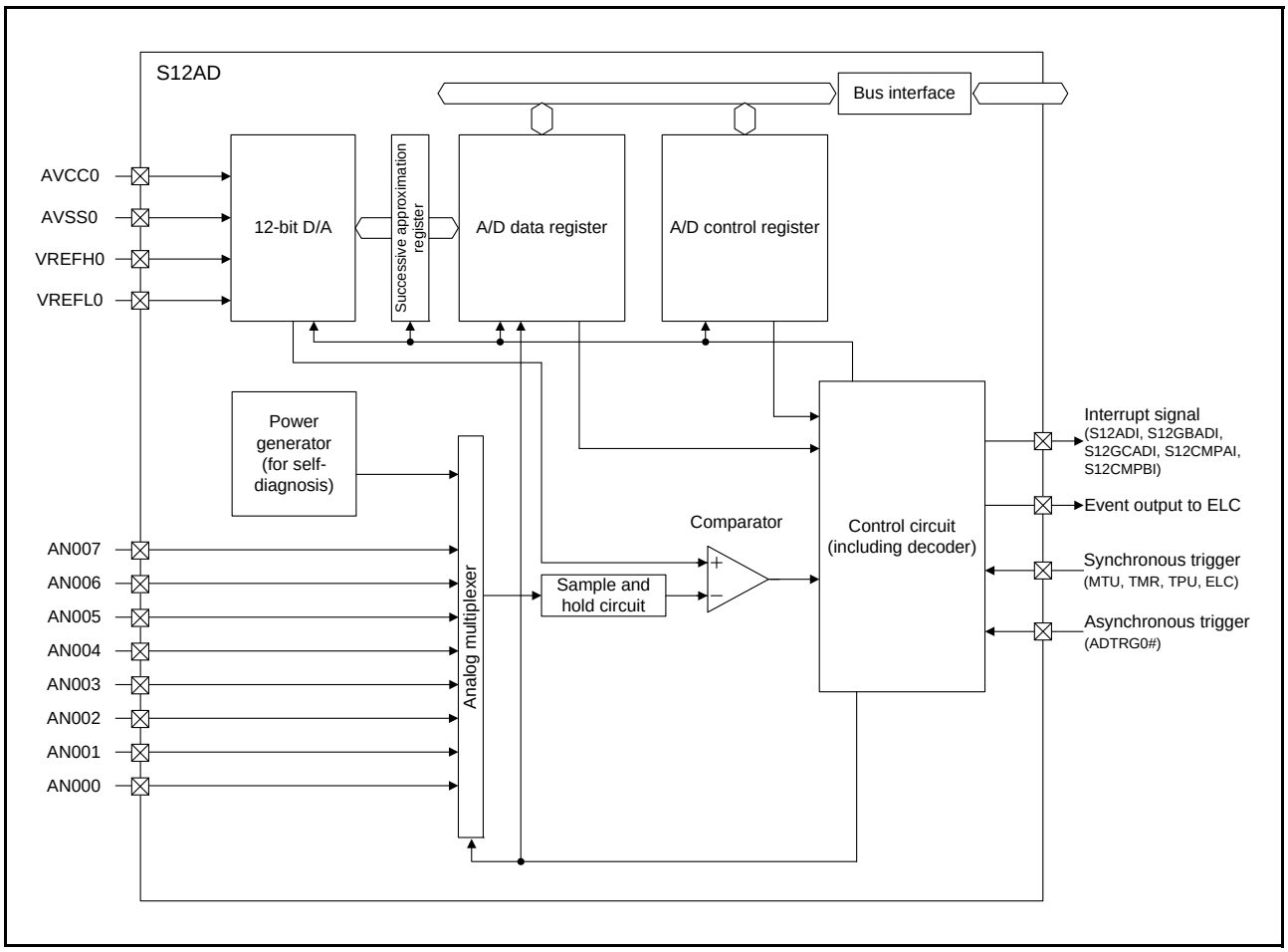


Figure 50.1 Block Diagram of 12-Bit A/D Converter (Unit 0)

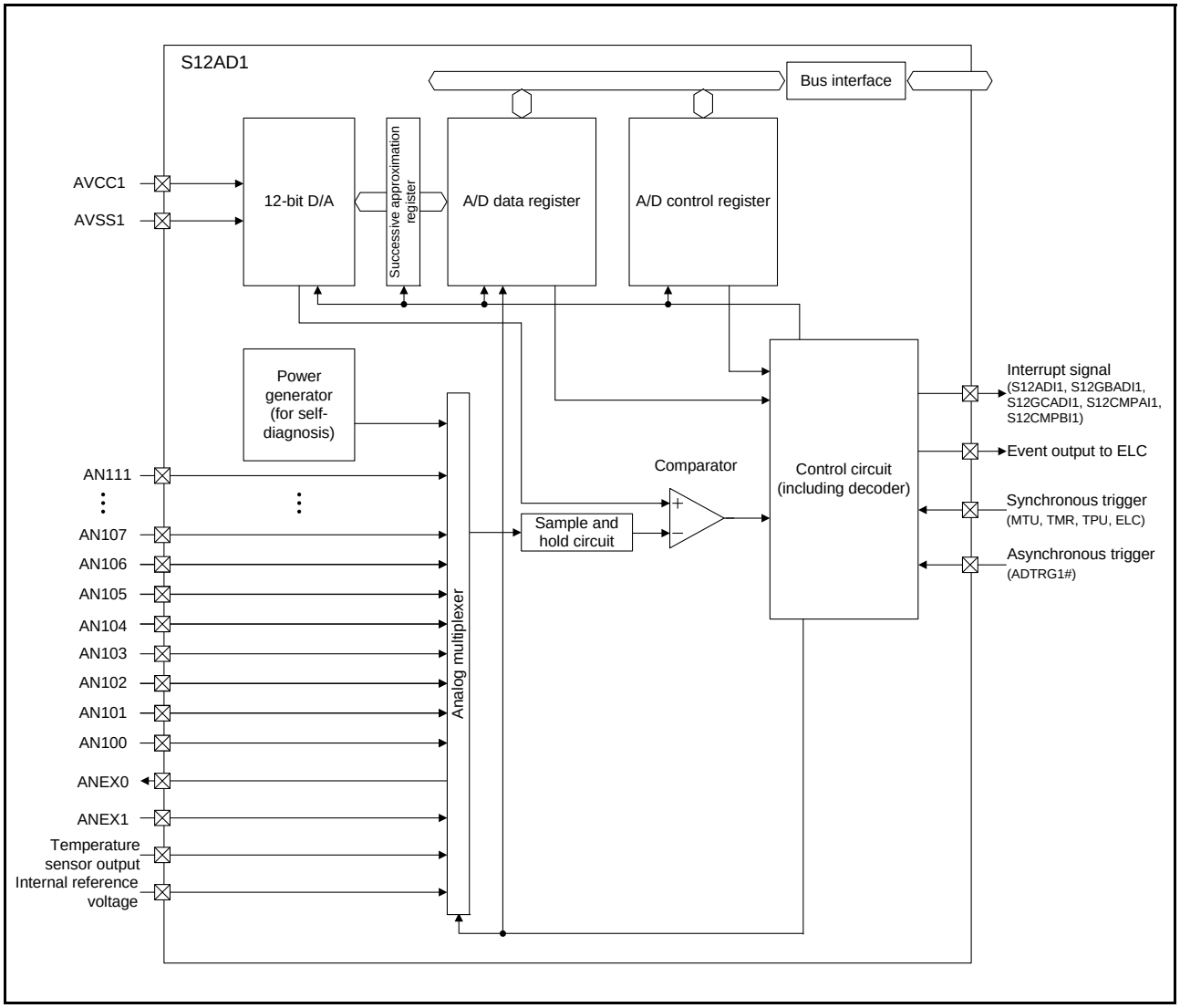


Figure 50.2 Block Diagram of 12-Bit A/D Converter (Unit 1)

Table 50.3 lists the input/output pins of the 12-bit A/D converter.

The 12-bit A/D converter consists of two units, unit 0 (S12AD) and unit 1 (S12AD1). These units can be operated independently. The input channels of S12AD and S12AD1 can be divided into three groups for operation.

Table 50.3 Input/Output Pins of 12-Bit A/D Converter

Unit	Pin Name	I/O	Function
Unit 0 (S12AD)	AVCC0	—	Analog block power supply pin
	AVSS0	—	Analog block ground pin
	VREFH0	Input	Reference power supply pin
	VREFL0	Input	Reference power supply ground pin
	AN000 to AN007	Input	Analog input pins
	ADTRG0#	Input	External trigger input pin for starting A/D conversion
Unit 1 (S12AD1)	AVCC1	—	Multiplexed analog block power supply and reference power supply pin functions
	AVSS1	—	Multiplexed analog block ground and reference power supply ground pin functions
	AN100 to AN111	Input	Analog input pins
	ANEX0	Output	Extended analog output pin
	ANEX1	Input	Extended analog input pin
	ADTRG1#	Input	External trigger input pin for starting A/D conversion

50.2 Register Descriptions

50.2.1 A/D Data Registers y (ADDRy) (y = 0 to 11), A/D Data Duplication Register (ADDBLDR), A/D Data Duplication Register A (ADDBLDRA), A/D Data Duplication Register B (ADDBLDRB), A/D Temperature Sensor Data Register (ADTSDR), A/D Internal Reference Voltage Data Register (ADOCDR)

Address(es): S12AD.ADDR0 0008 9020h, S12AD.ADDR1 0008 9022h, S12AD.ADDR2 0008 9024h, S12AD.ADDR3 0008 9026h, S12AD.ADDR4 0008 9028h, S12AD.ADDR5 0008 902Ah, S12AD.ADDR6 0008 902Ch, S12AD.ADDR7 0008 902Eh, S12AD.ADDBLDR 0008 9018h, S12AD.ADDBLDRA 0008 9084h, S12AD.ADDBLDRB 0008 9086h, S12AD1.ADDR0 0008 9120h, S12AD1.ADDR1 0008 9122h, S12AD1.ADDR2 0008 9124h, S12AD1.ADDR3 0008 9126h, S12AD1.ADDR4 0008 9128h, S12AD1.ADDR5 0008 912Ah, S12AD1.ADDR6 0008 912Ch, S12AD1.ADDR7 0008 912Eh, S12AD1.ADDR8 0008 9130h, S12AD1.ADDR9 0008 9132h, S12AD1.ADDR10 0008 9134h, S12AD1.ADDR11 0008 9136h, S12AD1.ADDBLDR 0008 9118h, S12AD1.ADDBLDRA 0008 9184h, S12AD1.ADDBLDRB 0008 9186h, S12AD1.ADTSDR 0008 911Ah, S12AD1.ADOCDR 0008 911Ch

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

The ADDRy registers (y = 0 to 7 for S12AD; y = 0 to 11 for S12AD1) are 16-bit read-only registers which store the A/D conversion results.
The ADDBLDR register is a 16-bit read-only register used in double trigger mode. The ADDBLDR register stores the results of A/D conversion when the conversion is started by the second trigger.
The ADDBLDRA and ADDBLDRB registers are 16-bit read-only registers that store the A/D conversion results in response to the respective triggers during extended operation in double trigger mode.
The ADTSDR register is a 16-bit read-only register that stores the A/D converted value of the temperature sensor output.
The ADOCDR register is a 16-bit read-only register that stores the A/D conversion results of the internal reference voltage.
The format of each register differs depending on the conditions below.

- Settings of the A/D data register format select bit (ADCER.ADRFMT) (flush-right or flush-left)
- Settings of the A/D data register bit resolution setting bits (ADCER.ADPRC[1:0]) (12-, 10-, or 8-bit)
- Settings of the addition count select bits (ADADC.ADC[2:0]) (2-, 3-, 4-, or 16-time conversion)
- Settings of the average mode enable bit (ADADC.AVEE) (add or average)

The data formats for each given condition are shown below.

(1) When A/D-Converted Value Addition/Average Mode is Not Selected

- Flush-right format with setting of 12-bit resolution
  The A/D-converted value is stored in bits 11 to 0. Bits 15 to 12 are read as 0.
- Flush-right format with setting of 10-bit resolution
  The A/D-converted value is stored in bits 9 to 0. Bits 15 to 10 are read as 0.
- Flush-right format with setting of 8-bit resolution
  The A/D-converted value is stored in bits 7 to 0. Bits 15 to 8 are read as 0.
- Flush-left format with setting of 12-bit resolution
  The A/D-converted value is stored in bits 15 to 4. Bits 3 to 0 are read as 0.
- Flush-left format with setting of 10-bit resolution
  The A/D-converted value is stored in bits 15 to 6. Bits 5 to 0 are read as 0.
- Flush-left format with setting of 8-bit resolution

The A/D-converted value is stored in bits 15 to 8. Bits 7 to 0 are read as 0.

(2) When A/D-Converted Average Mode is Selected

- Flush-right format with setting of 12-bit resolution
  The mean value of the A/D-converted results of the same channel is stored in bits 11 to 0.
  Bits 15 to 12 are read as 0.
- Flush-right format with setting of 10-bit resolution
  The mean value of the A/D-converted results of the same channel is stored in bits 9 to 0.
  Bits 15 to 10 are read as 0.
- Flush-right format with setting of 8-bit resolution
  The mean value of the A/D-converted results of the same channel is stored in bits 7 to 0.
  Bits 15 to 8 are read as 0.
- Flush-left format with setting of 12-bit resolution
  The mean value of the A/D-converted results of the same channel is stored in bits 15 to 4.
  Bits 3 to 0 are read as 0.
- Flush-left format with setting of 10-bit resolution
  The mean value of the A/D-converted results of the same channel is stored in bits 15 to 6.
  Bits 5 to 0 are read as 0.
- Flush-left format with setting of 8-bit resolution
  The mean value of the A/D-converted results of the same channel is stored in bits 15 to 8.
  Bits 7 to 0 are read as 0.

A/D-converted value average mode can be set only when 2- or 4-time conversion is selected in A/D-converted value addition mode.

(3) When A/D-Converted Value Addition Mode is Selected

- Flush-right format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 12-bit resolution
  The value added by the A/D-converted value of the same channel is stored in bits 13 to 0.
  Bits 15 and 14 are read as 0.
- Flush-right format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 10-bit resolution
  The value added by the A/D-converted value of the same channel is stored in bits 11 to 0.
  Bits 15 to 12 are read as 0.
- Flush-right format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 8-bit resolution
  The value added by the A/D-converted value of the same channel is stored in bits 9 to 0.
  Bits 15 to 10 are read as 0.
- Flush-right format (in A/D-converted value addition mode and when number of conversions is selected for 16 times) with setting of 12-bit resolution
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.
  When the 16-time conversion is selected, setting with 10- or 8-bit resolution is prohibited.
- Flush-left format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 12-bit resolution
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 2.
  Bits 1 and 0 are read as 0.
- Flush-left format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 10-bit resolution
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 4.
  Bits 3 to 0 are read as 0.
- Flush-left format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4

times) with setting of 8-bit resolution
The value added by the A/D-converted value of the same channel is stored in bits 15 to 6.
Bits 5 to 0 are read as 0.

- Flush-left format (in A/D-converted value addition mode and when number of conversions is selected for 16 times) with setting of 12-bit resolution
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.
  When the 16-time conversion is selected, setting with 10- or 8-bit resolution is prohibited.

When A/D-converted addition mode is selected, the value added by the A/D-converted value of the same channel is indicated. The number of A/D conversions can be set to 1, 2, 3, 4, or 16 times. If A/D-converted addition mode is selected, when the conversion count is set to 1 to 4 times, the value added by the A/D conversion result is retained in the A/D data register as 2-bit extended data of the conversion accuracy bits; when the conversion count is set to 16 times, the value added by the A/D conversion result is retained in the A/D data register as 4-bit extended data of the conversion accuracy bits. Even if A/D-converted value addition mode is selected, the value is stored in the A/D data register according to the settings of the A/D data register format select bits. When the 16-time conversion is selected, setting with 10- or 8-bit resolution is prohibited.

50.2.2 A/D Self-Diagnosis Data Register (ADRD)

Address(es): S12AD.ADRD 0008 901Eh, S12AD1.ADRD 0008 911Eh

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

ADRD is a 16-bit read-only register that stores the A/D conversion results based on the 12-bit A/D converter's self-diagnosis. In addition to the A/D-converted value, the self-diagnosis status is included in. In the ADRD register, the different formats are used depending on the conditions below.

- Settings of the A/D data register format select bit (ADCER.ADRFMT) (flush-right or flush-left)
- Settings of the A/D data register bit resolution setting bits (ADCER.ADPRC[1:0]) (12-, 10-, or 8-bit)

The A/D-converted value addition mode and A/D-converted value average mode cannot be applied to the A/D self-diagnosis function. For details of self-diagnosis, see section 50.2.9, A/D Control Extended Register (ADCER).

The data formats for each given condition are shown below.

- Flush-right format with setting of 12-bit resolution
  The A/D-converted value is stored in bits 11 to 0. The self-diagnosis status is stored in bits 15 and 14. Bits 13 and 12 are read as 0.
- Flush-right format with setting of 10-bit resolution
  The A/D-converted value is stored in bits 9 to 0. The self-diagnosis status is stored in bits 15 and 14. Bits 13 to 10 are read as 0.
- Flush-right format with setting of 8-bit resolution
  The A/D-converted value is stored in bits 7 to 0. The self-diagnosis status is stored in bits 15 and 14. Bits 13 to 8 are read as 0.
- Flush-left format with setting of 12-bit resolution
  The A/D-converted value is stored in bits 15 to 4. The self-diagnosis status is stored in bits 1 and 0. Bits 3 and 2 are read as 0.
- Flush-left format with setting of 10-bit resolution
  The A/D-converted value is stored in bits 15 to 6. The self-diagnosis status is stored in bits 1 and 0. Bits 5 to 2 are read as 0.
- Flush-left format with setting of 8-bit resolution
  The A/D-converted value is stored in bits 15 to 8. The self-diagnosis status is stored in bits 1 and 0. Bits 7 to 2 are read as 0.

Table 50.4 Self-Diagnosis Status Description

Bits 15 and 14 for flush-right format setting Bits 1 and 0 for flush-left format setting	Self-diagnosis status
00b	Self-diagnosis has never been executed since power-on.
01b	Self-diagnosis using the voltage of 0 V has been executed.
10b	Self-diagnosis using the voltage of reference power supply × 1/2 has been executed.
11b	Self-diagnosis using the voltage of reference power supply has been executed.

Note: For details of self-diagnosis, see section 50.2.9, A/D Control Extended Register (ADCER).

50.2.3 A/D Control Register (ADCSR)

Address(es): S12AD.ADCSR 0008 9000h, S12AD1.ADCSR 0008 9100h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADST	ADCS[1:0]		ADIE	—	—	TRGE	EXTRG	DBLE	GBADIE	—	DBLANS[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	DBLANS[4:0]	Double Trigger Channel Select	These bits select one analog input channel for double triggered operation. The setting is only effective while double trigger mode is selected.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	GBADIE	Group B Scan End Interrupt Enable	0: Disables interrupt generation upon group B scan completion. 1: Enables interrupt generation upon group B scan completion.	R/W
b7	DBLE	Double Trigger Mode Select	0: Deselects double trigger mode. 1: Selects double trigger mode.	R/W
b8	EXTRG	Trigger Select*1	0: A/D conversion is started by synchronous trigger. 1: A/D conversion is started by asynchronous trigger.	R/W
b9	TRGE	Trigger Start Enable	0: Disables A/D conversion to be started by synchronous or asynchronous trigger. 1: Enables A/D conversion to be started by synchronous or asynchronous trigger.	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	ADIE	Scan End Interrupt Enable	0: Disables interrupt generation upon scan completion. 1: Enables interrupt generation upon scan completion.	R/W
b14, b13	ADCS[1:0]	Scan Mode Select	b14 b13 0 0: Single scan mode 0 1: Group scan mode 1 0: Continuous scan mode 1 1: Setting prohibited	R/W
b15	ADST	A/D Conversion Start	0: Stops A/D conversion process. 1: Starts A/D conversion process.	R/W

Note 1. Starting A/D conversion using an external pin (asynchronous trigger)
Set the ADCSR.TRGE and EXTRG bits to 1 while a high-level signal is input to the external pin (ADTRG0# or ADTRG1#). Then, if the ADTRG0# or ADTRG1# signal is changed to low, the falling edge is detected and the scan process is started. In this case, the pulse width of the low-level input must be at least 1.5 clock cycles of PCLK.
The relationship between each unit and the external pin (asynchronous trigger) is shown below.

Unit	External pin (asynchronous trigger)
S12AD	ADTRG0#
S12AD1	ADTRG1#

ADCSR sets double trigger mode, A/D conversion start trigger; enables/disables scan end interrupt; selects the scan mode; and starts or stops A/D conversion.

DBLANS[4:0] Bits (Double Trigger Channel Select)

The DBLANS[4:0] bits select one of the channels for A/D conversion data duplication in double trigger mode. The A/D conversion results of the analog input of the channel selected by the DBLANS[4:0] bits are stored into the A/D data register y when conversion is started by the first trigger, and into the A/D data duplication register when started by the second trigger. Table 50.5 shows selection of the channel for double triggered operation.
When double trigger mode is selected, the channels selected by the ADANSA0 register are invalid, and the channel selected by the DBLANS[4:0] bits is subjected to A/D conversion instead.
When double trigger mode is selected in group scan mode, double trigger mode operation is performed for group A only

and not performed for group B or C. Also, in double trigger mode, the analog inputs of multiple channels, temperature sensor outputs, or internal reference voltage cannot be selected for group A, but can be selected for groups B and C.
The DBLANS[4:0] bits should be set while the ADST bit is 0. They should not be set simultaneously when 1 is written to the ADST bit.

Table 50.5 Relationship between DBLANS[4:0] Bits Settings and Double Trigger Enabled Channels

S12AD (Unit 0)		S12AD1 (Unit 1)			
DBLANS[4:0]	Duplication Channel	DBLANS[4:0]	Duplication Channel	DBLANS[4:0]	Duplication Channel
00000b	AN000	00000b	AN100	01000b	AN108
00001b	AN001	00001b	AN101	01001b	AN109
00010b	AN002	00010b	AN102	01010b	AN110
00011b	AN003	00011b	AN103	01011b	AN111
00100b	AN004	00100b	AN104		
00101b	AN005	00101b	AN105		
00110b	AN006	00110b	AN106		
00111b	AN007	00111b	AN107		

Note: Duplication cannot be selected for the A/D conversion data of self-diagnosis, temperature sensor output, and internal reference voltage.

GBADIE Bit (Group B Scan End Interrupt Enable)

The GBADIE bit enables or disables group B scan end interrupt in group scan mode.
A scan end interrupt for group B is provided individually for each unit. Table 50.6 shows the relationship between each unit and the scan end interrupt for group B.

Table 50.6 Relationship between Each Unit and Group B Scan End Interrupt

Unit	Group B Scan End Interrupt
S12AD	S12GBADI
S12AD1	S12GBADI1

DBLE Bit (Double Trigger Mode Select)

Double trigger mode has a function to store the resulting data of A/D conversion started by the first and second synchronous triggers into separate registers.
When double trigger mode is selected, the channels specified in the ADANSA0 register are invalid and the channel selected by the DBLANS[4:0] bits is effective instead. Double trigger mode can be only operated by the synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits. Do not generate an asynchronous or software trigger. The A/D conversion results started by the first trigger are stored into the A/D data register y and those started by the second trigger are stored into the A/D data duplication register. In this case, if the ADIE bit is set to 1, the interrupt is generated not upon completion of the first conversion but upon completion of the second conversion.
In continuous scan mode, double trigger mode should not be selected. In addition, double trigger mode should not be used for self-diagnosis, or conversion of the temperature sensor output and internal reference voltage. When using double trigger mode in group scan mode, A/D conversion of the internal reference voltage should not be selected for group A.
The DBLE bit should be set after the ADST bit has been set to 0.

EXTRG Bit (Trigger Select)

The EXTRG bit selects the synchronous trigger or the asynchronous trigger as the trigger for starting A/D conversion. In group scan mode, the setting of this bit is valid for the selected trigger of group A.
For groups B and C, A/D conversion is started by the selected synchronous trigger regardless of this bit setting.

TRGE Bit (Trigger Start Enable)

The TRGE bit enables or disables A/D conversion by the synchronous trigger and the asynchronous trigger.
This bit should be set to 1 in group scan mode.

ADIE Bit (Scan End Interrupt Enable)

The ADIE bit enables or disables the A/D scan end interrupt (S12ADI) in scans except for groups B and C scan in group scan mode.
With double trigger mode deselected, the A/D scan conversion end is generated after the first scan is completed if the ADIE bit is set to 1.
When the extended analog input is selected also, if the ADIE bit is set to 1 when A/D conversion is completed, an A/D scan conversion end interrupt is generated.
With double trigger mode selected, the A/D scan conversion end is generated after the second scan is completed if the ADIE bit is set to 1 as long as the scan is started by the synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits.
When scan is started by a software trigger, even with double trigger mode selected, the A/D scan conversion end interrupt is generated if the ADIE bit is set to 1 when the scan is completed. The A/D scan conversion end interrupt is provided individually for each unit. Table 50.7 shows the relationship between each unit and the A/D scan conversion end interrupt.

Table 50.7 Relationship between Each Unit and A/D scan conversion end Interrupt

Unit	A/D scan conversion end interrupt
S12AD	S12ADI
S12AD1	S12ADI1

ADCS[1:0] Bits (Scan Mode Select)

The ADCS[1:0] bits select the scan mode.
In single scan mode, A/D conversion is performed for the analog inputs selected with the ADANSA0 register in the ascending order of the channel number, and when one cycle of A/D conversion is completed for all the selected channels, the scan conversion is stopped.*1
In continuous scan mode, while the ADCSR.ADST bit is 1, A/D conversion is performed for the analog inputs selected with the ADANSA0 register in the ascending order of the channel number, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is repeated from the first channel. If the ADCSR.ADST bit is set to 0 during continuous scan, A/D conversion is stopped even if scanning is in progress.*1
In group scan mode, A/D conversion is performed for the analog inputs (group A) selected with the ADANSA0 register in the ascending order of the channel number after scanning is started by the synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is stopped.*1 A/D conversion is also performed for the analog inputs (group B or C) selected with the ADANSB0 register and the ADANSC0 register in the ascending order of the channel number after A/D conversion is started by the synchronous trigger selected by the ADSTRGR.TRSB[5:0] bits and ADGCTRGR.TRSC[5:0] bits, and when A/D conversion is completed for all the selected channels, A/D conversion is stopped.*1
When selecting group scan mode, different channels and triggers should be selected for groups A, B, and C.
When using two groups while group scan mode is set, use groups A and B (ADGCTRGR.GRCE bit = 0). When using three groups, use groups A, B, and C (ADGCTRGR.GRCE bit = 1).
When selecting the extended analog input, select single scan mode or sequence scan mode.
The ADCS[1:0] bits should be set while the ADST bit is 0. They should not be set simultaneously when 1 is written to the ADST bit.

Note 1. When the temperature sensor output or internal reference voltage is selected, after A/D conversion of the analog input, A/D conversion is performed on the temperature sensor output and internal reference voltage in this order.

Table 50.8 Selection of Scan Mode, Double Trigger Mode, and Targets for A/D Conversion

Scan Mode Setup	Double Trigger Mode Setting	Targets for A/D Conversion					
		Self-diagnosis	Analogue Input (Including Group A)	Analogue Input (Group B and Group C)	Temperature Sensor Output	Internal Reference Voltage	Extended Analog Input
Single scan	DBLE = 0	✓	✓	×	✓	✓	✓
	DBLE = 1	×	✓ (1 channel only)	×	×	×	×
Sequence scan	DBLE = 0	✓	✓	×	✓	✓	✓
	DBLE = 1	×	×	×	×	×	×
Group scan	DBLE = 0	✓	✓	✓	✓	✓	×
	DBLE = 1	×	✓ (1 channel only)	✓	✓ (Group B and Group C)	✓ (Group B and Group C)	×

✓: Selectable, ×: Not selectable

Note: When selecting the extended analog input, deselect other targets for A/D conversion.

ADST Bit (A/D Conversion Start)

The ADST bit starts or stops A/D conversion process.

Before the ADST bit is set to 1, set the A/D conversion clock, the conversion mode, and conversion target analog input.

[Setting conditions]

- 1 is written by software.
- The synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits is detected with ADCSR.EXTRG and ADCSR.TRGE bits being set to 0 and 1, respectively.
- The synchronous trigger selected by the ADSTRGR.TRSB[5:0] bits is detected with the ADCSR.TRGE bit being set to 1 in group scan mode.
- The asynchronous trigger is detected with the ADCSR.TRGE and ADCSR.EXTRG bits being set to 1 and the ADSTRGR.TRSA[5:0] bits being set to 000000b.
- With group priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), a group B or C trigger is detected and A/D conversion of group B or C is started.
- With group priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRSCN bit is set to 1 and A/D conversion of group B or C is restarted.
- With group priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRP bit is set to 1 and A/D conversion of the lowest-priority group is started.

[Clearing conditions]

- 0 is written by software.
- The A/D conversion of all the selected channels, temperature sensor output, or the internal reference voltage (for S12AD1 only) is completed in single scan mode.
- A/D conversion of extended analog inputs is completed in single scan mode.
- Group A scan is completed in group scan mode.
- Group B scan is completed in group scan mode.
- Group C scan is completed in group scan mode.
- With group priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRSCN bit is set to 1 and the scanning of the low-priority group started by a trigger is stopped.

Note: When group priority control operation mode has been enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), do not set the ADST bit to 1.

Note: When group priority control operation mode has been enabled (ADCSR.ADCS[1:0] bits = 01b and

ADGSPCR.PGS bit = 1) and ADGSPCR.GBRP = 1, do not set the ADST bit to 0. When forcibly terminating A/D conversion, follow the procedure for clearing the ADST bit.

Note: When the single-scan continuous function is used (ADGSPCR.GBRP bit = 1) with group priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADST bit remains as 1.

50.2.4 A/D Channel Select Register A0 (ADANSA0)

(1) S12AD.ADANSA0

Address(es): 0008 9004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ANSA007	ANSA006	ANSA005	ANSA004	ANSA003	ANSA002	ANSA001	ANSA000
Value after 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	ANSA000	A/D Conversion Channel Select	0: AN000 to AN007 are not subjected to conversion. 1: AN000 to AN007 are subjected to conversion.	R/W
b1	ANSA001			R/W
b2	ANSA002			R/W
b3	ANSA003			R/W
b4	ANSA004			R/W
b5	ANSA005			R/W
b6	ANSA006			R/W
b7	ANSA007			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

S12AD.ADANSA0 selects analog input channels for A/D conversion from among AN000 to AN007. In group scan mode, this register selects group A channels.

ANSA0n Bit (n = 00 to 07) (A/D Conversion Channel Select)

The ANSA0n bit select analog input channels for A/D conversion from among AN000 to AN007. The channels to be selected and the number of channels can be arbitrarily set. The ANSA000 bit corresponds to AN000 and the ANSA007 bit corresponds to AN007.

When double trigger mode is selected, the channel selected by S12AD.ADCSR.DBLANS[4:0] bits is selected in group A, and the ANSA0n bit setting is invalid.

The ANSA0n bit should be set while the S12AD.ADCSR.ADST bit is 0.

(2) S12AD1.ADANSA0

Address(es): 0008 9104h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ANSA011	ANSA010	ANSA009	ANSA008	ANSA007	ANSA006	ANSA005	ANSA004	ANSA003	ANSA002	ANSA001	ANSA000
Value after 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	ANSA000	A/D Conversion Channel Select	0: AN100 to AN111 are not subjected to conversion. 1: AN100 to AN111 are subjected to conversion.	R/W
b1	ANSA001			R/W
b2	ANSA002			R/W
b3	ANSA003			R/W
b4	ANSA004			R/W
b5	ANSA005			R/W
b6	ANSA006			R/W
b7	ANSA007			R/W
b8	ANSA008			R/W
b9	ANSA009			R/W
b10	ANSA010			R/W
b11	ANSA011			R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

S12AD1.ADANSA0 selects analog input channels for A/D conversion from among AN100 to AN111. In group scan mode, this register selects group A channels.

ANSA0n Bit (n = 00 to 11) (A/D Conversion Channel Select)

The ANSA0n bit select analog input channels for A/D conversion from among AN100 to AN111. The channels to be selected and the number of channels can be arbitrarily set. The ANSA000 bit corresponds to AN100 and the ANSA011 bit corresponds to AN111.

When double trigger mode is selected, the channel selected by the S12AD1.ADCSR.DBLANS[4:0] bits is selected in group A, and the ANSA0n bit setting is invalid.

The ANSA0n bit should be set while the S12AD1.ADCSR.ADST bit is 0.

50.2.5 A/D Channel Select Register B0 (ADANSB0)

(1) S12AD.ADANSB0

Address(es): 0008 9014h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ANSB007	ANSB006	ANSB005	ANSB004	ANSB003	ANSB002	ANSB001	ANSB000
Value after 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	ANSB000	A/D Conversion Channel Select	0: AN000 to AN007 are not subjected to conversion. 1: AN000 to AN007 are subjected to conversion.	R/W
b1	ANSB001			R/W
b2	ANSB002			R/W
b3	ANSB003			R/W
b4	ANSB004			R/W
b5	ANSB005			R/W
b6	ANSB006			R/W
b7	ANSB007			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

S12AD.ADANSB0 selects analog input channels for A/D conversion from among AN000 to AN007 in group B when group scan mode is selected. The S12AD.ADANSB0 register is not used in any scan mode other than group scan mode.

ANSB0n Bit (n = 00 to 07) (A/D Conversion Channel Select)

The ANSB0n bit select analog input channels for A/D conversion from among AN000 to AN007 in group B when group scan mode is selected. The S12AD.ADANSB0 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the channels corresponding to group A, selected with the S12AD.ADANSA0 register and the S12AD.ADCSR.DBLANS[4:0] bits in double trigger mode) should be excluded as the channels to be selected and the number of channels to be set.

The ANSB000 bit corresponds to AN000 and the ANSB007 bit corresponds to AN007.

The ANSB0n bit should be set while the S12AD.ADCSR.ADST bit is 0.

(2) S12AD1.ADANSB0

Address(es): 0008 9114h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ANSB011	ANSB010	ANSB009	ANSB008	ANSB007	ANSB006	ANSB005	ANSB004	ANSB003	ANSB002	ANSB001	ANSB000
Value after 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	ANSB000	A/D Conversion Channel Select	0: AN100 to AN111 are not subjected to conversion. 1: AN100 to AN111 are subjected to conversion.	R/W
b1	ANSB001			R/W
b2	ANSB002			R/W
b3	ANSB003			R/W
b4	ANSB004			R/W
b5	ANSB005			R/W
b6	ANSB006			R/W
b7	ANSB007			R/W
b8	ANSB008			R/W
b9	ANSB009			R/W
b10	ANSB010			R/W
b11	ANSB011			R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

S12AD1.ADANSB0 selects analog input channels for A/D conversion from among AN100 to AN111 in group B when group scan mode is selected. The S12AD1.ADANSB0 register is not used in any scan mode other than group scan mode.

ANSB0n Bit (n = 00 to 11) (A/D Conversion Channel Select)

The ANSB0n bit select analog input channels for A/D conversion from among AN100 to AN111 in group B when group scan mode is selected. The S12AD1.ADANSB0 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the channels corresponding to group A, selected with the S12AD1.ADANSA0 register and the ADCSR.DBLANS[4:0] bits in double trigger mode) should be excluded as the channels to be selected and the number of channels to be set.

The ANSB000 bit corresponds to AN100 and the ANSB011 bit corresponds to AN111.

The ANSB0n bit should be set while the S12AD1.ADCSR.ADST bit is 0.

50.2.6 A/D Channel Select Register C0 (ADANSC0)

(1) S12AD.ADANSC0

Address(es): 0008 90D4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ANSC007	ANSC006	ANSC005	ANSC004	ANSC003	ANSC002	ANSC001	ANSC000
Value after 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	ANSC000	A/D Conversion Channel Select	0: AN000 to AN007 are not subjected to conversion. 1: AN000 to AN007 are subjected to conversion.	R/W
b1	ANSC001			R/W
b2	ANSC002			R/W
b3	ANSC003			R/W
b4	ANSC004			R/W
b5	ANSC005			R/W
b6	ANSC006			R/W
b7	ANSC007			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

S12AD.ADANSC0 selects analog input channels for A/D conversion from among AN000 to AN007 in group C when group scan mode is selected. The S12AD.ADANSC0 register is not used in any scan mode other than group scan mode.

ANSC0n Bit (n = 00 to 07) (A/D Conversion Channel Select)

The ANSC0n bit select analog input channels for A/D conversion from among AN000 to AN007 in group C when group scan mode is selected. The S12AD.ADANSC0 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the channels corresponding to group A, selected with the S12AD.ADANSA0 register and the S12AD.ADCSR.DBLANS[4:0] bits in double trigger mode) should be excluded as the channels to be selected and the number of channels to be set.

The ANSC000 bit corresponds to AN000 and the ANSC007 bit corresponds to AN007.

The ANSC0n bit should be set while the S12AD.ADCSR.ADST bit is 0.

(2) S12AD1.ADANSC0

Address(es): 0008 91D4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ANSC011	ANSC010	ANSC009	ANSC008	ANSC007	ANSC006	ANSC005	ANSC004	ANSC003	ANSC002	ANSC001	ANSC000
Value after 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	ANSC000	A/D Conversion Channel Select	0: AN100 to AN111 are not subjected to conversion. 1: AN100 to AN111 are subjected to conversion.	R/W
b1	ANSC001			R/W
b2	ANSC002			R/W
b3	ANSC003			R/W
b4	ANSC004			R/W
b5	ANSC005			R/W
b6	ANSC006			R/W
b7	ANSC007			R/W
b8	ANSC008			R/W
b9	ANSC009			R/W
b10	ANSC010			R/W
b11	ANSC011			R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

S12AD1.ADANSC0 selects analog input channels for A/D conversion from among AN100 to AN111 in group C when group scan mode is selected. The S12AD1.ADANSC0 register is not used in any scan mode other than group scan mode.

ANSC0n Bit (n = 00 to 11) (A/D Conversion Channel Select)

The ANSC0n bit select analog input channels for A/D conversion from among AN100 to AN111 in group C when group scan mode is selected. The S12AD1.ADANSC0 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the channels corresponding to group A, selected with the S12AD1.ADANSA0 register and the S12AD1.ADCSR.DBLANS[4:0] bits in double trigger mode) should be excluded as the channels to be selected and the number of channels to be set.

The ANSC000 bit corresponds to AN100 and the ANSC011 bit corresponds to AN111.

The ANSC0n bit should be set while the S12AD1.ADCSR.ADST bit is 0.

50.2.7 A/D-Converted Value Addition/Average Function Channel Select Register 0 (ADADS0)

(1) S12AD.ADADS0

Address(es): 0008 9008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ADS007	ADS006	ADS005	ADS004	ADS003	ADS002	ADS001	ADS000
Value after 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	ADS000	A/D-Converted Value Addition/ Average Channel Select	0: A/D-converted value addition/average mode for AN000 to AN007 is not selected. 1: A/D-converted value addition/average mode for AN000 to AN007 is selected.	R/W
b1	ADS001			R/W
b2	ADS002			R/W
b3	ADS003			R/W
b4	ADS004			R/W
b5	ADS005			R/W
b6	ADS006			R/W
b7	ADS007			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

S12AD.ADADS0 selects channels AN000 to AN007 on which A/D conversion is performed successively 2, 3, 4, or 16 times and then converted values are added (integrated) or averaged.

ADS0n Bit (n = 00 to 07) (A/D-Converted Value Addition/Average Channel Select)

When the ADS0n bit of the number that is the same as that of A/D-converted channel selected by the S12AD.ADANSA0.ANSA0n bit (n = 00 to 07) or S12AD.ADCSR.DBLANS[4:0] bits and S12AD.ADANSB0.ANSB0n bit is set to 1, A/D conversion of analog input of the selected channels is performed successively 2, 3, 4, or 16 times that is set with the S12AD.ADADC.ADC[2:0] bits.

When the S12AD.ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D data register.

When the S12AD.ADADC.AVEE bit is 1, the mean value of the results obtained by addition (integration) is stored in the A/D data register.

As for the channel on which the A/D conversion is performed and addition/average mode is not selected, a normal one-time conversion is executed and the conversion result is stored to the A/D data register.

The ADS0n bit should be set while the S12AD.ADCSR.ADST bit is 0.

(2) S12AD1.ADADS0

Address(es): 0008 9108h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	ADS011	ADS010	ADS009	ADS008	ADS007	ADS006	ADS005	ADS004	ADS003	ADS002	ADS001	ADS000
Value after 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	ADS000	A/D-Converted Value Addition/ Average Channel Select	0: A/D-converted value addition/average mode for AN100 to AN111 is not selected. 1: A/D-converted value addition/average mode for AN100 to AN111 is selected.	R/W
b1	ADS001			R/W
b2	ADS002			R/W
b3	ADS003			R/W
b4	ADS004			R/W
b5	ADS005			R/W
b6	ADS006			R/W
b7	ADS007			R/W
b8	ADS008			R/W
b9	ADS009			R/W
b10	ADS010			R/W
b11	ADS011			R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

S12AD1.ADADS0 selects channels AN100 to AN111 on which A/D conversion is performed successively 2, 3, 4, or 16 times and then converted values are added (integrated) or averaged.

ADS0n Bit (n = 00 to 11) (A/D-Converted Value Addition/Average Channel Select)

When the ADS0n bit of the number that is the same as that of A/D-converted channel selected by the S12AD1.ADANSA0.ANSA0n bit (n = 00 to 11) or S12AD1.ADCSR.DBLANS[4:0] bits and S12AD1.ADANSB0.ANSB0n bit and S12AD1.ADANSC0.ANSC0n bit is set to 1, A/D conversion of analog input of the selected channels is performed successively 2, 3, 4, or 16 times that is set with the S12AD1.ADADC.ADC[2:0] bits.
When the S12AD1.ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D data register.
When the S12AD1.ADADC.AVEE bit is 1, the mean value of the results obtained by addition (integration) is stored in the A/D data register.
As for the channel on which the A/D conversion is performed and addition/average mode is not selected, a normal one-time conversion is executed and the conversion result is stored to the A/D data register.
The ADS0n bit should be set while the S12AD1.ADCSR.ADST bit is 0.

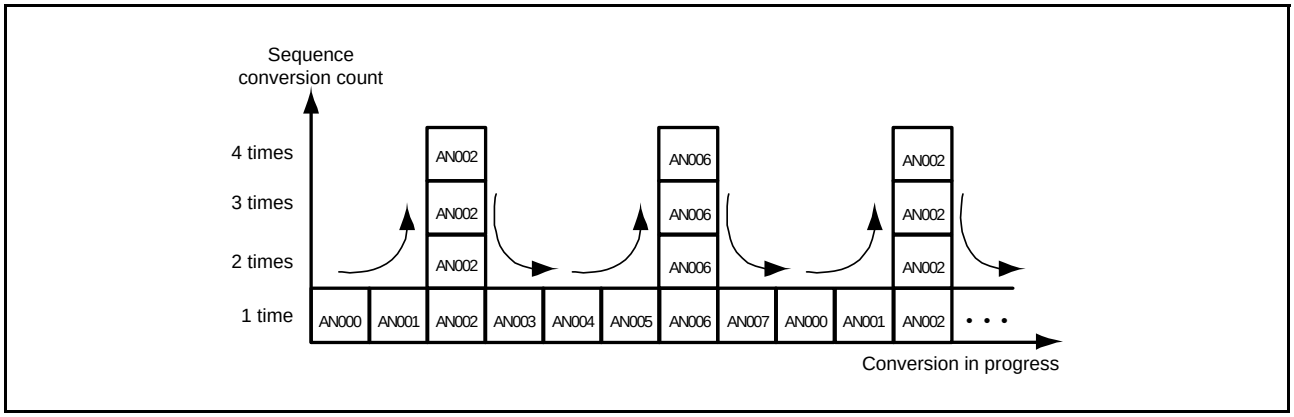


Figure 50.3 Scan Conversion Sequence with S12AD.ADADC.ADC[2:0] = 011b, S12AD.ADADC.AVEE = 0, S12AD.ADADS0.ADS002 = 1, and S12AD.ADADS0.ADS006 = 1

50.2.8 A/D-Converted Value Addition/Average Count Select Register (ADADC)

Address(es): S12AD.ADADC 0008 900Ch, S12AD1.ADADC 0008 910Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	AVEE	—	—	—	—	ADC[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	ADC[2:0]	Addition Count Select	b2 b0 0 0 0: 1-time conversion (no addition; same as normal conversion) 0 0 1: 2-time conversion (addition once) 0 1 0: 3-time conversion (addition twice)*1 0 1 1: 4-time conversion (addition three times) 1 0 1: 16-time conversion (addition 15 times)*1 Settings other than above are prohibited.	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	AVEE	Average Mode Enable	0: Addition mode is selected. 1: Average mode is selected.	R/W

Note 1. The AVEE bit is enabled only when 2-time or 4-time conversion is selected. When average mode is selected (ADADC.AVEE bit = 1), do not set 3-time conversion (ADADC.ADC[2:0] = 010b) nor 16-time conversion (ADADC.ADC[2:0] = 101b).

ADADC sets the addition count for A/D conversion of the channel, temperature sensor output, and internal reference voltage for which A/D-converted value addition/average mode is selected, and selects either addition or average mode.

ADC[2:0] Bits (Addition Count Select)

The ADC[2:0] bits set the addition count common to the channels for which A/D conversion and A/D-converted value addition/average mode is selected, including the channels selected in double trigger mode (by ADCSR.DBLANS[4:0] bits), and to A/D conversion of temperature sensor output or internal reference voltage.
When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the addition count to one time (ADADC.ADC[2:0] = 000b), three times (ADADC.ADC[2:0] = 010b), or 16 times (ADADC.ADC[2:0] = 101b).
The ADC[2:0] bits should be set while the ADCSR.ADST bit is 0.

AVEE Bit (Average Mode Enable)

The AVEE bit selects addition or average mode for A/D conversion of the channel for which A/D conversion and A/D-converted value addition/average mode is selected, including the channels selected in double trigger mode (by ADCSR.DBLANS[4:0] bits), temperature sensor output, and internal reference voltage.
When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the addition count to one time (ADADC.ADC[2:0] = 000b), three times (ADADC.ADC[2:0] = 010b), or 16 times (ADADC.ADC[2:0] = 101b). The mean value of 1-time, 3-time, and 16-time conversion cannot be obtained.
The AVEE bit should be set while the ADCSR.ADST bit is 0.

50.2.9 A/D Control Extended Register (ADCER)

Address(es): S12AD.ADCER 0008 900Eh, S12AD1.ADCER 0008 910Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADRFMT	—	—	—	DIAGM	DIAGLD	DIAGVAL[1:0]		—	—	ACE	—	—	ADPRC[1:0]		—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2, b1	ADPRC[1:0]	A/D Conversion Resolution Setting	b2 b1 0 0: Perform A/D conversion with setting for 12-bit resolution 0 1: Perform A/D conversion with setting for 10-bit resolution 1 0: Perform A/D conversion with setting for 8-bit resolution 1 1: Setting prohibited	R/W
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	ACE	A/D Data Register Automatic Clearing Enable	0: Disables automatic clearing. 1: Enables automatic clearing.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	DIAGVAL[1:0]	Self-Diagnosis Conversion Voltage Select	b9 b8 0 0: Setting prohibited in self-diagnosis voltage fixed mode 0 1: Uses the voltage of 0 V for self-diagnosis. 1 0: Uses the voltage of reference power supply × 1/2 for self-diagnosis. 1 1: Uses the voltage of reference power supply for self-diagnosis.	R/W
b10	DIAGLD	Self-Diagnosis Mode Select	0: Rotation mode for self-diagnosis voltage 1: Fixed mode for self-diagnosis voltage	R/W
b11	DIAGM	Self-Diagnosis Enable	0: Disables self-diagnosis of 12-bit A/D converter. 1: Enables self-diagnosis of 12-bit A/D converter.	R/W
b14 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	ADRFMT	A/D Data Register Format Select	0: 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

ADCER sets self-diagnosis mode, format of A/D data registers y (ADDRy), and automatic clearing of A/D data registers.

ADPRC[1:0] Bits (A/D Conversion Resolution Setting)

The ADPRC[1:0] bits select whether A/D conversion is performed with setting for 8-, 10-, or 12-bit resolution.
When the resolution of the A/D conversion is changed, the bit width of the valid data to be stored in the result register and time for A/D conversion are also changed. For details, refer to section 50.3.7, Analog Input Sampling Time and Scan Conversion Time.
The ADPRC[1:0] bits should be set while the ADCSR.ADST bit is 0.

ACE Bit (A/D Data Register Automatic Clearing Enable)

The ACE bit enables or disables automatic clearing (all 0) of ADDRy, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, or ADOCDR after any of these registers have been read by the CPU and DTC. Automatic clearing of the A/D data register is enabled to detect a failure which has not been updated in the A/D data register.

DIAGVAL[1:0] Bits (Self-Diagnosis Conversion Voltage Select)

The DIAGVAL[1:0] bits select the voltage value used in self-diagnosis voltage fixed mode. For details, refer to the descriptions of the ADCER.DIAGLD bit.
Self-diagnosis should not be executed by setting the ADCER.DIAGLD bit to 1 when the ADCER.DIAGVAL[1:0] bits are set to 00b.

DIAGLD Bit (Self-Diagnosis Mode Select)

The DIAGLD bit selects whether the three voltage values are rotated or the fixed voltage is used in self-diagnosis. Setting this bit (ADCER.DIAGLD) to 0 allows conversion of the voltages in rotation mode where 0, the reference power supply × 1/2, and the reference power supply are converted in this order. When self-diagnosis rotation mode is selected after a reset, self-diagnosis is performed from 0 V. When self-diagnosis voltage fixed mode is selected, the fixed voltage specified by the ADCER.DIAGVAL[1:0] bits is converted. In self-diagnosis voltage rotation mode, the self-diagnosis voltage value does not return to 0 when scan conversion is completed. When scan conversion is restarted, therefore, rotation starts at the voltage value following the previous value. If fixed mode is switched to rotation mode, rotation starts at the fixed voltage value.
The DIAGLD bit should be set while the ADCSR.ADST bit is 0.

DIAGM Bit (Self-Diagnosis Enable)

The DIAGM bit enables or disables self-diagnosis.
Self-diagnosis is used to detect a failure of the 12-bit A/D converter. Specifically, one of the internally generated voltage values 0, the reference power supply × 1/2, and the reference power supply is converted. When conversion is completed, information on the converted voltage and the conversion result is stored into the self-diagnosis data register (ADRD). ADRD can then be read out by software to determine whether the conversion result falls within the normal range (normal) or not (abnormal). Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages is converted. When self-diagnosis is selected in group scan mode, self-diagnosis is separately executed in groups A, B, and C.
The DIAGM bit should be set while the ADCSR.ADST bit is 0.

ADRFMT Bit (A/D Data Register Format Select)

The ADRFMT bit specifies flush-right or flush-left for the data to be stored in ADDRy, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, ADOCDR, ADRD, ADCMPDR0, ADCMPDR1, ADWINLLB, or ADWINULB.
The ADRFMT bit should be set while the ADCSR.ADST bit is 0.

For details on the format of each data register, see section 50.2.1, A/D Data Registers y (ADDRy) (y = 0 to 11), A/D Data Duplication Register (ADDBLDR), A/D Data Duplication Register A (ADDBLDRA), A/D Data Duplication Register B (ADDBLDRB), A/D Temperature Sensor Data Register (ADTSDR), A/D Internal Reference Voltage Data Register (ADOCDR), section 50.2.2, A/D Self-Diagnosis Data Register (ADRD), section 50.2.22, A/D Comparison Function Window A Lower Level Setting Register (ADCMPDR0), section 50.2.23, A/D Comparison Function Window A Upper Level Setting Register (ADCMPDR1), section 50.2.28, A/D Comparison Function Window B Lower Level Setting Register (ADWINLLB), and section 50.2.29, A/D Comparison Function Window B Upper Level Setting Register (ADWINULB).

50.2.10 A/D Conversion Start Trigger Select Register (ADSTRGR)

Address(es): S12AD.ADSTRGR 0008 9010h, S12AD1.ADSTRGR 0008 9110h

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

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	TRSB[5:0]	A/D Conversion Start Trigger Select for Group B	Select the A/D conversion start trigger for group B in group scan mode.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	TRSA[5:0]	A/D Conversion Start Trigger Select	Select the A/D conversion start trigger in single scan mode and continuous scan mode. In group scan mode, the A/D conversion start trigger for group A is selected.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADSTRGR selects the A/D conversion start trigger.

TRSB[5:0] Bits (A/D Conversion Start Trigger Select for Group B)

The TRSB[5:0] bits select the trigger to start scanning of the analog input selected in group B. The TRSB[5:0] bits require to be set only in group scan mode and are not used in any other scan mode. For the scan conversion start trigger for group B, setting a software trigger or an asynchronous trigger is prohibited. Therefore, the TRSB[5:0] bits should be set to the value other than 000000b and the ADCSR.TRGE bit should be set to 1 in group scan mode.

When two groups are selected (ADGCTRGR.GRCE = 0) during group priority operation in group scan mode, setting the ADGSPCR.GBRP bit to 1 allows group B to continuously operate in single scan mode. When setting the ADGSPCR.GBRP bit to 1, set the TRSB[5:0] bits to 3Fh. Note that the issuance period of trigger for A/D conversion must be more than or equal to the actual scan conversion time (t_{SCAN}). If the issuance period is less than t_{SCAN}, A/D conversion by the trigger may have no effect.

When the trigger from the module (MTU) operated in 120 MHz is selected as an A/D conversion start trigger, a delay of the period for synchronization processing occurs. See section 50.3.7, Analog Input Sampling Time and Scan Conversion Time for details.

Table 50.9 lists the A/D conversion startup sources selected by the TRSB[5:0] bits.

TRSA[5:0] Bits (A/D Conversion Start Trigger Select)

The TRSA[5:0] bits select the trigger to start A/D conversion in single scan mode and continuous scan mode. In group scan mode, the trigger to start scanning of the analog input selected in group A is selected. When scanning is executed in group scan mode or double trigger mode, set the ADCSR.TRGE bit to 1.

- When using the A/D conversion startup source of a synchronous trigger, set the ADCSR.TRGE bit to 1 and set the ADCSR.EXTRG bit to 0.
- When using the asynchronous trigger, set the ADCSR.TRGE bit to 1 and set the ADCSR.EXTRG bit to 1.
- Software trigger (ADCSR.ADST) is enabled regardless of the settings of the ADCSR.TRGE bit, the ADCSR.EXTRG bit, and the TRSA[5:0] bits.

Note that the issuance period of trigger for A/D conversion must be more than or equal to the actual scan conversion time (t_{SCAN}). If the issuance period is less than t_{SCAN}, A/D conversion by a trigger may have no effect. When the trigger from the module (MTU) operated in 120 MHz is selected as an A/D conversion start trigger, a delay of the period for synchronization processing occurs. See section 50.3.7, Analog Input Sampling Time and Scan Conversion Time

for details.

Table 50.10 lists the selection of A/D conversion start sources selected by the TRSA[5:0] bits.

Table 50.9 Selection of A/D Activation Sources by the TRSB[5:0] Bits

Module	Source	Remarks	TRSB[5]	TRSB[4]	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
Trigger source deselection state			1	1	1	1	1	1
MTU	TRGA0N	Compare match/input capture from MTU0.TGRA	0	0	0	0	0	1
	TRGA1N	Compare match/input capture from MTU1.TGRA	0	0	0	0	1	0
	TRGA2N	Compare match/input capture from MTU2.TGRA	0	0	0	0	1	1
	TRGA3N	Compare match/input capture from MTU3.TGRA	0	0	0	1	0	0
	TRGA4N	Compare match/input capture from MTU4.TGRA or underflow (trough) of MTU4.TCNT in complementary PWM mode	0	0	0	1	0	1
	TRGA6N	Compare match/input capture from MTU6.TGRA	0	0	0	1	1	0
	TRGA7N	Compare match/input capture from MTU7.TGRA or underflow (trough) of MTU7.TCNT in complementary PWM mode	0	0	0	1	1	1
	TRG0N	Compare match from MTU0.TGRE	0	0	1	0	0	0
	TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT	0	0	1	0	0	1
	TRG4BN	Compare match between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	0
	TRG4AN or TRG4BN	Compare match between MTU4.TADCORA and MTU4.TCNT, or compare match between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	1
	TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT, and compare match between MTU4.TADCORB and MTU4.TCNT (when interrupt skipping function 2 is used)	0	0	1	1	0	0
	TRG7AN	Compare match between MTU7.TADCORA and MTU7.TCNT	0	0	1	1	0	1
	TRG7BN	Compare match between MTU7.TADCORB and MTU7.TCNT	0	0	1	1	1	0
	TRG7AN or TRG7BN	Compare match between MTU7.TADCORA and MTU7.TCNT, or compare match between MTU7.TADCORB and MTU7.TCNT	0	0	1	1	1	1
	TRG7ABN	Compare match between MTU7.TADCORA and MTU7.TCNT, and compare match between MTU7.TADCORB and MTU7.TCNT (when interrupt skipping function 2 is used)	0	1	0	0	0	0
TMR	TMTRG0AN_0	Compare match between TMR0.TCORA and TMR0.TCNT	0	1	1	1	0	1
	TMTRG0AN_1	Compare match between TMR2.TCORA and TMR2.TCNT	0	1	1	1	1	0
TPU	TPTRGAN	Compare match/input capture from TPUn.TGRAn (n = 0 to 5)	0	1	1	1	1	1
	TPTRG0AN	Compare match/input capture from TPU0.TGRA0	1	0	0	0	0	0
ELC	ELCTRG0N /ELCTRG1N	A/D start source from the ELC	1	1	0	0	0	0

Table 50.10 Selection of A/D Activation Sources by the TRSA[5:0] Bits

Module	Source	Remarks	TRSA[5]	TRSA[4]	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
Trigger source deselection state			1	1	1	1	1	1
MTU	TRGA0N	Compare match/input capture from MTU0.TGRA	0	0	0	0	0	1
	TRGA1N	Compare match/input capture from MTU1.TGRA	0	0	0	0	1	0
	TRGA2N	Compare match/input capture from MTU2.TGRA	0	0	0	0	1	1
	TRGA3N	Compare match/input capture from MTU3.TGRA	0	0	0	1	0	0
	TRGA4N	Compare match/input capture from MTU4.TGRA or underflow (trough) of MTU4.TCNT in complementary PWM mode	0	0	0	1	0	1
	TRGA6N	Compare match/input capture from MTU6.TGRA	0	0	0	1	1	0
	TRGA7N	Compare match/input capture from MTU7.TGRA or underflow (trough) of MTU7.TCNT in complementary PWM mode	0	0	0	1	1	1
	TRG0N	Compare match from MTU0.TGRE	0	0	1	0	0	0
	TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT	0	0	1	0	0	1
	TRG4BN	Compare match between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	0
	TRG4AN or TRG4BN	Compare match between MTU4.TADCORA and MTU4.TCNT, or compare match between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	1
	TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT, and compare match between MTU4.TADCORB and MTU4.TCNT (when interrupt skipping function 2 is used)	0	0	1	1	0	0
	TRG7AN	Compare match between MTU7.TADCORA and MTU7.TCNT	0	0	1	1	0	1
	TRG7BN	Compare match between MTU7.TADCORB and MTU7.TCNT	0	0	1	1	1	0
	TRG7AN or TRG7BN	Compare match between MTU7.TADCORA and MTU7.TCNT, or compare match between MTU7.TADCORB and MTU7.TCNT	0	0	1	1	1	1
	TRG7ABN	Compare match between MTU7.TADCORA and MTU7.TCNT, and compare match between MTU7.TADCORB and MTU7.TCNT (when interrupt skipping function 2 is used)	0	1	0	0	0	0
TMR	TMTRG0AN_0	Compare match between TMR0.TCORA and TMR0.TCNT	0	1	1	1	0	1
	TMTRG0AN_1	Compare match between TMR2.TCORA and TMR2.TCNT	0	1	1	1	1	0
TPU	TPTRGAN	Compare match/input capture from TPUn.TGRAn (n = 0 to 5)	0	1	1	1	1	1
	TPTRG0AN	Compare match/input capture from TPU0.TGRA0	1	0	0	0	0	0
ELC	ELCTRG0N /ELCTRG1N	A/D start source from the ELC	1	1	0	0	0	0

50.2.11 A/D Conversion Extended Input Control Register (ADEXICR)

Address(es): S12AD1.ADEXICR 0008 9112h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	EXOEN	EXSEL[1:0]		—	OCSB	TSSB	OCSA	TSSA	—	—	—	—	—	—	OCSAD	TSSAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TSSAD	Temperature Sensor Output A/D-Converted Value Addition/Average Mode Select	0: Temperature sensor output A/D-converted value addition/average mode is not selected. 1: Temperature sensor output A/D-converted value addition/average mode is selected.	R/W
b1	OCSAD	Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select	0: Internal reference voltage A/D-converted value addition/average mode is not selected. 1: Internal reference voltage A/D-converted value addition/average mode is selected.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	TSSA	Temperature Sensor Output A/D Conversion Select	0: A/D conversion of temperature sensor output is not performed. 1: A/D conversion of temperature sensor output is performed.	R/W
b9	OCSA	Internal Reference Voltage A/D Conversion Select	0: A/D conversion of internal reference voltage is not performed. 1: A/D conversion of internal reference voltage is performed.	R/W
b10	TSSB	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
b11	OCSB	Internal Reference Voltage A/D Conversion Select	0: A/D conversion of internal reference voltage is not performed. 1: A/D conversion of internal reference voltage is performed.	R/W
b12	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b14, b13	EXSEL[1:0]	Extended Analog Input Select	b14 b13 0 0: Analog input channels (ANn) 0 1: ANEX1 1 0: Setting prohibited 1 1: Setting prohibited	R/W
b15	EXOEN	Extended Analog Output Control	0: Output disabled 1: Output enabled	R/W

ADEXICR specifies the settings of A/D conversion of the temperature sensor output, internal reference voltage, or extended analog input.

TSSAD Bit (Temperature Sensor Output A/D-Converted Value Addition/Average Mode Select)

When A/D conversion of the temperature sensor output is selected and the TSSAD bit is set to 1, A/D conversion of the temperature sensor output is performed in sequence for the count set in the ADADC.ADC[1:0] bits (2, 3, 4, or 16 times). The added (accumulated) total value is returned to the A/D temperature sensor data register (ADTSDR) when the ADADC.AVEE bit is 0 and the average is returned to ADTSDR when the ADADC.AVEE bit is 1. The TSSAD bit should be set while the ADCSR.ADST bit is 0.

OCSAD Bit (Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select)

When the OCSAD bit is set to 1, A/D conversion of the internal reference voltage is selected and performed successively 2, 3, 4, or 16 times that is set with the ADADC.ADC[2:0] bits. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D internal reference voltage data register (ADOCDR). When the ADADC.AVEE bit is 1, the mean value is stored in ADOCDR.
The OCSAD bit should be set while the ADCSR.ADST bit is 0.

TSSA Bit (Temperature Sensor Output A/D Conversion Select)

This bit selects A/D conversion of the temperature sensor output for group A in single scan mode, sequence scan mode, or group scan mode. When A/D conversion of the temperature sensor output is performed, set the ADCSR.DBLE bit to 0.
The TSSA bit should be set while the ADCSR.ADST bit is 0.

OCSA Bit (Internal Reference Voltage A/D Conversion Select)

This bit selects A/D conversion of the internal reference voltage for group A in single scan mode, sequence scan mode, or group scan mode. When A/D conversion of the internal reference voltage is performed, set the ADCSR.DBLE bit to 0.
The OCSA bit should be set while the ADCSR.ADST bit is 0. Wait for at least 400 ns after setting the OCSA bit to 1 to start A/D conversion.

TSSB Bit (Temperature Sensor Output A/D Conversion Select)

This bit selects A/D conversion of the temperature sensor output for group B in group scan mode.
This bit should be set while the ADCSR.ADST bit is 0. When the TSSA bit is 1, do not set the TSSB bit to 1.

OCSB Bit (Internal Reference Voltage A/D Conversion Select)

This bit selects A/D conversion of the internal reference voltage for group B in group scan mode.
This bit should be set while the ADCSR.ADST bit is 0. When the OCSA bit is 1, do not set the OCSB bit to 1. Wait for at least 400 ns after setting the OCSB bit to 1 to start A/D conversion.

EXSEL[1:0] Bits (Extended Analog Input Select)

These bits can select extended analog input ANEX1 besides analog input channels (ANn).
When ANEX1 is selected, ANEX0 should be input to ANEX1 through an external operational amplifier. In addition, only analog input channels AN100 to AN107 are selectable. Do not select channels AN108 to AN111. For details, refer to section 50.3.5.1, Usage of ANEX1.

EXOEN Bit (Extended Analog Output Control)

This bit controls the extended analog output (ANEX0). When the output is enabled, the multiplexed value of AN100 to AN107 among the analog input channels of unit 1. Do not enable output while the EXSEL[1:0] bits are 00b.

50.2.12 A/D Group C Extended Input Control Register (ADGCEXCR)

Address(es): S12AD1.ADGCEXCR 0008 91D8h

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

Bit	Symbol	Bit Name	Description	R/W
b0	TSSC	Group C Temperature Sensor Output A/D Conversion Select	0: Temperature sensor output is not A/D converted 1: Temperature sensor output is A/D converted	R/W
b1	OCSC	Group C Internal Reference Voltage A/D Conversion Select	0: Internal reference voltage is not A/D converted 1: Internal reference voltage is A/D converted	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The ADGCEXCR register specifies extended input for group C.

TSSC Bit (Group C Temperature Sensor Output A/D Conversion Select)

This bit selects A/D conversion of the temperature sensor output for group C in group scan mode.

The TSSC bit should be set while the ADCSR.ADST bit is 0. The TSSC bit should not be set to 1 while the TSSA or TSSB bit is 1.

OCSC Bit (Group C Internal Reference Voltage A/D Conversion Select)

This bit selects A/D conversion of the internal reference voltage for group C in group scan mode.

The OCSC bit should be set while the ADCSR.ADST bit is 0. The OCSC bit should not be set while the OCSA or OCSB bit is 1. Wait for at least 400 ns after setting the OCSC bit to 1 to start A/D conversion.

50.2.13 A/D Group C Trigger Select Register (ADGCTRGR)

Address(es): S12AD.ADGCTRGR 0008 90D9h, S12AD1.ADGCTRGR 0008 91D9h

	b7	b6	b5	b4	b3	b2	b1	b0
	GRCE	GCADIE	TRSC[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	TRSC[5:0]	Group C A/D Conversion Start Trigger Select	Select the A/D conversion start trigger for group C in group scan mode.	R/W
b6	GCADIE	Group C Scan End Interrupt Enable	0: Disables interrupt generation after completion of group C scan 1: Enables interrupt generation after completion of group C scan	R/W
b7	GRCE	Group C A/D Conversion Operation Enable	Enables A/D conversion operation for group C. 0: Group C is not used 1: Group C is used	R/W

ADGCTRGR enables operation for group C and selects the A/D conversion start trigger. For details on group priority operation, see Table 50.16 and Table 50.17.

TRSC[5:0] Bits (Group C A/D Conversion Start Trigger Select)

These bits select the trigger to start scanning of the analog input selected in group C. These bits are used for group scan mode only; not used for any other modes. Software trigger or asynchronous trigger cannot be set as the scan conversion trigger for group C. When using group C in group scan mode, set the TRSC[5:0] bits to a value other than 000000b, set the ADCSR.TRGE bit to 1, and set the GRCE bit to 1.

When group C is used during group priority control in group scan mode and the ADGSPCR.GBRP bit is set to 1, group C can be continuously operated in single scan mode. When continuously operating group C in single scan mode, set the TRSC[5:0] bits to 3Fh and disable trigger selection.

Note that the issuance period of trigger for A/D conversion must be more than or equal to the actual scan conversion time (t_{SCAN}). If the issuance period is less than tSCAN, A/D conversion by a trigger may have no effect.

When the trigger from the module (MTU) operated in 120 MHz is selected as an A/D conversion start trigger, a delay of the period for synchronization processing occurs. See section 50.3.7, Analog Input Sampling Time and Scan Conversion Time for details.

Table 50.11 lists the selection of A/D conversion start sources selected by the TRSA[5:0] bits for group C.

Table 50.11 Selection of A/D Activation Sources by the TRSC[5:0] Bits

Module	Source	Remarks	TRSC[5]	TRSC[4]	TRSC[3]	TRSC[2]	TRSC[1]	TRSC[0]
Trigger source deselection state			1	1	1	1	1	1
MTU	TRGA0N	Compare match/input capture from MTU0.TGRA	0	0	0	0	0	1
	TRGA1N	Compare match/input capture from MTU1.TGRA	0	0	0	0	1	0
	TRGA2N	Compare match/input capture from MTU2.TGRA	0	0	0	0	1	1
	TRGA3N	Compare match/input capture from MTU3.TGRA	0	0	0	1	0	0
	TRGA4N	Compare match/input capture from MTU4.TGRA or underflow (trough) of MTU4.TCNT in complementary PWM mode	0	0	0	1	0	1
	TRGA6N	Compare match/input capture from MTU6.TGRA	0	0	0	1	1	0
	TRGA7N	Compare match/input capture from MTU7.TGRA or underflow (trough) of MTU7.TCNT in complementary PWM mode	0	0	0	1	1	1
	TRG0N	Compare match from MTU0.TGRE	0	0	1	0	0	0
	TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT	0	0	1	0	0	1
	TRG4BN	Compare match between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	0
	TRG4AN or TRG4BN	Compare match between MTU4.TADCORA and MTU4.TCNT, or compare match between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	1
	TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT, and compare match between MTU4.TADCORB and MTU4.TCNT (when interrupt skipping function 2 is used)	0	0	1	1	0	0
	TRG7AN	Compare match between MTU7.TADCORA and MTU7.TCNT	0	0	1	1	0	1
	TRG7BN	Compare match between MTU7.TADCORB and MTU7.TCNT	0	0	1	1	1	0
	TRG7AN or TRG7BN	Compare match between MTU7.TADCORA and MTU7.TCNT, or compare match between MTU7.TADCORB and MTU7.TCNT	0	0	1	1	1	1
	TRG7ABN	Compare match between MTU7.TADCORA and MTU7.TCNT, and compare match between MTU7.TADCORB and MTU7.TCNT (when interrupt skipping function 2 is used)	0	1	0	0	0	0
TMR	TMTRG0AN_0	Compare match between TMR0.TCORA and TMR0.TCNT	0	1	1	1	0	1
	TMTRG0AN_1	Compare match between TMR2.TCORA and TMR2.TCNT	0	1	1	1	1	0
TPU	TPTRGAN	Compare match/input capture from TPUn.TGRAn (n = 0 to 5)	0	1	1	1	1	1
	TPTRG0AN	Compare match/input capture from TPU0.TGRA0	1	0	0	0	0	0
ELC	ELCTRG0N /ELCTRG1N	A/D start source from the ELC	1	1	0	0	0	0

GCADIE Bit (Group C Scan End Interrupt Enable)

This bit enables or disables scan end interrupt generation for group C. A scan end interrupt for group C is provided individually for each unit. Table 50.12 shows the relationship between each unit and the scan end interrupt for group C.

Table 50.12 Relationship between Each Unit and Group C Scan End Interrupt

Unit	Group C Scan End Interrupt
S12AD	S12GCADI
S12AD1	S12GCADI1

GRCE Bit (Group C A/D Conversion Operation Enable)

When using group C in group scan mode, set the GRCE bit to 1.

When the GRCE bit is 0, trigger input for group C is disabled.

During group priority operation (the ADGSPCR.PGS bit is 1) with group C used, when the ADGSPCR.GBRP bit is set to 1, single scan for group C is continuously operated. When the GRCE bit is set to 1, single scan for group B is not continuously operated.

The GRCE bit should be set while the ADCSR.ADST bit is 0.

50.2.14 A/D Sampling State Register n (ADSSTRn) (n = 0 to 11, T, O)

Address(es): S12AD.ADSSTR0 0008 90E0h, S12AD.ADSSTR1 0008 90E1h, S12AD.ADSSTR2 0008 90E2h, S12AD.ADSSTR3 0008 90E3h, S12AD.ADSSTR4 0008 90E4h, S12AD.ADSSTR5 0008 90E5h, S12AD.ADSSTR6 0008 90E6h, S12AD.ADSSTR7 0008 90E7h, S12AD1.ADSSTRT 0008 91DEh, S12AD1.ADSSTRO 0008 91DFh, S12AD1.ADSSTR0 0008 91E0h, S12AD1.ADSSTR1 0008 91E1h, S12AD1.ADSSTR2 0008 91E2h, S12AD1.ADSSTR3 0008 91E3h, S12AD1.ADSSTR4 0008 91E4h, S12AD1.ADSSTR5 0008 91E5h, S12AD1.ADSSTR6 0008 91E6h, S12AD1.ADSSTR7 0008 91E7h, S12AD1.ADSSTR8 0008 91E8h, S12AD1.ADSSTR9 0008 91E9h, S12AD1.ADSSTR10 0008 91EAh, S12AD1.ADSSTR11 0008 91EBh

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

The ADSSTRn register sets the sampling time for analog input.
If one state is one ADCLK (A/D conversion clock) cycle and the ADCLK clock is 60 MHz, one state is 16.7 ns. The initial value is 11 states. If the impedance of analog input signal source is too high to secure sufficient sampling time or if the ADCLK clock is slow, the sampling time can be adjusted. The ADSSTRn register should be set while the ADCSR.ADST bit is 0. The lower-limit value for sampling time differs depending on the PCLK to ADCLK frequency ratio.
Set a value that is 5 states or more when PCLK to ADCLK frequency ratio = 1:1, 2:1, 4:1, or 8:1.

Table 50.13 shows the relationship between the A/D sampling state register and the relevant channels.
For details, refer to section 50.3.7, Analog Input Sampling Time and Scan Conversion Time.

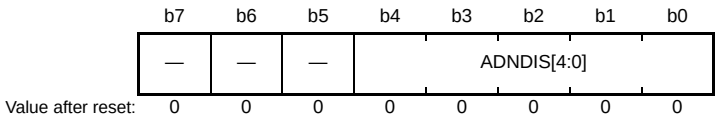
Table 50.13 Relationship between A/D Sampling State Register and Relevant Channels

Unit	Register Name	Relevant Channels
S12AD	ADSSTR0 register	AN000, Self-Diagnosis
	ADSSTR1 register	AN001
	ADSSTR2 register	AN002
	ADSSTR3 register	AN003
	ADSSTR4 register	AN004
	ADSSTR5 register	AN005
	ADSSTR6 register	AN006
	ADSSTR7 register	AN007
S12AD1	ADSSTR0 register	AN100, Self-Diagnosis
	ADSSTR1 register	AN101
	ADSSTR2 register	AN102
	ADSSTR3 register	AN103
	ADSSTR4 register	AN104
	ADSSTR5 register	AN105
	ADSSTR6 register	AN106
	ADSSTR7 register	AN107
	ADSSTR8 register	AN108
	ADSSTR9 register	AN109
	ADSSTR10 register	AN110
	ADSSTR11 register	AN111
	ADSSTRT register	Temperature sensor output
	ADSSTRO register	Internal reference voltage

Note 1. When performing A/D conversion of the temperature sensor output or internal reference voltage, the sampling time should be 5 µs or longer.

50.2.15 A/D Disconnection Detection Control Register (ADDISCR)

Address(es): S12AD.ADDISCR 0008 907Ah, S12AD1.ADDISCR 0008 917Ah



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	ADNDIS[4:0]	A/D Disconnection Detection Assist Setting	b4 ADNDIS[4]: Discharge/precharge selected 0: Discharge 1: Precharge b3 to b0 ADNDIS[3:0]: Discharge/precharge period	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADDISCR sets the disconnection detection assist function.

ADNDIS[4:0] Bits (A/D Disconnection Detection Assist Setting)

These bits select either precharge or discharge and the period of precharge/discharge for the A/D disconnection detection assist function. Setting the ADNDIS[4] bit = 1 allows to select precharge and setting the ADNDIS[4] bit = 0 allows to select discharge. The period of precharge/discharge can be set with the ADNDIS[3:0] bits. When the ADNDIS[3:0] bits = 0000b, the disconnection detection assist function is not effective. Setting of the ADNDIS[3:0] bits to 0001b is prohibited. Except for the case of ADNDIS[3:0] = 0000b or 0001b, the specified value indicates the number of states for the period of precharge/discharge. The ADNDIS[4:0] bits should be set when the ADCSR.ADST bit is 0.
The disconnection detection assist function cannot be used when the temperature sensor output or internal reference voltage is converted or when the self-diagnosis function is used.

50.2.16 A/D Group Scan Priority Control Register (ADGSPCR)

Address(es): S12AD.ADGSPCR 0008 9080h, S12AD1.ADGSPCR 0008 9180h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	GBRP	LGRRS	—	—	—	—	—	—	—	—	—	—	—	—	GBRSCN	PGS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PGS	Group Priority Control Setting*1	0: Operation is without group priority control 1: Operation is with group priority control	R/W
b1	GBRSCN	Low-Priority Group Restart Setting*2	(Enabled only when PGS = 1. Reserved when PGS = 0.) 0: Scanning for the group is not restarted after having been discontinued due to group priority control. 1: Scanning for the group is restarted after having been discontinued due to group priority control.	R/W
b13 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	LGRRS	Restart Channel Select	(Enabled when PGS = 1 and GBRSCN = 1. Reserved when PGS = 0 or GBRSCN = 0.) 0: Scanning is restarted from the scan start channel. 1: Scanning is restarted from the channel on which A/D conversion is not completed.	R/W
b15	GBRP	Single Scan Continuous Start*3	(Enabled only when PGS = 1. Reserved when PGS = 0.) 0: Single scan is not continuously activated. 1: Single scan for the lowest-priority group is continuously activated.	R/W

Note 1. When the PGS bit is to be set to 1, the ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode). If the bits are set to any other values, proper operation is not guaranteed.
Note 2. When the GBRSCN bit is to be set to 1, the frequency ratio of peripheral module clock PCLK to A/D conversion clock ADCLK should be set to 1:1.
Note 3. When the GBRP bit has been set to 1, single scan is performed continuously for the lowest-priority group regardless of the setting of the GBRSCN bit.

ADGSPCR is used to discontinue scanning of the low-priority group and make settings for priority control of scanning for the priority group in group scan mode.
For the settings on group priority operation, see Table 50.16 and Table 50.17.

PGS Bit (Group Priority Control Setting)

This bit sets the priority of operation in group scan mode. Set this bit to 1 when giving priority to operation on the group. When the PGS bit is to be set to 1, the ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode).
During group priority operation, a trigger to start scanning for the priority group is accepted during scan for the low-priority group, and scan for the priority group is started after scan for the low-priority group was discontinued. The priority order is group A > group B > group C.
When a trigger to start scanning for group B is accepted during scan for group C, group C scan is discontinued, and scan for group B is started. When a trigger to start scanning for group A is accepted during scan for group C, group C scan is discontinued, and scan for group A is started.
Likewise, when a trigger to start scanning for group A is accepted during scan for group B, group B scan is discontinued and scan for group A is started.
When setting the PGS bit to 0, clearing should be performed by software according to section 50.6.2, Notes on Stopping A/D Conversion. When setting the PGS bit to 1, follow the procedure described in section 50.3.4.3, Operation under Group Priority Control.

GBRSCN Bit (Low-Priority Group Restart Setting)

This bit controls the restarting of scan operation during group priority control.
If a scan operation on the low-priority group has been stopped by a priority group trigger input with the GBRSCN bit set to 1, the scan operation is restarted after the scanning of the priority group is completed. Also, if a low-priority trigger is input during scan for the priority group, the scan operation on the low-priority group is restarted after the scan for the priority group is completed.
If the GBRSCN bit has been set to 0, triggers that are input during A/D conversion are ignored. Also, the ADCSR.ADST bit must be 0 when the GBRSCN bit is to be set.
The setting of the GBRSCN bit is enabled when the PGS bit is set to 1.

LGRRS Bit (Restart Channel Select)

This bit sets the channel on which scan is restarted during group priority control. The setting of the LGRRS bit is enabled when the PGS and GBRSCN bits are set to 1.
If a scan operation on the low-priority group has been stopped due to group priority operation with the LGRRS bit set to 0, the scan operation is restarted from the start channel after the scan for the priority group is completed.
If a scan operation on the low-priority group has been stopped due to group priority operation with the LGRRS bit set to 1, the scan operation is restarted*1 on the channel on which A/D conversion is not competed after the scan for the priority group is completed.
The LGRRS bit should be set while the ADCSR.ADST bit is 0.

Note 1. If A/D conversion on the addition set channel is not completed for the set number of times when scanning is stopped, A/D conversion on the channel is restarted for the set number of times when scanning is restarted.

GBRP Bit (Single Scan Continuous Start)

This bit is set when the lowest-priority group is continuously operated in single scan mode while group priority operation is set. The lowest-priority group is group C when groups A, B, and C are used; group B when groups A and B are used.
Setting the GBRP bit to 1 starts a single scan on the lowest-priority group. On completion of the scan, another single scan on the lowest-priority group is automatically started.
If scanning has been stopped due to group priority operation, single scan on the lowest-priority group is automatically restarted on completion of the A/D conversion on the priority group.
Disable the trigger input for the lowest-priority group before setting the GBRP bit to 1. When the GBRP bit is set to 1, only the lowest-priority group is scanned again even if the GBRSCN bit is 0.
The ADCSR.ADST bit must be 0 when the GBRP bit is to be set.
The setting of the GBRP bit is enabled when the PGS bit is 1.

50.2.17 A/D Comparison Function Control Register (ADCMPCR)

Address(es): S12AD.ADCMPCR 0008 9090h, S12AD1.ADCMPCR 0008 9190h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CMPAIE	WCMPE	CMPBIE	—	CMPAE	—	CMPBE	—	—	—	—	—	—	—	CMPAB[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	CMPAB[1:0]	Window A/B Complex Conditions Setting	b1 b0 0 0: Window A comparison condition matched OR window B comparison condition matched 0 1: Window A comparison condition matched EXOR window B comparison condition matched 1 0: Window A comparison condition matched AND window B comparison condition matched 1 1: Setting is prohibited	R/W
b8 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9	CMPBE	Comparison Window B Enable	0: Comparison window B disabled 1: Comparison window B enabled	R/W
b10	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b11	CMPAE	Comparison Window A Enable	0: Comparison window A disabled 1: Comparison window A enabled	R/W
b12	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b13	CMPBIE	Comparison Window B Interrupt Enable	0: Comparison interrupt by a match with the comparison condition (window B) is disabled 1: Comparison interrupt by a match with the comparison condition (window B) is enabled	R/W
b14	WCMPE	Window Function Setting	0: The window function is disabled Window A or B operates as a comparator for comparing between one value on the low side and the A/D converted value 1: The window function is enabled Window A or B operates as a window comparator for comparing between two values on the high and low sides and the A/D converted value	R/W
b15	CMPAIE	Comparison Window A Interrupt Enable	0: Comparison interrupt by a match with the comparison condition (window A) is disabled 1: Comparison interrupt by a match with the comparison condition (window A) is enabled	R/W

The ADCMPCR register is used to set the settings for the comparison window function (windows A and B).

CMPAB[1:0] Bits (Window A/B Complex Conditions Setting)

The CMPAB[1:0] bits are enabled when single scan mode and window A/B are both enabled (CMPAE bit = 1 and CMPBE bit = 1). These bits specify the monitor conditions of the ADWINMON.MONCOMB bits. The CMPAB[1:0] bits should be set when the ADCSR.ADST bit is 0.

CMPBE Bit (Comparison Window B Enable)

This bit is used to disable or enable comparison window B. The CMPBE bit should be set when the ADCSR.ADST bit is 0.

To set the following registers, set this bit to 0.

- A/D channel select registers A0/B0 (ADANSA0 and ADANSB0)
- Bits OCSB, TSSB, OCSA, and TSSA in the A/D conversion extended input control register (ADEXICR.OCSB, TSSB, OCSA, and TSSA)

- Bits OCSC and TSSC in the A/D group C extended input control register (ADGCEXCR.OCSC and TSSC)
- The CMPCHB[5:0] bits in the window B channel select register (ADCMPBNSR.CMPCHB[5:0])

CMPAE Bit (Comparison Window A Enable)

This bit is used to disable or enable comparison window A. The CMPAE bit should be set when the ADCSR.ADST bit is 0.

To set the following registers, set this bit to 0.

- A/D channel select registers A0/B0 (ADANSA0 and ADANSB0)
- Bits OCSB, TSSB, OCSA, and TSSA in the A/D conversion extended input control register (ADEXICR.OCSB, TSSB, OCSA, and TSSA)
- Bits OCSC and TSSC in the A/D group C extended input control register (ADGCEXCR.OCSC and TSSC)
- Window A channel select registers 0 (ADCMPANSR0)
- Window A extended input select register (ADCMPANSER)

CMPBIE Bit (Comparison Window B Interrupt Enable)

Enables or disables comparison interrupts by a window B comparison condition match. A single comparison interrupt exists for each unit. Table 50.14 lists the relationship between each unit and its comparison interrupt.

WCMPE Bit (Window Function Setting)

This bit is used to disable or enable the window function. The WCMPE bit should be set when the ADCSR.ADST bit is 0.

CMPAIE Bit (Comparison Window A Interrupt Enable)

Enables or disables comparison interrupts by a window A comparison condition match. A single comparison interrupt exists for each unit. Table 50.14 lists the relationship between each unit and its comparison interrupt.

Table 50.14 Relationship Between Each Unit and its Comparison Interrupt Unit

	Comparison Interrupt	
Unit	**When Window A Comparison Condition is Met**	**When Window B Comparison Condition is Met**
S12AD	S12CMPAI	S12CMPBI
S12AD1	S12CMPAI1	S12CMPBI1

50.2.18 A/D Comparison Function Window A Channel Select Register 0 (ADCMPANSR0)

(1) S12AD.ADCMPANSR0

Address(es): 0008 9094h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CMPCHA007	CMPCHA006	CMPCHA005	CMPCHA004	CMPCHA003	CMPCHA002	CMPCHA001	CMPCHA000
Value after 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	CMPCHA000	Comparison Window A Channel Select	0: AN000 to AN007 are excluded from the targets for comparison window A 1: AN000 to AN007 are included as the targets for comparison window A	R/W
b1	CMPCHA001			R/W
b2	CMPCHA002			R/W
b3	CMPCHA003			R/W
b4	CMPCHA004			R/W
b5	CMPCHA005			R/W
b6	CMPCHA006			R/W
b7	CMPCHA007			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register selects analog inputs AN000 to AN007 of the channels that perform comparison with the conditions of comparison window A.

CMPCHA0n Bit (n = 00 to 07) (Comparison Window A Channel Select)

The comparison function is enabled when the CMPCHA0n bit with the same index number as the A/D conversion channel selected by the ADANSA0.ANSA0n bit (n = 00 to 07), ADANSB0.ANSB0n bit, and ADANSC0.ANSC0n bit is set to 1.

The CMPCHA0n bit should be set to 1 while the ADCSR.ADST bit is 0.

(2) S12AD1.ADCMPANSR0

Address(es): 0008 9194h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CMPCHA011	CMPCHA010	CMPCHA009	CMPCHA008	CMPCHA007	CMPCHA006	CMPCHA005	CMPCHA004	CMPCHA003	CMPCHA002	CMPCHA001	CMPCHA000
Value after 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	CMPCHA000	Comparison Window A Channel Select	0: AN100 to AN111 are excluded from the targets for comparison window A 1: AN100 to AN111 are included as the targets for comparison window A	R/W
b1	CMPCHA001			R/W
b2	CMPCHA002			R/W
b3	CMPCHA003			R/W
b4	CMPCHA004			R/W
b5	CMPCHA005			R/W
b6	CMPCHA006			R/W
b7	CMPCHA007			R/W
b8	CMPCHA008			R/W
b9	CMPCHA009			R/W
b10	CMPCHA010			R/W
b11	CMPCHA011			R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register selects analog inputs AN100 to AN111 of the channels that perform comparison with the conditions of comparison window A.

CMPCHA0n Bit (n = 00 to 11) (Comparison Window A Channel Select)

The comparison function is enabled when the CMPCHA0n bit with the same index number as the A/D conversion channel selected by the ADANSA0.ANSA0n bit (n = 00 to 11), ADANSB0.ANSB0n bit, and ADANSC0.ANSC0n bit is set to 1.

The CMPCHA0n bit should be set while the ADCSR.ADST bit is 0.

50.2.19 A/D Comparison Function Window A Extended Input Select Register (ADCMPANSER)

Address(es): S12AD1.ADCMPANSER 0008 9192h

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

Bit	Symbol	Bit Name	Description	R/W
b0	CMPSTS	Temperature Sensor Output Comparison Select	0: Temperature sensor output is excluded from the targets for comparison window A 1: Temperature sensor output is included as the targets for comparison window A	R/W
b1	CMPSOC	Internal Reference Voltage Compare Select	0: Internal reference voltage is excluded from the targets for comparison window A 1: Internal reference voltage is included as the targets for comparison window A	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register selects whether the temperature sensor output or internal reference voltage is compared under the conditions of comparison window A.

CMPSTS Bit (Temperature Sensor Output Comparison Select)

The comparison window A is enabled when the CMPSTS bit is set to 1 while the ADEXICR.TSSA, ADEXICR.TSSB, or ADGCEXCR.TSSC bit is 1. The CMPSTS bit should be set while the ADCSR.ADST bit is 0.

CMPSOC Bit (Internal Reference Voltage Compare Select)

The comparison window A function is enabled when the ADEXICR.OCSA, ADEXICR.OCSB, or ADGCEXCR.OCSC bit is 1. The CMPSOC bit should be set while the ADCSR.ADST bit is 0.

50.2.20 A/D Comparison Function Window A Comparison Condition Setting Register 0 (ADCMPLR0)

(1) S12AD.ADCMPLR0

Address(es): 0008 9098h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CMPLCHA007	CMPLCHA006	CMPLCHA005	CMPLCHA004	CMPLCHA003	CMPLCHA002	CMPLCHA001	CMPLCHA000
Value after 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	CMPLCHA000	Comparison Window A Comparison Condition Select	When the window function is disabled (the ADCMPCR.WCMPE bit is 0) 0: ADCMPDR0 register value > A/D converted value 1: ADCMPDR0 register value < A/D converted value When the window function is enabled (the ADCMPCR.WCMPE bit is 1) 0: A/D converted value < ADCMPDR0 register value or ADCMPDR1 register value < A/D converted value 1: ADCMPDR0 register value < A/D converted value < ADCMPDR1 register value	R/W
b1	CMPLCHA001			R/W
b2	CMPLCHA002			R/W
b3	CMPLCHA003			R/W
b4	CMPLCHA004			R/W
b5	CMPLCHA005			R/W
b6	CMPLCHA006			R/W
b7	CMPLCHA007			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register specifies the conditions for comparison of the values in the S12AD.ADCMPDR0/ADCMPDR1 register and the value of A/D conversion. The S12AD.ADCSR.ADST bit is 0.

CMPLCHA0n Bit (n = 00 to 07) (Comparison Window A Comparison Condition Select)

This bit specifies the comparison conditions of the channels (AN000 to AN007) targeted for the window A comparison conditions. The condition can be specified for each analog input for comparison. The CMPLCHA000 bit corresponds to AN000 and the CMPLCHA007 bit corresponds to AN007.

(2) S12AD1.ADCMPLR0

Address(es): 0008 9198h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CMPLCHA011	CMPLCHA010	CMPLCHA009	CMPLCHA008	CMPLCHA007	CMPLCHA006	CMPLCHA005	CMPLCHA004	CMPLCHA003	CMPLCHA002	CMPLCHA001	CMPLCHA000
Value after 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	CMPLCHA000	Comparison Window A Comparison Condition Select	When the window function is disabled (the ADCMPCR.WCMPE bit is 0) 0: ADCMPDR0 register value > A/D converted value 1: ADCMPDR0 register value < A/D converted value When the window function is enabled (the ADCMPCR.WCMPE bit is 1) 0: A/D converted value < ADCMPDR0 register value or ADCMPDR1 register value < A/D converted value 1: ADCMPDR0 register value < A/D converted value < ADCMPDR1 register value	R/W
b1	CMPLCHA001			R/W
b2	CMPLCHA002			R/W
b3	CMPLCHA003			R/W
b4	CMPLCHA004			R/W
b5	CMPLCHA005			R/W
b6	CMPLCHA006			R/W
b7	CMPLCHA007			R/W
b8	CMPLCHA008			R/W
b9	CMPLCHA009			R/W
b10	CMPLCHA010			R/W
b11	CMPLCHA011			R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register specifies the conditions for comparing between the values of the S12AD1.ADCMPDR0/ADCMPDR1 registers and the result of A/D conversion. The S12AD1.ADCMPLR0 register should be set while the S12AD1.ADCSR.ADST bit is 0.

CMPLCHA0n Bit (n = 00 to 11) (Comparison Window A Comparison Condition Select)

This bit specifies the conditions for comparison with the target channels for window A (AN100 to AN111). The conditions can be specified for each analog input for comparison. The CMPLCHA000 bit corresponds to AN100 and the CMPLCHA011 bit corresponds to AN111.

When the result of comparison of each analog input matches with the pre-set condition, the S12AD1.ADCMPSR0.CMPSTCHA0n flag (n = 00 to 11) is set to 1 and a comparison interrupt (S12CMPAI1) is generated. The conditions for comparison are shown in Figure 50.4.

(1) Comparison condition when the window function is disabled

CMPLCHA0n = 0

ADCMPDR0 register value ≤ A/D converted value	Not matched
ADCMPDR0 register value > A/D converted value	Matched

CMPLCHA0n = 1

ADCMPDR0 register value < A/D converted value	Matched
ADCMPDR0 register value ≥ A/D converted value	Not matched

(2) Comparison condition when the window function is enabled

CMPLCHA0n = 0

ADCMPDR1 register value < A/D converted value	Matched
ADCMPDR0 register value ≤ A/D converted value ≤ ADCMPDR1 register value	Not matched
A/D converted value < ADCMPDR0 register value	Matched

CMPLCHA0n = 1

ADCMPDR1 register value ≤ A/D converted value	Not matched
ADCMPDR0 register value < A/D converted value < ADCMPDR1 register value	Matched
A/D converted value ≤ ADCMPDR0 register value	Not matched

Figure 50.4 Details of Comparison Condition: Comparison Function Window A

50.2.21 A/D Comparison Function Window A Extended Input Comparison Condition Setting Register (ADCMPLER)

Address(es): S12AD1.ADCMPLER 0008 9193h

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

Bit	Symbol	Bit Name	Description	R/W
b0	CMPLTS	Comparison Window A Temperature Sensor Output Comparison Condition Select	When the window A function is disabled (the ADCMPCR.WCMPE bit is 0) 0: ADCMPDR0 register value > A/D converted value 1: ADCMPDR0 register value < A/D converted value When the window A function is enabled (the ADCMPCR.WCMPE bit is 1) 0: A/D converted value < ADCMPDR0 register value or A/D converted value > ADCMPDR1 register value 1: ADCMPDR0 register value < A/D converted value < ADCMPDR1 register value	R/W
b1	CMPLOC	Comparison Window A Internal Reference Voltage Comparison Condition Select	When the window A function is disabled (the ADCMPCR.WCMPE bit is 0) 0: ADCMPDR0 register value > A/D converted value 1: ADCMPDR0 register value < A/D converted value When the window A function is enabled (the ADCMPCR.WCMPE bit is 1) 0: A/D converted value < ADCMPDR0 register value or A/D converted value > ADCMPDR1 register value 1: ADCMPDR0 register value < A/D converted value < ADCMPDR1 register value	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register specifies the conditions for comparison of the value of the ADCMPDR0/ADCMPDR1 register and the result of A/D conversion. The ADCMPLER register should be set while the ADCSR.ADST bit is 0.

CMPLTS Bit (Comparison Window A Temperature Sensor Output Comparison Condition Select)

This bit selects the comparison conditions of temperature sensor output targeted for window A.
When the result of comparison of temperature sensor output matches with the pre-set condition, the ADCMPSER.CMPFTS flag is set to 1. A comparison interrupt (S12CMPAI1) is generated.
Figure 50.4 shows the conditions for comparison.

CMPLOC Bit (Comparison Window A Internal Reference Voltage Comparison Condition Select)

This bit selects the comparison conditions of the internal reference voltage targeted for window A.
When the result of comparison of the internal reference voltage matches with the pre-set condition, the ADCMPSER.CMPFOC flag is set to 1. A comparison interrupt (S12CMPAI1) is generated.
Figure 50.4 shows the conditions for comparison.

50.2.22 A/D Comparison Function Window A Lower Level Setting Register (ADCMPDR0)

Address(es): S12AD.ADCMPDR0 0008 909Ch, S12AD1.ADCMPDR0 0008 919Ch

	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

This is a readable and writable register that specifies the reference data when the comparison window A function is used. The ADCMPDR0 register specifies the lower level of window A.
Writing to the ADCMPDR0 register is enabled even when A/D conversion is in progress. Rewriting the register value during the A/D conversion dynamically changes the reference data.
Satisfy the condition of [Upper limit level ≥ Lower limit level (ADCMPDR1 setting value ≥ ADCMPDR0 setting value)] when setting.

The data formats of the register for each given condition are shown below.

- Setting of the A/D data register format select bit (flush-right or -left format)
- Setting of the A/D conversion resolution setting bit (12, 10, or 8 bits)
- Setting of the A/D converted value addition/averaging function channel select register (when A/D converted value average mode is selected or not selected)
- Setting of the A/D converted value addition/average count select register (when addition/averaging mode is selected and addition count is selected)

Note: When the comparison values are set in a different format from that of the A/D data register y (ADDRy), correct result of comparison cannot be obtained.

(1) When A/D converted value addition/averaging mode is not selected

- Flush-right format with setting for 12-bit resolution
  The comparison level (lower) to be compared is set in bits 11 to 0. Write 0 to bits 15 to 12.
- Flush-right format with setting for 10-bit resolution
  The comparison level (lower) to be compared is set in bits 9 to 0. Write 0 to bits 15 to 10.
- Flush-right format with setting for 8-bit resolution
  The comparison level (lower) to be compared is set in bits 7 to 0. Write 0 to bits 15 to 8.
- Flush-left format with setting for 12-bit resolution
  The comparison level (lower) to be compared is set in bits 15 to 4. Write 0 to bits 3 to 0.
- Flush-left format with setting for 10-bit resolution
  The comparison level (lower) to be compared is set in bits 15 to 6. Write 0 to bits 5 to 0.
- Flush-left format with setting for 8-bit resolution
  The comparison level (lower) to be compared is set in bits 15 to 8. Write 0 to bits 7 to 0.

(2) When A/D converted value addition/averaging mode is selected

- Flush-right format with setting for 12-bit resolution
  The comparison level (lower) to be compared with the value of A/D conversion of the same channel is set in bits 11 to 0. Write 0 to bits 15 to 12.
- Flush-right format with setting for 10-bit resolution
  The comparison level (lower) to be compared with the value of A/D conversion of the same channel is set in bits 9 to 0. Write 0 to bits 15 to 10.

- Flush-right format with setting for 8-bit resolution
  The comparison level (lower) to be compared with the value of A/D conversion of the same channel is set in bits 7 to 0. Write 0 to bits 15 to 8.
- Flush-left format with setting for 12-bit resolution
  The comparison level (lower) to be compared with the value of A/D conversion of the same channel is set in bits 15 to 4. Write 0 to bits 3 to 0.
- Flush-left format with setting for 10-bit resolution
  The comparison level (lower) to be compared with the value of A/D conversion of the same channel is set in bits 15 to 6. Write 0 to bits 5 to 0.
- Flush-left format with setting for 8-bit resolution
  The comparison level (lower) to be compared with the value of A/D conversion of the same channel is set in bits 15 to 8. Write 0 to bits 7 to 0.

A/D-converted value average mode can be set only when 2- or 4-time conversion is selected in A/D-converted value addition mode.

(3) When A/D converted value averaging mode is selected

- Flush-right format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 12-bit resolution
  The comparison level (lower) to be compared with the value of A/D conversion of the same channel is set in bits 13 to 0. Write 0 to bits 15 and 14.
- Flush-right format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 10-bit resolution
  The comparison level (lower) to be compared with the value of A/D conversion of the same channel is set in bits 11 to 0. Write 0 to bits 15 and 12.
- Flush-right format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 8-bit resolution
  The comparison level (lower) to be compared with the value of A/D conversion of the same channel is set in bits 9 to 0. Write 0 to bits 15 and 10.
- Flush-right format (in A/D-converted value addition mode and when number of conversions is selected for 16 times) with setting of 12-bit resolution
  The comparison level (lower) to be compared with the value of A/D conversion of the same channel is set in bits 15 to 0.
- Flush-left format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 12-bit resolution
  The comparison level (lower) to be compared with the A/D converted value of the same channel is set in bits 15 to 2. Write 0 to bits 1 and 0.
- Flush-left format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 10-bit resolution
  The comparison level (lower) to be compared with the A/D converted value of the same channel is set in bits 15 to 4. Write 0 to bits 3 to 0.
- Flush-left format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 8-bit resolution
  The comparison level (lower) to be compared with the A/D converted value of the same channel is set in bits 15 to 6. Write 0 to bits 5 to 0.
- Flush-left format (in A/D-converted value addition mode and when number of conversions is selected for 16 times) with setting of 12-bit resolution
  The comparison level (lower) to be compared with the A/D converted value of the same channel is set in bits 15 to 0.

When A/D converted value addition mode is selected, the value for comparison with the result of cumulative A/D converted value of the same channel is specified. Conversion 1, 2, 3, 4, or 16 times is selectable. When A/D converted value addition mode is selected and if conversion 1 to 4 is selected, the resolution of the results of conversion will be extended by 2 bits and so set a value for comparison with the results in the ADCMPDR0 register. When conversion 16 times is selected, the resolution of the results of conversion will be extended by 4 bits and so set a value for comparison with the results in the ADCMPDR0 register.
Even when A/D converted value addition mode is selected, set the value for reference based on the setting of the A/D data register format select bit.

50.2.23 A/D Comparison Function Window A Upper Level Setting Register (ADCMPDR1)

Address(es): S12AD.ADCMPDR1 0008 909Eh, S12AD1.ADCMPDR1 0008 919Eh

	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

This is a readable and writable register that specifies the data for reference when the comparison window A function is used. The register specifies the upper level of window A.
A write operation to the ADCMPDR1 register is enabled even during A/D conversion. Rewriting the value of the register during A/D conversion dynamically changes the reference data.
Satisfy the condition of [upper limit level ≥ lower limit level (ADCMPDR1 setting value ≥ ADCMPDR0 setting value)] when setting.
The ADCMPDR1 register is not used when the window function is disabled.

The data formats of the register for each given condition are shown below.

- Setting of the A/D data register format select bit (flush-right or -left format)
- Setting of the A/D conversion resolution setting bit (12, 10, or 8 bits)
- Setting of the A/D converted value addition/averaging function channel select register (when A/D converted value averaging mode is selected or not selected)
- Setting of the A/D converted value addition/averaging time select register (when addition/averaging mode is select and addition count is selected)

Note: When the comparison values are set in a different format from that of the A/D data register y (ADDRy), correct result of comparison cannot be obtained.

(1) When A/D-converted value addition/averaging mode is not selected

- Flush-right format with the setting for 12-bit resolution
  The comparison level (upper) is set in bits 11 to 0. Write 0 to bits 15 to 12.
- Flush-right format with the setting for 10-bit resolution
  The comparison level (upper) is set in bits 9 to 0. Write 0 to bits 15 to 10.
- Flush-right format with the setting for 8-bit resolution
  The comparison level (upper) is set in bits 7 to 0. Write 0 to bits 15 to 8.
- Flush-left format with the setting for 12-bit resolution
  The comparison level (upper) is set in bits 15 to 4. Write 0 to bits 3 to 0.
- Flush-left format with the setting for 10-bit resolution
  The comparison level (upper) is set in bits 15 to 6. Write 0 to bits 5 to 0.
- Flush-left format with the setting for 8-bit resolution
  The comparison level (upper) is set in bits 15 to 8. Write 0 to bits 7 to 0.

(2) When A/D-converted averaging mode is selected

- Flush-right format with the setting for 12-bit resolution
  The comparison level (upper) to be compared with the A/D-converted results of the same channel is set in bits 11 to 0. Write 0 to bits 15 to 12.
- Flush-right format with the setting for 10-bit resolution
  The comparison level (upper) to be compared with the A/D-converted results of the same channel is set in bits 9 to

0. Write 0 to bits 15 to 10.

- Flush-right format with the setting for 8-bit resolution
  The comparison level (upper) to be compared with the A/D-converted results of the same channel is set in bits 7 to 0. Write 0 to bits 15 to 8.
- Flush-left format with the setting for 12-bit resolution
  The comparison level (upper) to be compared with the A/D-converted results of the same channel is set in bits 15 to 4. Write 0 to bits 3 to 0.
- Flush-left format with the setting for 10-bit resolution
  The comparison level (upper) to be compared with the A/D-converted results of the same channel is set in bits 15 to 6. Write 0 to bits 5 to 0.
- Flush-left format with the setting for 8-bit resolution
  The comparison level (upper) to be compared with the A/D-converted results of the same channel is set in bits 15 to 8. Write 0 to bits 7 to 0.

A/D-converted value average mode can be set only when 2- or 4-time conversion is selected in A/D-converted value addition mode.

(3) When A/D converted value averaging mode is selected

- Flush-right format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 12-bit resolution
  The comparison level (upper) to be compared with the A/D converted value of the same channel is set bits 13 to 0. Write 0 to bits 15 to 14.
- Flush-right format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 10-bit resolution
  The comparison level (upper) to be compared with the A/D converted value of the same channel is set bits 11 to 0. Write 0 to bits 15 to 12.
- Flush-right format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 8-bit resolution
  The comparison level (upper) to be compared with the A/D converted value of the same channel is set bits 9 to 0. Write 0 to bits 15 to 10.
- Flush-right format (in A/D-converted value addition mode and when number of conversions is selected for 16 times) with setting of 12-bit resolution
  The comparison level (upper) to be compared with the A/D converted value of the same channel is set bits 15 to 0.
- Flush-left format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 12-bit resolution
  The comparison level (upper) to be compared with the A/D converted value of the same channel is set in bits 15 to 2. Write 0 to bits 1 and 0.
- Flush-left format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 10-bit resolution
  The comparison level (upper) to be compared with the A/D converted value of the same channel is set in bits 15 to 4. Write 0 to bits 3 to 0.
- Flush-left format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 8-bit resolution
  The comparison level (upper) to be compared with the A/D converted value of the same channel is set in bits 15 to 6. Write 0 to bits 5 to 0.
- Flush-left format (in A/D-converted value addition mode and when number of conversions is selected for 16 times) with setting of 12-bit resolution
  The comparison level (upper) to be compared with the A/D converted value of the same channel is set in bits 15 to 0.

When A/D converted value addition mode is selected, the value for comparison with the result of cumulative A/D converted value of the same channel is specified. Conversion 1, 2, 3, 4, or 16 times is selectable. When A/D converted value addition mode is selected and if conversion 1 to 4 is selected, the resolution of the results of conversion will be extended by 2 bits and so set a value for comparison with the results in the ADCMPDR1 register. When conversion 16 times is selected, the resolution of the results of conversion will be extended by 4 bits and so set a value for comparison with the results in the ADCMPDR1 register.
Even when A/D converted value addition mode is selected, set the value for reference based on the setting of the A/D data register format select bit.

50.2.24 A/D Comparison Function Window A Channel Status Register 0 (ADCMPSR0)

(1) S12AD.ADCMPSR0

Address(es): 0008 90A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	CMPST CHA007	CMPST CHA006	CMPST CHA005	CMPST CHA004	CMPST CHA003	CMPST CHA002	CMPST CHA001	CMPST CHA000
Value after 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	CMPSTCHA000	Comparison Window A Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1) These bits indicate the comparison result of the channels (AN000 to AN007) for the window A comparison condition 0: Comparison condition is not satisfied. 1: Comparison condition is satisfied.	R/W
b1	CMPSTCHA001			R/W
b2	CMPSTCHA002			R/W
b3	CMPSTCHA003			R/W
b4	CMPSTCHA004			R/W
b5	CMPSTCHA005			R/W
b6	CMPSTCHA006			R/W
b7	CMPSTCHA007			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This is a register that stores the result of comparison by the comparison window A function.

CMPSTCHA0n Flag (n = 00 to 07) (Comparison Window A Flag)

These are status flags that indicate the result of comparison of the channels for the window A comparison conditions. When the conversion result matches with the comparison condition that is set in the ADCMPLR0.CMPLCHA0n bit (n = 00 to 07) upon completion of A/D conversion, these flags are set to 1. When the ADCMPCR.CMPIE bit is 1, a comparison interrupt request (S12CMPAI) is generated upon the setting of the flag.
The CMPSTCHA000 flag corresponds to AN000 and the CMPSTCHA007 flag corresponds to AN007.
1 cannot be written to the CMPSTCHA0n flag.
[Setting condition]

- When ADCMPCR.CMPAE = 1 and the condition set in the ADCMPLR0.CMPLCHA0n bit is satisfied.

[Clearing condition]

- When 0 is written after reading the state of 1.

(2) S12AD1.ADCMPSR0

Address(es): 0008 91A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CMPST CHA011	CMPST CHA010	CMPST CHA009	CMPST CHA008	CMPST CHA007	CMPST CHA006	CMPST CHA005	CMPST CHA004	CMPST CHA003	CMPST CHA002	CMPST CHA001	CMPST CHA000
Value after 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	CMPSTCHA000	Comparison Window A Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1) These bits indicate the comparison result of the channels (AN100 to AN111) for the window A comparison condition 0: The comparison condition is not satisfied. 1: The comparison condition is satisfied	R/W
b1	CMPSTCHA001			R/W
b2	CMPSTCHA002			R/W
b3	CMPSTCHA003			R/W
b4	CMPSTCHA004			R/W
b5	CMPSTCHA005			R/W
b6	CMPSTCHA006			R/W
b7	CMPSTCHA007			R/W
b8	CMPSTCHA008			R/W
b9	CMPSTCHA009			R/W
b10	CMPSTCHA010			R/W
b11	CMPSTCHA011			R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This is a register that stores the result of comparison of comparison window A.

CMPSTCHA0n Flag (n = 00 to 11) (Comparison Window A Flag)

These are status flags that indicate the result of comparison of the channels (AN100 to AN111) of the comparison condition for window A. When the conversion result matches with the comparison condition specified in the ADCMPLR0.CMPLCHA0n bit (n = 00 to 11) upon completion of A/D conversion, this flag is set to 1. When the ADCMPCR.CMPIE bit is 1, a comparison interrupt request (S12CMPAI1) is generated upon the setting of the flag. The CMPSTCHA000 flag corresponds to AN100 and the CMPSTCHA011 flag corresponds to AN111.
1 cannot be written to the CMPSTCHA0n flag.
[Setting condition]

- When ADCMPCR.CMPAE = 1, the condition set in the ADCMPLR0.CMPLCHA0n bit is satisfied.

[Clearing condition]

- When 0 is written after reading the state of 1.

50.2.25 A/D Comparison Function Window A Extended Input Channel Status Register (ADCMPSER)

Address(es): S12AD1.ADCMPSER 0008 91A4h

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

Bit	Symbol	Bit Name	Description	R/W
b0	CMPFTS	Comparison Window A Temperature Sensor Output Comparison Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1) This bit indicates the comparison result of the temperature sensor output 0: The comparison condition is not satisfied. 1: The comparison condition is satisfied.	R/W
b1	CMPFOC	Comparison Window A Internal Reference Voltage Comparison Flag	When window A operation is enabled (ADCMPCR.CMPAE = 1) This bit indicates the comparison result of the internal reference voltage 0: The comparison condition is not satisfied. 1: The comparison condition is satisfied.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This is a register that stores the result of the comparison of the comparison window A function.

CMPFTS Flag (Comparison Window A Temperature Sensor Output Comparison Flag)

This is a status flag that indicates the result of comparison of the temperature sensor output. When the conversion result matches with the comparison condition specified in the ADCMPLER.CMPLTS bit upon completion of A/D conversion, this flag is set to 1. When the ADCMPCR.CMPIE bit is 1, a comparison interrupt request (S12CMPAI1) is generated upon the setting of the flag.

1 cannot be written to the CMPFTS flag.

[Setting condition]

- When ADCMPCR.CMPAE = 1, the condition set in the ADCMPLER.CMPLTS bit is satisfied.

[Clearing condition]

- When 0 is written after reading the state of 1.

CMPFOC Flag (Comparison Window A Internal Reference Voltage Comparison Flag)

This is a status flag that indicates the result of comparison of the internal reference voltage. When the conversion result matches with the comparison condition specified in the ADCMPLER.CMPLOC bit upon completion of A/D conversion, this flag is set to 1. When the ADCMPCR.CMPIE bit is 1, a comparison interrupt request (S12CMPAI1) is generated upon the setting of the flag.

1 cannot be written to the CMPFOC flag.

[Setting condition]

- When ADCMPCR.CMPAE = 1, the condition set in the ADCMPLER.CMPLOC bit is satisfied.

[Clearing condition]

- When 0 is written after reading the state of 1.

50.2.26 A/D Comparison Function Window A/B Status Monitoring Register (ADWINMON)

Address(es): S12AD.ADWINMON 0008 908Ch, S12AD1.ADWINMON 0008 918Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	MONCMPB	MONCMPA	—	—	—	MONCOMB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MONCOMB	Combination Result Monitoring	When window A operation is enabled (ADCMPCR.CMPAE = 1) These bits indicate the comparison result of the internal reference voltage 0: The comparison condition is not satisfied. 1: The comparison condition is satisfied.	R
b3 to b1	—	Reserved	These bits are read as 0.	R
b4	MONCMPA	Comparison Result Monitor A	0: The comparison condition for window A is not satisfied. 1: The comparison condition for window A is satisfied.	R
b5	MONCMPB	Comparison Result Monitor B	0: The comparison condition for window B is not satisfied. 1: The comparison condition for window B is satisfied.	R
b7, b6	—	Reserved	These bits are read as 0.	R

This register can monitor the results of comparison and the combined result.

MONCOMB Bit (Combination Result Monitoring)

This is a dedicated bit for reading the combined result of comparison condition A and B that were set in the ADCMPCR.CMPAB[1:0] bits as a combined condition.

[Setting condition]

- When ADCMPCR.CMPAE = 1 and ADCMPCR.CMPBE = 1, the condition set in the ADCMPCR.CMPAB[1:0] bits is satisfied.

[Clearing condition]

- No match with the combined condition set in the ADCMPCR.CMPAB[1:0] bits
- When ADCMPCR.CMPAE = 0 or ADCMPCR.CMPBE = 0

MONCMPA Bit (Comparison Result Monitor A)

This is a dedicated bit for reading 1 when the condition set in the ADCMPLR0 or ADCMPLER matches with the A/D converted value of the channel targeted for window A and for reading 0 when such condition is not satisfied.

[Setting condition]

- When ADCMPCR.CMPAE = 1, the condition set in the ADCMPLR0.CMPLCHA0n bit is satisfied.

[Clearing condition]

- When ADCMPCR.CMPAE = 1, the condition set in the ADCMPLR0.CMPLCHA0n bit is not satisfied.
- When ADCMPCR.CMPAE = 0, (ADCMPCR.CMPAE is automatically cleared to 0)

MONCMPB Bit (Comparison Result Monitor B)

This is a dedicated bit for reading 1 when the condition set in the ADCMPBNSR.CMPLB matches with the A/D converted value of the channel targeted for window B and for reading 0 when such condition is not satisfied.

[Setting condition]

- When ADCMPCR.CMPBE = 1, the condition set in the ADCMPBNSR.CMPLB bit is satisfied.

[Clearing condition]

- When ADCMPCR.CMPBE = 1, the condition set in the ADCMPBNSR.CMPLB bit is not satisfied.
- When ADCMPCR.CMPBE = 0, (ADCMPCR.CMPBE is automatically cleared to 0)

50.2.27 A/D Comparison Function Window B Channel Select Register (ADCMPBNSR)

(1) S12AD.ADCMPBNSR

Address(es): 0008 90A6h

	b7	b6	b5	b4	b3	b2	b1	b0
	CMPLB	—	CMPCHB[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	CMPCHB[5:0]	Comparison Window B Channel Select	Select the channel that compares with the condition of comparison window B b5 b0 0 0 0 0 0 0: AN000 0 0 0 0 0 1: AN001 0 0 0 0 1 0: AN002 : : 0 0 0 1 1 0: AN006 0 0 0 1 1 1: AN007 Settings other than above are prohibited.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	CMPLB	Comparison Window B Comparison Condition Setting	When the window function is disabled (the ADCMPCR.WCMPE bit is 0) 0: ADWINLLB register value > A/D converted value 1: ADWINLLB register value < A/D converted value When the window function is enabled (the ADCMPCR.WCMPE bit is 1) 0: A/D converted value < ADWINLLB register value or ADWINULB register value < A/D converted value 1: ADWINLLB register value < A/D converted value < ADWINULB register value	R/W

This register is used to specify the comparison window B function.

CMPCHB[5:0] Bits (Comparison Window B Channel Select)

These bits select the channel to be compared with the comparison window B from AN000 to AN007.
When the number of the A/D conversion channel selected in the ADANSAy.ANSAyn bit (y = 0, n = 00 to 07) and ADANSBy.ANSByn bit, the comparison window B function is enabled.
The CMPCHB[5:0] bits should be set while the ADCSR.ADST bit is 0.

CMPLB Bit (Comparison Window B Comparison Condition Setting)

This bit specifies the condition for comparison of the channels for window B. When the comparison result of each analog input matches the pre-set condition, the ADCMPBSR.CMPSTB flag is set to 1, a comparison interrupt request (S12CMPBI) is generated. Figure 50.5 shows the conditions for comparison.

(2) S12AD1.ADCMPBNSR

Address(es): 0008 91A6h

	b7	b6	b5	b4	b3	b2	b1	b0
	CMPLB	—	CMPCHB[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	CMPCHB[5:0]	Comparison Window B Channel Select	Select the channel that compares with the condition of comparison window B b5 b0 0 0 0 0 0 0: AN100 0 0 0 0 0 1: AN101 0 0 0 0 1 0: AN102 : : 0 0 0 1 1 0: AN106 0 0 0 1 1 1: AN107 : : 0 0 1 0 1 1: AN111 1 0 0 0 0 0: Temperature sensor 1 0 0 0 0 1: Internal reference voltage Settings other than above are prohibited.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	CMPLB	Comparison Window B Comparison Condition Setting	When window function is disabled (the ADCMPCR.WCMPE bit is 0) 0: ADWINLLB register value > A/D converted value 1: ADWINLLB register value < A/D converted value When window function is enabled (the ADCMPCR.WCMPE bit is 1) 0: A/D converted value < ADWINLLB register value or ADWINULB register value < A/D converted value 1: ADWINLLB register value < A/D converted value < ADWINULB register value	R/W

This register is used to specify the compare window B function.

CMPCHB[5:0] Bits (Comparison Window B Channel Select)

These bits select the channels to be compared under the comparison window B condition from AN100 to AN111, temperature sensor, internal reference voltage.

When the number of the A/D conversion channel selected in the ADANSAy.ANSAyn bit (y = 0, n = 00 to 11) and ADANSBy.ANSByn bit is specified, the comparison window B function is enabled.

The CMPCHB[5:0] bits should be set while the ADCSR.ADST bit is 0.

CMPLB Bit (Comparison Window B Comparison Condition Setting)

This bit specifies the condition for comparison of the channels for window B. When the comparison result of each analog input matches the pre-set condition, the ADCMPBSR.CMPSTB flag is set to 1, a comparison interrupt request (S12CMPBI1) is generated. Figure 50.5 shows the conditions for comparison.

(1) Comparison condition when the window function is disabled

CMPLB = 0

ADWINLLB register value ≤ A/D converted value	Not matched
ADWINLLB register value > A/D converted value	Matched

CMPLB = 1

ADWINLLB register value < A/D converted value	Matched
ADWINLLB register value ≥ A/D converted value	Not matched

(2) Comparison condition when the window function is enabled

CMPLB = 0

AD converted value > ADWINULB register value	Matched
ADWINLLB register value ≤ AD converted value ≤ ADWINULB register value	Not matched
AD converted value < ADWINLLB register value	Matched

CMPLB = 1

AD converted value ≥ ADWINULB register value	Not matched
ADWINLLB register value < A/D converted value < ADWINULB register value	Matched
AD converted value ≤ ADWINLLB register value	Not matched

Figure 50.5 Details of Comparison Condition: Comparison Function Window B

50.2.28 A/D Comparison Function Window B Lower Level Setting Register (ADWINLLB)

Address(es): S12AD.ADWINLLB 0008 90A8h, S12AD1.ADWINLLB 0008 91A8h

	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

This is a readable and writable register that specifies reference data when comparison window B is in use. The register specifies the lower level of window B.
A write operation to the ADCMPDR1 register is enabled even during A/D conversion. Rewriting the value of the register during A/D conversion dynamically changes the reference data.
Satisfy the condition of [upper limit level ≥ lower limit level (ADWINULB setting value ≥ ADWINLLB setting value)] when setting.

The data formats of the register for each given condition are shown below.

- Setting of the A/D data register format select bit (flush-right or -left format)
- Setting of the A/D conversion resolution setting bit (12, 10, or 8 bits)
- Setting of the A/D converted value addition/averaging function channel select register (A/D converted value averaging mode: selected or not selected)
- Setting of the A/D converted value addition/averaging count select register (addition/averaging mode selected, addition count selected)

Note: When the comparison values are set in a different format from that of the A/D data register y (ADDRy), correct result of comparison cannot be obtained.

(1) When A/D-converted value addition/averaging mode is not selected

- Flush-right format with the setting for 12-bit resolution
  The comparison level (lower) is set in bits 11 to 0. Write 0 to bits 15 to 12.
- Flush-right format with the setting for 10-bit resolution
  The comparison level (lower) is set in bits 9 to 0. Write 0 to bits 15 to 10.
- Flush-right format with the setting for 8-bit resolution
  The comparison level (lower) is set in bits 7 to 0. Write 0 to bits 15 to 8.
- Flush-left format with the setting for 12-bit resolution
  The comparison level (lower) is set in bits 15 to 4. Write 0 to bits 3 to 0.
- Flush-left format with the setting for 10-bit resolution
  The comparison level (lower) is set in bits 15 to 6. Write 0 to bits 5 to 0.
- Flush-left format with the setting for 8-bit resolution
  The comparison level (lower) is set in bits 15 to 8. Write 0 to bits 7 to 0.

(2) When A/D-converted averaging mode is selected

- Flush-right format with the setting for 12-bit resolution
  The comparison level (lower) to be compared with the A/D-converted results of the same channel is set in bits 11 to 0. Write 0 to bits 15 to 12.
- Flush-right format with the setting for 10-bit resolution
  The comparison level (lower) to be compared with the A/D-converted results of the same channel is set in bits 9 to 0. Write 0 to bits 15 to 10.

- Flush-right format with the setting for 8-bit resolution
  The comparison level (lower) to be compared with the A/D-converted results of the same channel is set in bits 7 to 0. Write 0 to bits 15 to 8.
- Flush-left format with the setting for 12-bit resolution
  The comparison level (lower) to be compared with the A/D-converted results of the same channel is set in bits 15 to 4. Write 0 to bits 3 to 0.
- Flush-left format with the setting for 10-bit resolution
  The comparison level (lower) to be compared with the A/D-converted results of the same channel is set in bits 15 to 6. Write 0 to bits 5 to 0.
- Flush-left format with the setting for 8-bit resolution
  The comparison level (lower) to be compared with the A/D-converted results of the same channel is set in bits 15 to 8. Write 0 to bits 7 to 0.

A/D-converted value average mode can be set only when 2- or 4-time conversion is selected in A/D-converted value addition mode.

(3) When A/D-converted addition mode is selected

- Flush-right format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 12-bit resolution
  The comparison level (lower) to be compared with the A/D-converted results of the same channel is set in bits 13 to 0. Write 0 to bits 15 and 14.
- Flush-right format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 10-bit resolution
  The comparison level (lower) to be compared with the A/D-converted results of the same channel is set in bits 11 to 0. Write 0 to bits 15 to 12.
- Flush-right format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 8-bit resolution
  The comparison level (lower) to be compared with the A/D-converted results of the same channel is set in bits 9 to 0. Write 0 to bits 15 and 10.
- Flush-right format (in A/D-converted value addition mode and when number of conversions is selected for 16 times) with setting of 12-bit resolution
  The comparison level (lower) to be compared with the A/D-converted results of the same channel is set in bits 15 to 0.
- Flush-left format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 12-bit resolution
  The comparison level (lower) to be compared with the A/D-converted results of the same channel is set in bits 15 to 2. Write 0 to bits 1 and 0.
- Flush-left format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 10-bit resolution
  The comparison level (lower) to be compared with the A/D-converted results of the same channel is set in bits 15 to 4. Write 0 to bits 3 to 0.
- Flush-left format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 8-bit resolution
  The comparison level (lower) to be compared with the A/D-converted results of the same channel is set in bits 15 to 6. Write 0 to bits 5 to 0.
- Flush-left format (in A/D-converted value addition mode and when number of conversions is selected for 16 times) with setting of 12-bit resolution
  The comparison level (lower) to be compared with the A/D-converted results of the same channel is set in bits 15 to 0.

When A/D converted value addition mode is selected, the value for comparison with the result of cumulative A/D converted value of the same channel is specified. Conversion 1, 2, 3, 4, or 16 times is selectable. When A/D converted value addition mode is selected and if conversion 1 to 4 is selected, the resolution of the results of conversion will be extended by 2 bits and so set a value for comparison with the results in the ADWINLLB register. When conversion 16 times is selected, the resolution of the results of conversion will be extended by 4 bits and so set a value for comparison with the results in the ADWINLLB register.
Even when A/D converted value addition mode is selected, set the value for reference based on the setting of the A/D data register format select bit.

50.2.29 A/D Comparison Function Window B Upper Level Setting Register (ADWINULB)

Address(es): S12AD.ADWINULB 0008 90AAh, S12AD1.ADWINULB 0008 91AAh

	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

This is a readable and writable register that specifies the data for reference when the comparison window B function is used. The register specifies the upper level of window B.
A write operation to the register is enabled even during A/D conversion. Rewriting the value of the register during A/D conversion dynamically changes the reference data.
Satisfy the condition of [upper limit level ≥ lower limit level (ADWINULB setting value ≥ ADWINLLB setting value)] when setting.
The register is not used when the window function is disabled.

The format of the register differs depending on the conditions below.

- Setting of the A/D data register format select bit (flush-right or -left format)
- Setting of the A/D conversion resolution setting bit (12, 10, or 8 bits)
- Setting of the A/D-converted value addition/averaging function channel select register (when A/D converted value averaging mode is selected or not selected)
- Setting of the A/D converted value addition/averaging count select register (when addition/averaging mode is selected and addition count is selected)

Note: When the comparison values are set in a different format from that of the A/D data register y (ADDRy), correct result of comparison cannot be obtained.

(1) When A/D-converted value addition/averaging mode is not selected

- Flush-right format with the setting for 12-bit resolution
  The comparison level (upper) is set in bits 11 to 0. Write 0 to bits 15 to b12.
- Flush-right format with the setting for 10-bit resolution
  The comparison level (upper) is set in bits 9 to 0. Write 0 to bits 15 to b10.
- Flush-right format with the setting for 8-bit resolution
  The comparison level (upper) is set in bits 7 to 0. Write 0 to bits 15 to b8.
- Flush-left format with the setting for 12-bit resolution
  The comparison level (upper) is set in bits 15 to 4. Write 0 to bits 3 to 0.
- Flush-left format with the setting for 10-bit resolution
  The comparison level (upper) is set in bits 15 to 6. Write 0 to bits 5 to 0.
- Flush-left format with the setting for 8-bit resolution
  The comparison level (upper) is set in bits 15 to 8. Write 0 to bits 7 to 0.

(2) When A/D-converted averaging mode is selected

- Flush-right format with the setting for 12-bit resolution
  The comparison level (upper) to be compared with the A/D-converted results of the same channel is set in bits 11 to 0. Write 0 to bits 15 to 12.
- Flush-right format with the setting for 10-bit resolution
  The comparison level (upper) to be compared with the A/D-converted results of the same channel is set in bits 9 to

0. Write 0 to bits 15 to 10.

- Flush-right format with the setting for 8-bit resolution
  The comparison level (upper) to be compared with the A/D-converted results of the same channel is set in bits 7 to 0. Write 0 to bits 15 to 8.
- Flush-left format with the setting for 12-bit resolution
  The comparison level (upper) to be compared with the A/D-converted results of the same channel is set in bits 15 to 4. Write 0 to bits 3 to 0.
- Flush-left format with the setting for 10-bit resolution
  The comparison level (upper) to be compared with the A/D-converted results of the same channel is set in bits 15 to 6. Write 0 to bits 5 to 0.
- Flush-left format with the setting for 8-bit resolution
  The comparison level (upper) to be compared with the A/D-converted results of the same channel is set in bits 15 to 8. Write 0 to bits 7 to 0.

A/D-converted value average mode can be set only when 2- or 4-time conversion is selected in A/D-converted value addition mode.

(3) When A/D-converted addition mode is selected

- Flush-right format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 12-bit resolution
  The comparison level (upper) to be compared with the A/D-converted results of the same channel is set in bits 13 to 0. Write 0 to bits 15 and 14.
- Flush-right format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 10-bit resolution
  The comparison level (upper) to be compared with the A/D-converted results of the same channel is set in bits 11 to 0. Write 0 to bits 15 to 12.
- Flush-right format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 8-bit resolution
  The comparison level (upper) to be compared with the A/D-converted results of the same channel is set in bits 9 to 0. Write 0 to bits 15 to 10.
- Flush-right format (in A/D-converted value addition mode and when number of conversions is selected for 16 times) with setting of 12-bit resolution
  The comparison level (upper) to be compared with the A/D-converted results of the same channel is set in bits 13 to 0.
- Flush-left format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 12-bit resolution
  The comparison level (upper) to be compared with the A/D-converted results of the same channel is set in bits 15 to 2. Write 0 to bits 1 and 0.
- Flush-left format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 10-bit resolution
  The comparison level (upper) to be compared with the A/D-converted results of the same channel is set in bits 15 to 4. Write 0 to bits 3 to 0.
- Flush-left format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times) with setting of 8-bit resolution
  The comparison level (upper) to be compared with the A/D-converted results of the same channel is set in bits 15 to 6. Write 0 to bits 5 to 0.
- Flush-left format (in A/D-converted value addition mode and when number of conversions is selected for 16 times) with setting of 12-bit resolution
  The comparison level (upper) to be compared with the A/D-converted results of the same channel is set in bits 15 to 0.

When A/D converted value addition mode is selected, the value for comparison with the result of cumulative A/D converted value of the same channel is specified. Conversion 1, 2, 3, 4, or 16 times is selectable. When A/D converted value addition mode is selected and if conversion 1 to 4 is selected, the resolution of the results of conversion will be extended by 2 bits and so set a value for comparison with the results in the ADWINLLB register. When conversion 16 times is selected, the resolution of the results of conversion will be extended by 4 bits and so set a value for comparison with the results in the ADWINLLB register.
Even when A/D converted value addition mode is selected, set the value for reference based on the setting of the A/D data register format select bit.

50.2.30 A/D Comparison Function Window B Channel Status Register (ADCMPBSR)

(1) S12AD.ADCMPBSR

Address(es): 0008 90ACh

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

Bit	Symbol	Bit Name	Description	R/W
b0	CMPSTB	Comparison Window B Flag	0: The comparison condition is not satisfied. 1: The comparison condition is satisfied.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is used to store results of comparison of the comparison window B function.

CMPSTB Flag (Comparison Window B Flag)

This is a status flag that indicates the result of comparison of the channels for window B comparison conditions (AN000 to AN007.
When the conversion result matches with the comparison condition that is set in the ADCMPBNSR.CMPCHB[5:0] bits upon completion of A/D conversion, this flag is set to 1. When the ADCMPCR.CMPIE bit is 1, a comparison interrupt request (S12CMPBI) is generated upon the setting of the flag.
1 cannot be written to the CMPSTB flag.
[Setting condition]

- When ADCMPCR.CMPBE = 1 and the condition set in the ADCMPBNSR.CMPLB bit is satisfied.

[Clearing condition]

- When 0 is written after reading the state of 1.

(2) S12AD1.ADCMPBSR

Address(es): 0008 91ACh

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

Bit	Symbol	Bit Name	Description	R/W
b0	CMPSTB	Comparison Window B Flag	0: The comparison condition is not satisfied. 1: The comparison condition is satisfied.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This is a register that stores the result of comparison by comparison window B function.

CMPSTB Flag (Comparison Window B Flag)

This is a status flag that indicates the result of comparison of the channels for window B comparison conditions (AN100 to AN111, temperature sensor, internal reference voltage).
When the comparison condition that is set in the ADCMPBNSR.CMPCHB[5:0] bits is a match upon completion of A/D conversion, this flag is set to 1. When the ADCMPCR.CMPIE bit is 1, a comparison interrupt request (S12CMPBI1) is generated upon the setting of the flag.
1 cannot be written to the CMPSTB flag.
[Setting condition]

- When ADCMPCR.CMPBE = 1 and the condition set in the ADCMPBNSR.CMPLB bit is satisfied.

[Clearing condition]

- When 0 is written after reading the state of 1.

50.2.31 A/D Conversion Time Setting Register (ADSAM)

Address(es): S12AD.ADSAM 0008 906Eh, S12AD1.ADSAM 0008 916Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	SAM	—	—	—	—	—
Value after 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
b5	SAM	Conversion Time Setting	0: Sets conversion time for high-speed. 1: Sets conversion time for middle-speed.	R/W
b15 to b6	—	Reserved	These bits are read as 0. The write value should be 0.	R

This register specifies conversion time for unit 0 (high-speed) and unit 1 (middle-speed).
Set the PSW.I bit to 0 (interrupt disabled) and the ADSAMPR.PRO[1:0] bits to 11 (writing enabled) before rewriting the register. Furthermore, disable writing to the register by setting the ADSAMPR.PRO[1:0] bits to 10 immediately after rewriting the register.

SAM Bit (Conversion Time Setting)

Set this bit to 0 for unit 0 which is a high-speed A/D converter. Setting this bit to 0 sets conversion time 13 states for 12-bit resolution, 11 states for 10-bit resolution, or 9 states for 8-bit resolution.
Set this bit to 1 for unit 1 which is a middle-speed A/D converter. Setting this bit to 1 sets conversion time 15 states for 12-bit resolution, 13 states for 10-bit resolution, or 11 states for 8-bit resolution.
Set the bit while the ADCSR.ADST bit is 0.
Note that the electrical characteristics cannot be satisfied when any setting other than the above is made to the respective units.

50.2.32 A/D Conversion Time Setting Protection Release Register (ADSAMPR)

Address(es): S12AD.ADSAMPR 0008 9063h, S12AD1.ADSAMPR 0008 9163h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PRO[1:0]	A/D Conversion Time Setting Register Protection	b1 b0 0 0: Writing to or reading from the A/D conversion time setting register is disabled (initial value. The value is read as 00b). 1 0: Writing to or reading from the A/D conversion time setting register is disabled (The value is read as 00b). 1 1: Writing to or reading from the A/D conversion time setting register is enabled (The value is read as 01b). Settings other than above are prohibited.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

This register enables accesses to the A/D conversion time setting register (ADSAM).

Set the PRO[1:0] bits to 10b (writing disabled) after setting the A/D conversion time setting register (ADSAM).
The PRO[1:0] bits are read as 00b while writing to the bits is disabled, and 01b while writing to the bits is enabled.

50.3 Operation

50.3.1 Scanning Operation

In scanning, A/D conversion is performed sequentially on the analog inputs of the specified channels.
A scan conversion is performed in three operating modes: single scan mode, continuous scan mode, and group scan mode. In single scan mode, one or more specified channels are scanned once. In continuous scan mode, one or more specified channels are scanned repeatedly until the ADCSR.ADST bit is cleared to 0 from 1 by software. In group scan mode, the selected channels of groups A, B, and C are scanned once after starting to be scanned according to the respective synchronous trigger.
In single scan mode and continuous scan mode, A/D conversion is performed for ANn channels selected by the ADANSA0 register, starting from the channel with the smallest number n. In group scan mode, A/D conversion is performed for ANn channels of groups A, B, and C selected by the ADANSA0 register, ADANSB0 register, and ADANSC0 register, respectively, starting from the channel with the smallest number n.
When self-diagnosis is selected, it is executed once at the beginning of each scan and one of the three voltages internally generated in the 12-bit A/D converter is converted.
The temperature sensor output and internal reference voltage can be selected with analog input of the channels, and A/D conversion is performed in the order of the analog input of the channels, temperature sensor output, and internal reference voltage. When the extended analog is selected, perform A/D conversion in single scan mode or sequence scan mode.
Double trigger mode is to be used with single scan mode or group scan mode. With double trigger mode being enabled, A/D conversion data of a channel selected by the ADCSR.DBLANS[4:0] bits is duplicated only if the conversion is started by the synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits. In group scan mode, the double trigger function can be used only for group A.
Extended double trigger mode indicates a state when the following synchronous trigger (two synchronous trigger sources enabled) is selected by the TRSA[5:0] bits in the A/D conversion start trigger select register (ADSTRGR) in double trigger mode.

- TRG4AN or TRG4BN (the ADSTRGR.TRSA[5:0] bits are set to 001011b)
- TRG7AN or TRG7BN (the ADSTRGR.TRSA[5:0] bits are set to 001111b)

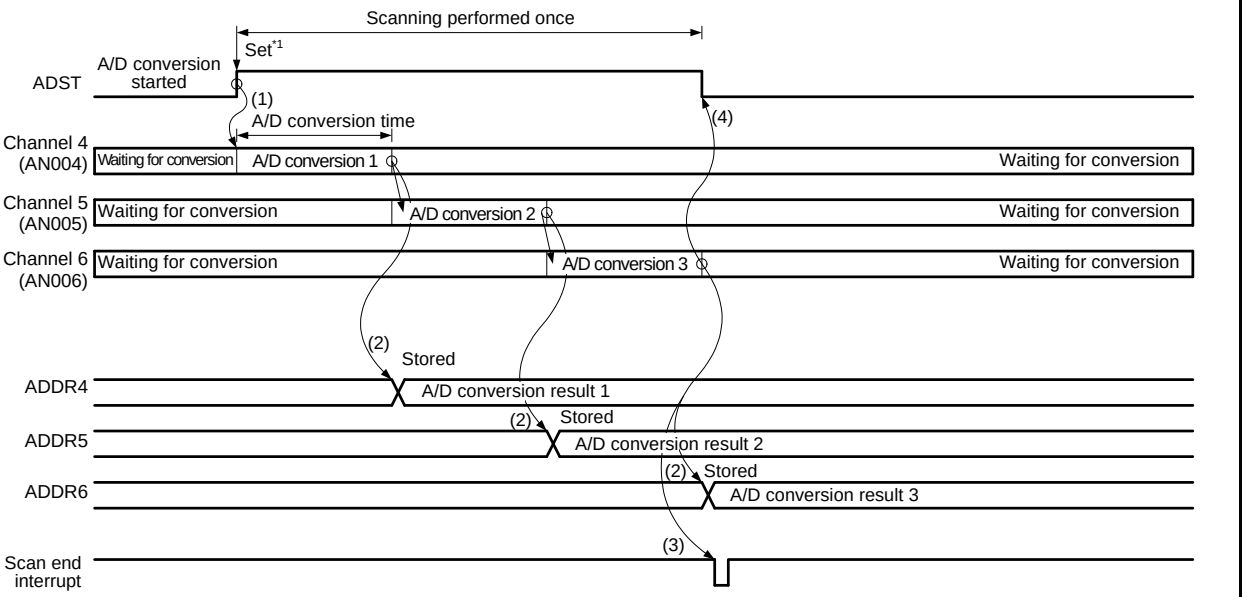
In extended double trigger mode, in addition to normal operations in double trigger mode, A/D conversion data is stored in A/D data duplication register A (ADDBLDRA) or A/D data duplication register B (ADDBLDRB) depending on the trigger type. If two types of triggers have occurred simultaneously in this mode, A/D conversion data is not sorted by the trigger sources and is stored in data duplication register B (ADDBLDRB).
Note that if a new trigger is input during A/D conversion caused by another trigger, the new trigger is ignored.

50.3.2 Single Scan Mode

50.3.2.1 Basic Operation

In basic operation of single scan mode, A/D conversion is performed once on the analog input of the specified channels as below.

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, or synchronous or asynchronous trigger input, A/D conversion is performed for ANn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.
(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(3) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).
(4) The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a wait state.



Note 1. ↓ indicates the instruction is executed by software.

Figure 50.6 Example of Operation in Single Scan Mode (Basic Operation: AN004, AN005, AN006 Selected)

50.3.2.2 Channel Selection and Self-Diagnosis

When channels and self-diagnosis are selected, A/D conversion is performed once for the reference voltage supplied to the 12-bit A/D converter as below. After that, A/D conversion is performed only once on the analog input of the selected channels.

(1) A/D conversion for self-diagnosis is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, or synchronous or asynchronous trigger input.
(2) When A/D conversion for self-diagnosis is completed, A/D conversion result is stored into the A/D self-diagnosis data register (ADRD), and A/D conversion is performed for ANn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.
(3) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(4) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).
(5) The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a wait state.

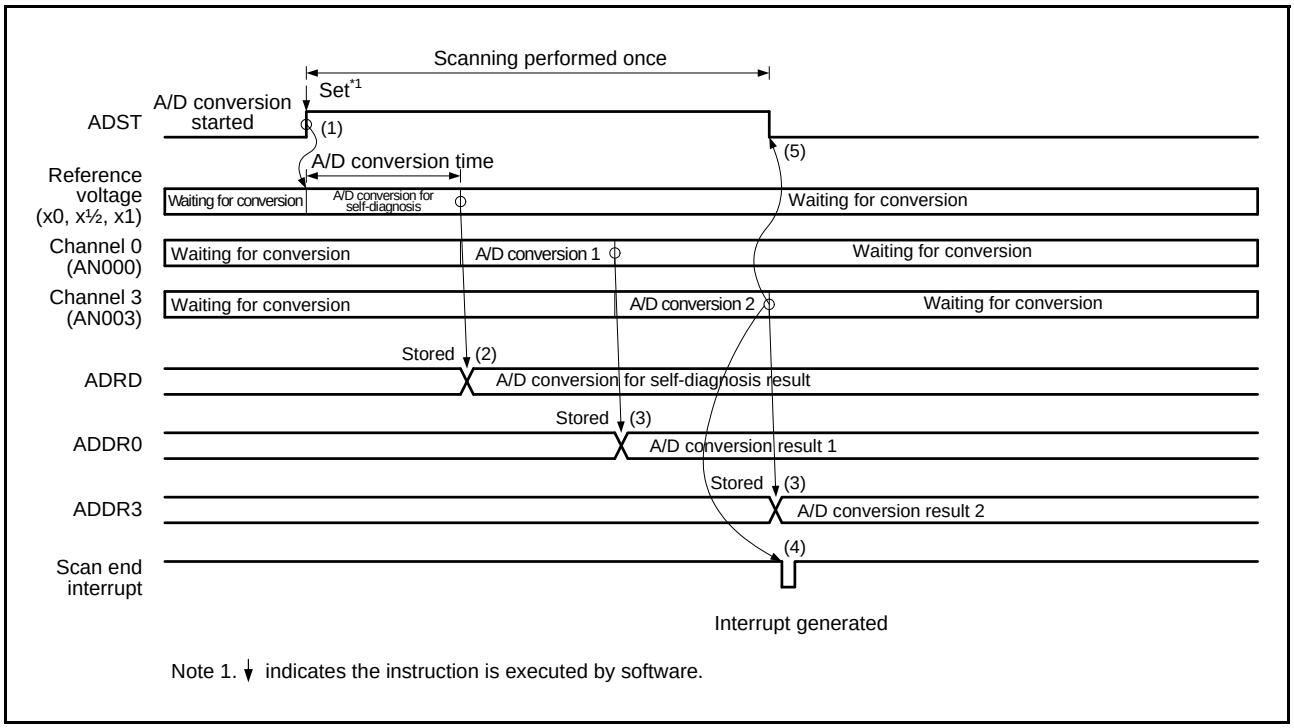


Figure 50.7 Example of Operation in Single Scan Mode (Basic Operation: AN000, AN003 Selected + Self-Diagnosis)

50.3.2.3 A/D Conversion With Temperature Sensor Output/Internal Reference Voltage Selected

When temperature sensor output or internal reference voltage is selected with channels, A/D conversion of analog input of the channels is performed as described below. After that, A/D conversion is performed only once for the temperature sensor output or internal reference voltage. When temperature sensor output and internal reference voltage are both selected, A/D conversion is performed for the temperature sensor output first and then internal reference voltage. Deselecting channels and selecting only one of temperature sensor output and internal reference voltage is possible.

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, or synchronous or asynchronous trigger input, and A/D conversion is performed for ANn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.
(2) Each time A/D conversion of a single channel is completed, the A/D converted value is stored in the corresponding A/D data register (ADDRy), and then A/D conversion for temperature sensor output is started.
(3) When A/D conversion of the temperature sensor output is completed, the A/D converted value is stored in the corresponding A/D temperature sensor data register (ADTSDR), and then A/D conversion for internal reference voltage is started.
(4) After completion of A/D conversion of the internal reference voltage, the A/D converted value is stored in the corresponding A/D internal reference voltage data register (ADOCDR), and if the ADCSR.ADIE bit is set to 1 (when the generation of an interrupt after scanning is enabled), a scan end interrupt request is generated.
(5) The ADCSR.ADST bit holds 1 (A/D conversion start) during A/D conversion, and the bit is automatically cleared upon completion of the A/D conversion. The 12-bit A/D converter enters into a wait state.

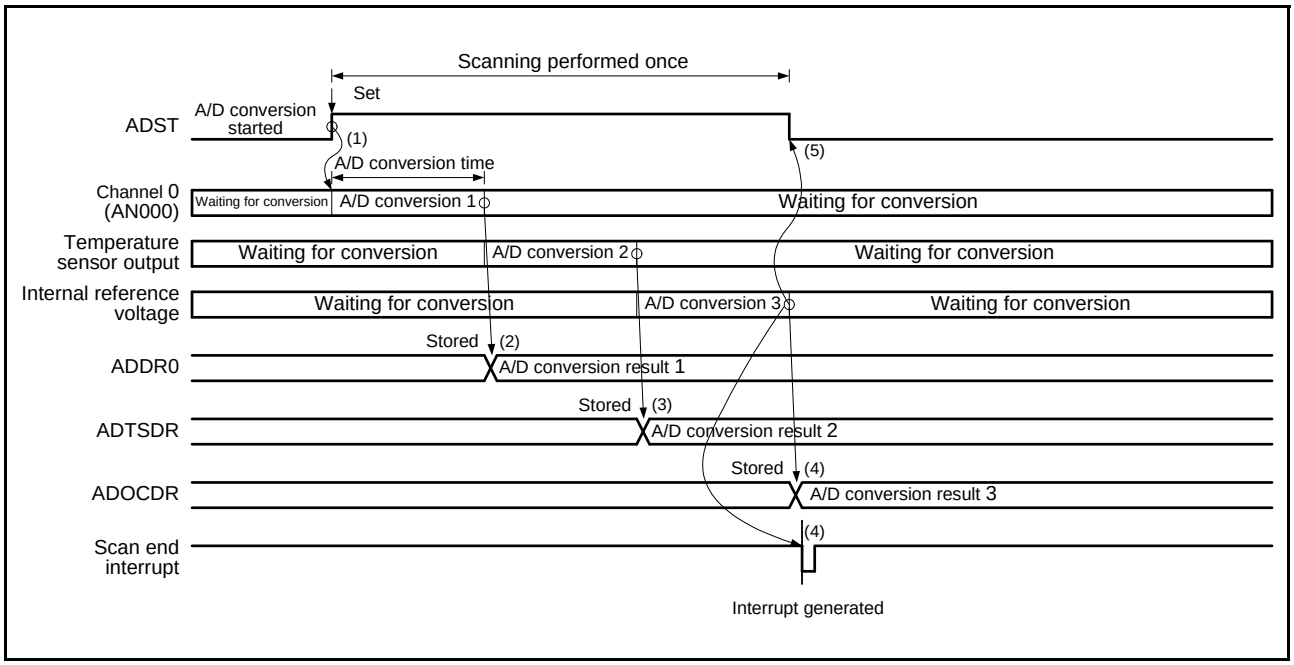


Figure 50.8 Example of Operation in Single Scan Mode (Basic Operation: AN000, Temperature Sensor Output, and Internal Reference Voltage Selected)

50.3.2.4 A/D Conversion in Double Trigger Mode

In single scan mode with double trigger mode, single scan operation started by synchronous trigger is performed twice as below.

Deselect self-diagnosis and set the temperature sensor A/D conversion select bit (ADEXICR.TSSA, ADEXICR.TSSB) and internal reference voltage A/D conversion select bit (ADEXICR.OCSA, ADEXICR.OCSB) to 0.

Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1. When the ADCSR.DBLE bit is set to 1, channel selection using the ADANSA0 register is invalid. In double trigger mode, synchronous triggers should be selected using the ADSTRGR.TRSA[5:0] bits, the ADCSR.EXTRG bit should be set to 0, and the ADCSR.TRGE bit should be set to 1. Software trigger should not be used.

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by synchronous trigger input, A/D conversion is started on the single channel selected by the ADCSR.DBLANS[4:0] bits.
(2) When A/D conversion is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(3) The ADCSR.ADST bit is automatically cleared to 0 and the 12-bit A/D converter enters a wait state. Here, a scan end interrupt request is not generated irrespective of the ADCSR.ADIE bit setting (interrupt generation upon scanning completion enabled).
(4) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by the second trigger input, A/D conversion is started on the single channel selected by the ADCSR.DBLANS[4:0] bits.
(5) When A/D conversion is completed, the A/D conversion result is stored into the A/D data duplication register (ADDBLDR), which is exclusively used in double trigger mode.
(6) If the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled), a scan end interrupt request is generated.
(7) The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion is completed. Then the 12-bit A/D converter enters a wait state.

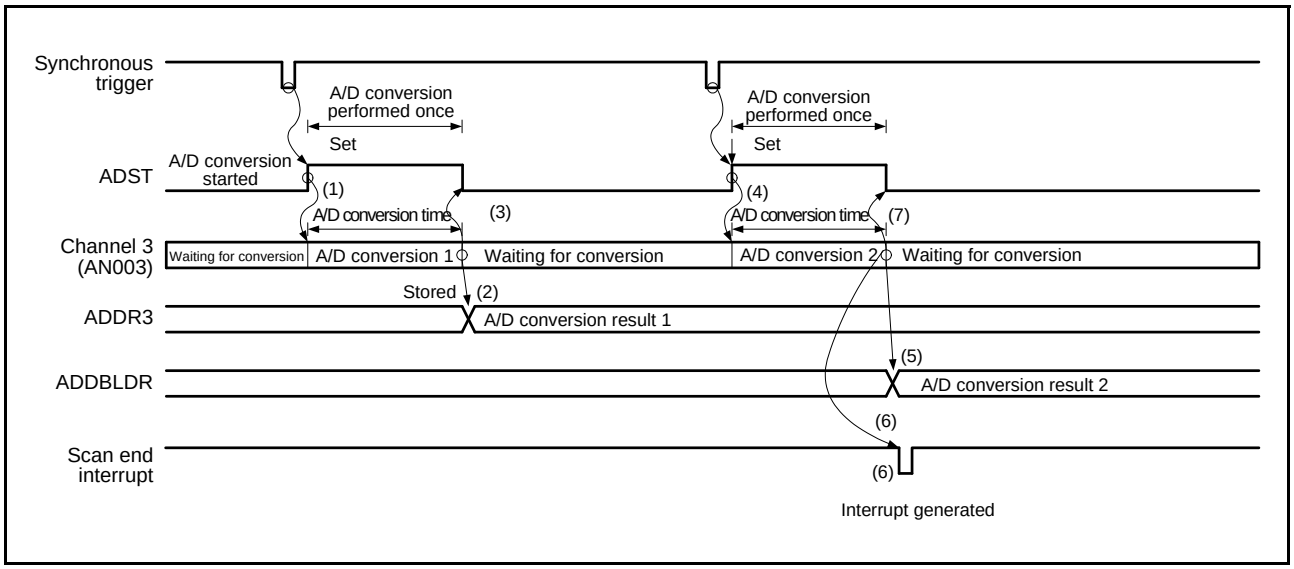


Figure 50.9 Example of Operation in Single Scan Mode (Double Trigger Mode Selected; AN003 Duplicated)

50.3.2.5 A/D Conversion in Extended Double Trigger Mode

When the double-trigger mode is selected in single-scanning mode, with TRG4AN or TRG4BN, TRG7AN or TRG7BN selected in the TRSA[5:0] bits of the A/D conversion start trigger select register (ADSTRGR), proceed with single scanning twice as follows.
Deselect self-diagnosis and set the temperature sensor A/D conversion select bit (ADEXICR.TSSA, ADEXICR.TSSB) and internal reference voltage A/D conversion select bit (ADEXICR.OCSA, ADEXICR.OCSB) to 0.
Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1. When the ADCSR.DBLE bit is set to 1, channel selection using the ADANSA0 register is invalid. In extended double trigger mode, the ADCSR.EXTRG bit should be set to 0, and the ADCSR.TRGE bit should be set to 1. Software trigger should not be used.

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by TRG4AN input, A/D conversion is started on the single channel selected by the ADCSR.DBLANS[4:0] bits.
(2) When A/D conversion is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy) and A/D data-duplication register A (ADDBLDRA).
(3) The ADCSR.ADST bit is automatically cleared to 0 and the 12-bit A/D converter enters a wait state. Here, a scan end interrupt request is not generated irrespective of the ADCSR.ADIE bit setting (interrupt generation upon scanning completion enabled).
(4) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by TRG4BN input, A/D conversion is started on the single channel selected by the ADCSR.DBLANS[4:0] bits.
(5) When A/D conversion is completed, the A/D conversion result is stored into A/D data duplication register (ADDBLDR) and A/D data duplication register B (ADDBLDRB).
(6) If the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled), a scan end interrupt request is generated.
(7) The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion is completed. Then the 12-bit A/D converter enters a wait state.

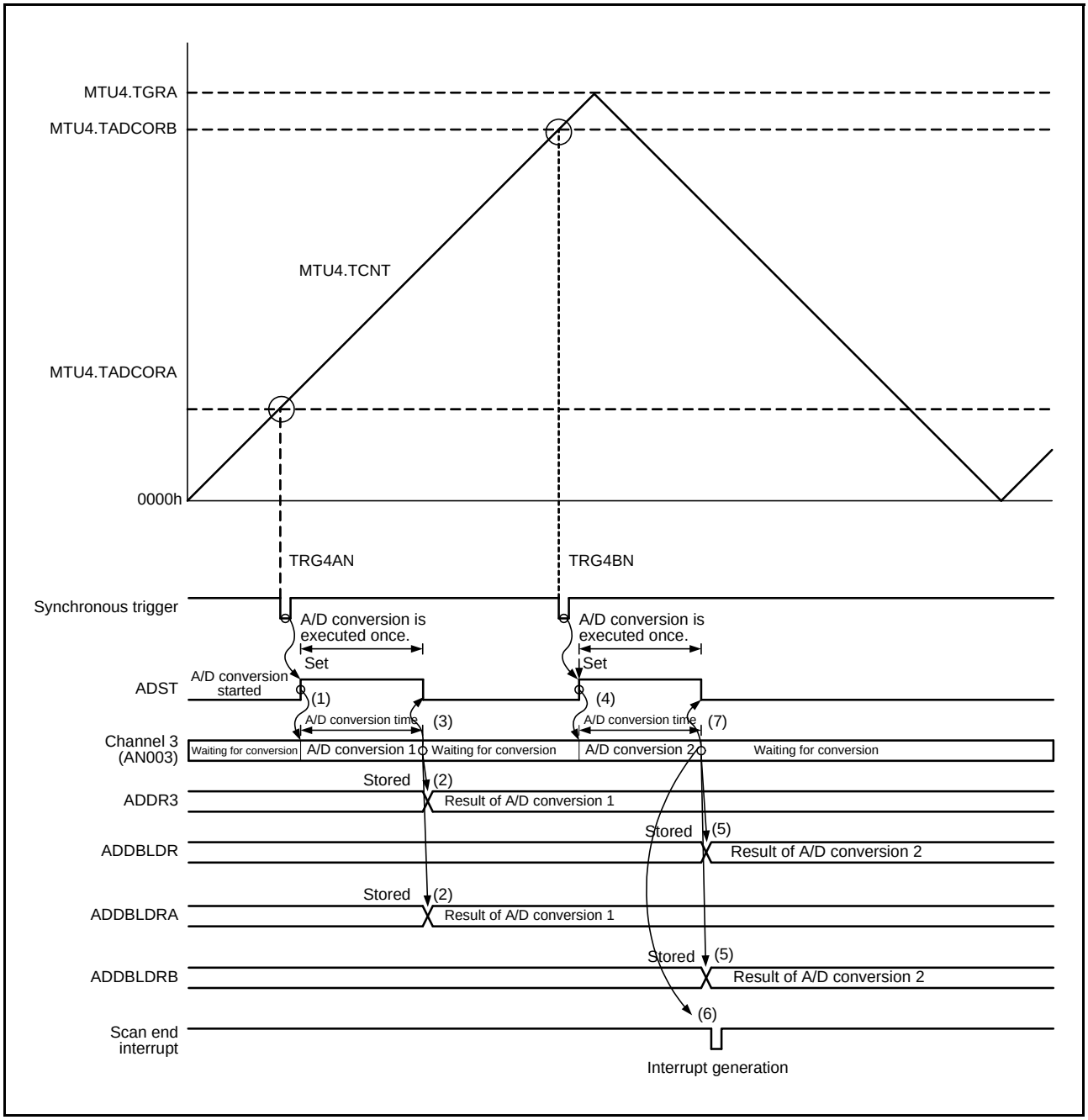


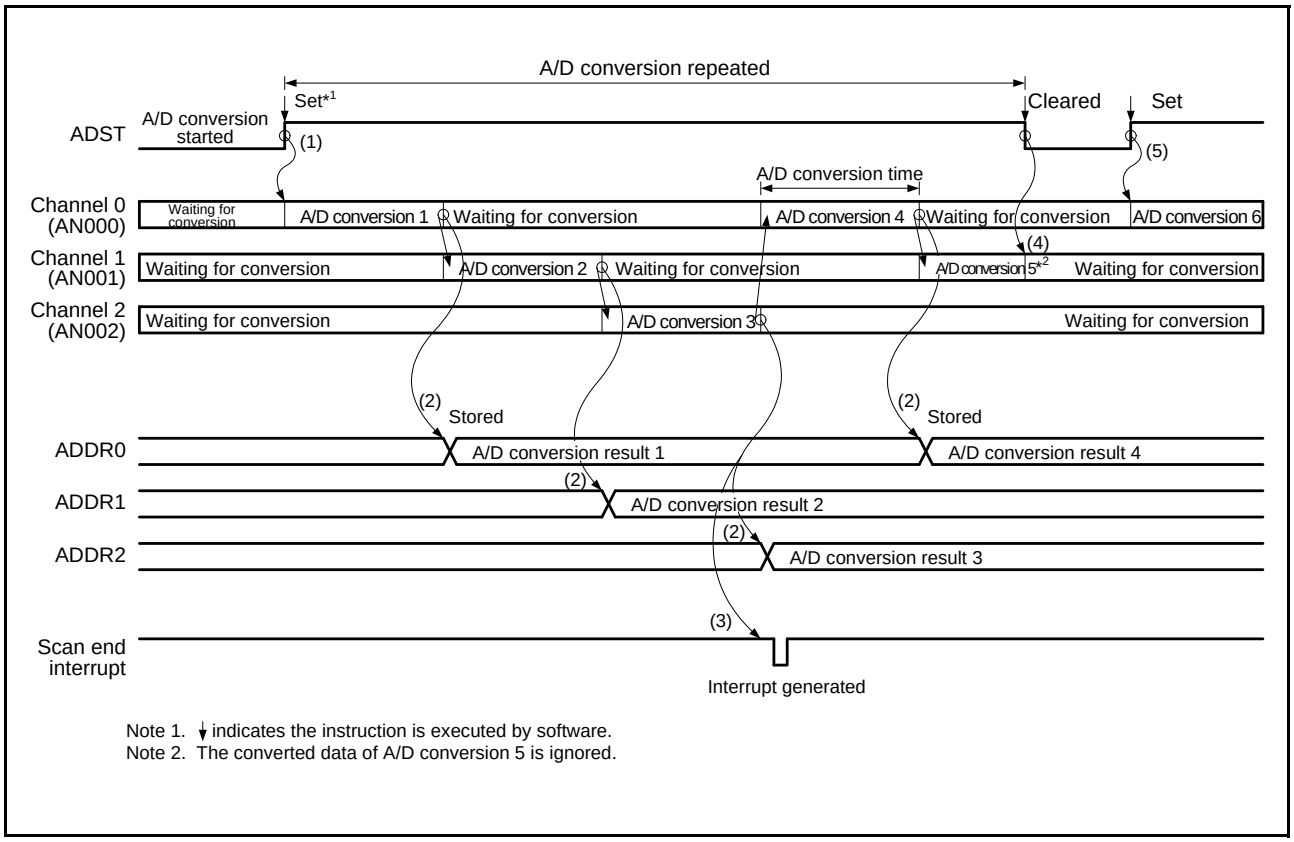
Figure 50.10 Example of Extended Operation in Double Trigger Mode (1)
(Duplication Selected for AN003, TRG4AN or TRG4BN Selected, First Trigger is TRG4AN)

50.3.3 Continuous Scan Mode

50.3.3.1 Basic Operation

In basic operation of continuous scan mode, A/D conversion is performed repeatedly on the analog input of the specified channels as below.

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, or synchronous or asynchronous trigger input, A/D conversion is performed for ANn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.
(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(3) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).
The 12-bit A/D converter sequentially starts A/D conversion for ANn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.
(4) The ADCSR.ADST bit is not automatically cleared to 0 and steps 2 and 3 are repeated as long as the bit remains 1 (A/D conversion start). When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a wait state.
(5) When the ADST bit is later set to 1 (A/D conversion start), A/D conversion is started again for ANn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.



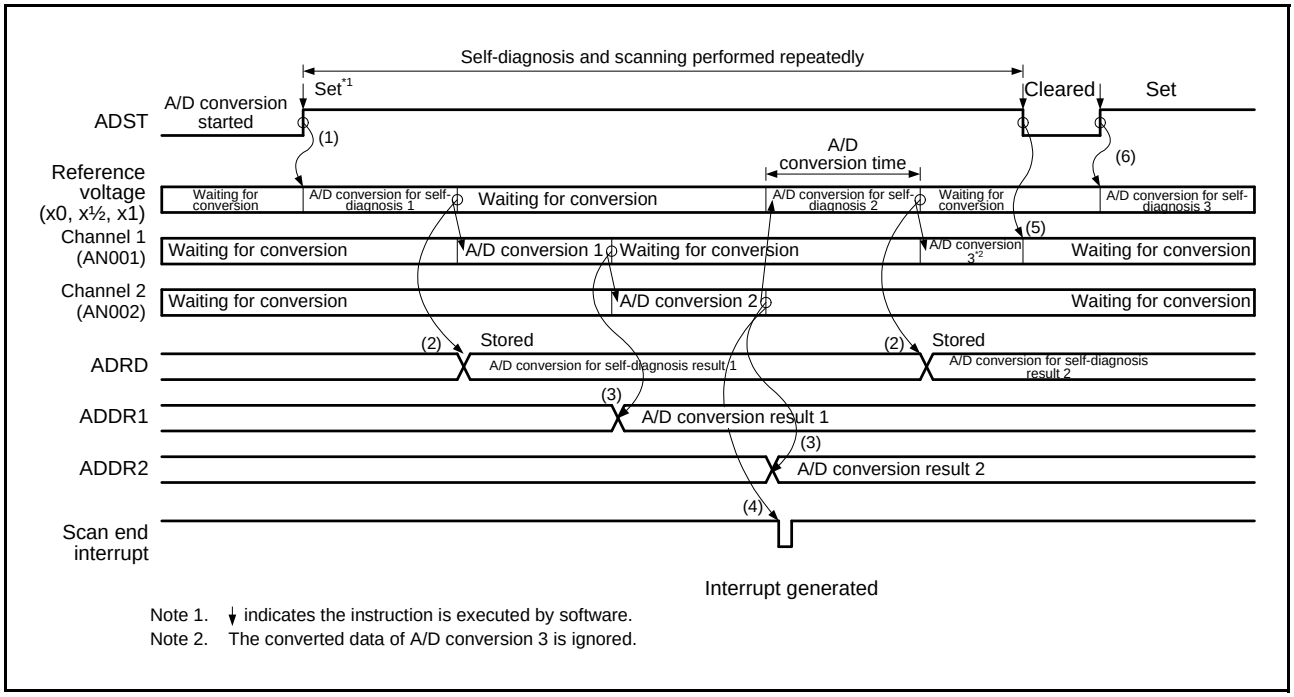
Note 1. ↓ indicates the instruction is executed by software.
Note 2. The converted data of A/D conversion 5 is ignored.

Figure 50.11 Example of Operation in Continuous Scan Mode (Basic Operation: AN000 to AN002 Selected)

50.3.3.2 Channel Selection and Self-Diagnosis

When channels and self-diagnosis are selected at the same time, A/D conversion is first performed for the reference voltage supplied to the 12-bit A/D converter, and then A/D conversion is performed on the analog input of the selected channels, which sequence is repeated as below.

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, or synchronous or asynchronous trigger input, A/D conversion for self-diagnosis is started first.
(2) When A/D conversion for self-diagnosis is completed, the A/D conversion result is stored into the A/D self-diagnosis data register (ADRD). A/D conversion is then performed for ANn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.
(3) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(4) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled). At the same time, the 12-bit A/D converter starts A/D conversion for self-diagnosis and then starts A/D conversion on ANn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.
(5) The ADCSR.ADST bit is not automatically cleared and steps 2 to 4 are repeated as long as the bit remains 1. When the ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a wait state.
(6) When the ADCSR.ADST bit is later set to 1 (A/D conversion start), the A/D conversion for self-diagnosis is started again.



Note 1. ↓ indicates the instruction is executed by software.
Note 2. The converted data of A/D conversion 3 is ignored.

Figure 50.12 Example of Operation in Continuous Scan Mode (Basic Operation; AN001 and AN002 Selected + Self-Diagnosis)

50.3.3.3 A/D Conversion With Temperature Sensor Output/Internal Reference Voltage Selected

When temperature sensor output or internal reference voltage are selected with channels, A/D conversion of analog input of the channels is performed, after that A/D conversion is performed for the temperature sensor output or internal reference voltage as described below. After that, A/D conversion is performed repeatedly on the analog input of the selected channels. When temperature sensor output and internal reference voltage are both selected, A/D conversion is performed temperature sensor output first and then internal reference voltage.
Deselecting channels and selecting only one of temperature sensor output and internal reference voltage is possible.

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, or synchronous or asynchronous trigger input, and A/D conversion is performed for ANn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.
(2) Each time A/D conversion of a single channel is completed, the A/D converted value is stored in the corresponding A/D data register (ADDRy), and then A/D conversion for temperature sensor output is started.
(3) When A/D conversion of the temperature sensor output is completed, the A/D converted value is stored in the corresponding A/D temperature sensor data register (ADTSDR), and then A/D conversion for internal reference voltage is started.
(4) After completion of A/D conversion of the internal reference voltage, the A/D converted value is stored in the corresponding A/D internal reference voltage data register (ADOCDR), and if the ADCSR.ADIE bit is set to 1 (when interrupt generation after scanning is enabled), a scan end interrupt request is generated. The 12-bit A/D converter sequentially starts A/D conversion for ANn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.
(5) While the ADCSR.ADST bit holds 1 without being automatically cleared, steps 2 to 4 above are repeated. When the ADCSR.ADST bit is cleared to 0 (A/D conversion stop), A/D conversion is stopped and the 12-bit A/D converter enters into a wait state.
(6) After that, when the ADSHMSR.SHMD bit is set to 1 (A/D conversion start), A/D conversion is performed for ANn channels selected in the ADANSA0 register starting from the channel with the smallest number n.

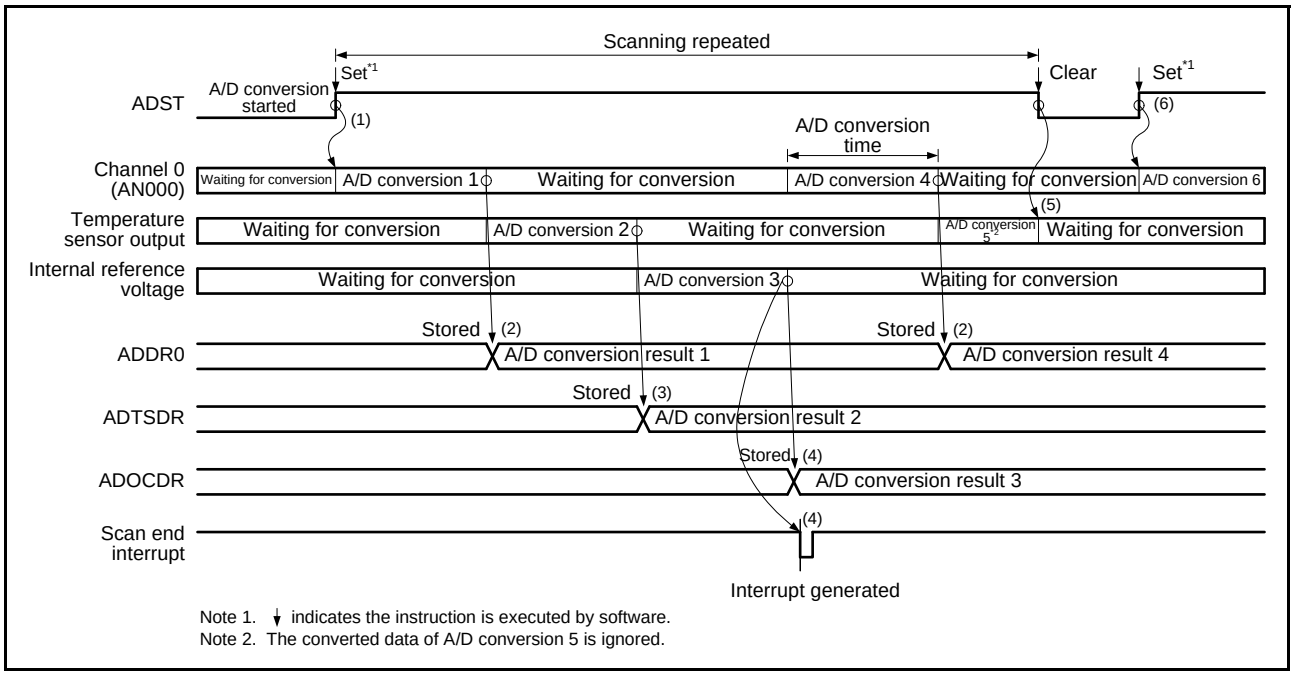


Figure 50.13 Example of Operation in Sequence Scan Mode (Basic Operation: AN000, Temperature Sensor Output and Internal Reference Voltage Selected)

50.3.4 Group Scan Mode

50.3.4.1 Basic Operation

Either two (groups A and B) or three (groups A, B, and C) can be selected as the number of the groups to be used in group scan mode.
In basic operation of group scan mode, A/D conversion is performed once on the analog inputs of all the specified channels in groups A and B, or groups A, B, and C after scan is started by a synchronous trigger as below. Scan operation of each group is similar to the scan operation in single scan mode.
The synchronous triggers of groups A and B, or groups A, B, and C can be selected using the TRSA[5:0], TRSB[5:0], and TRSC[5:0] bits in ADSTRGR, respectively. Different triggers should be used for each group A, B, and C so that scanning of groups A, B, and C does not occur simultaneously. Software trigger should not be used.
The channels to be scanned are selected using register ADANSA0 and bits TSSA and OCSA in register ADEXICR for group A, register ADANSB0 and bits TSSB and OCSB in register ADEXICR for group B, and register ADANSC0 and bits TSSC and OCSC in register ADGCEXCR for group C.
When self-diagnosis is selected in group scan mode, self-diagnosis is separately executed for groups A and B, or groups A, B, and C.
The following describes operation in group scan mode using a trigger from the MTU. The TRG4AN, TRG4BN, and TRG4ABN triggers from the MTU are assumed to be used to start conversion of groups A, B and C, respectively.

(1) Scanning of group A is started by the TRG4AN trigger from the MTU.
(2) When group A scanning is completed, a scan end interrupt is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).
(3) Scanning of group B is started by the TRG4BN trigger from the MTU.
(4) When group B scanning is completed, a group B scan end interrupt is generated if the ADCSR.GBADIE bit is 1 (interrupt generation upon group B scan completion enabled).
(5) Scanning of group C is started by the TRG4ABN trigger from the MTU.
(6) When group C scanning is completed, a group C scan end interrupt is generated if the ADGCTRGR.GCADIE bit is 1 (interrupt generation upon group C scan completion enabled).

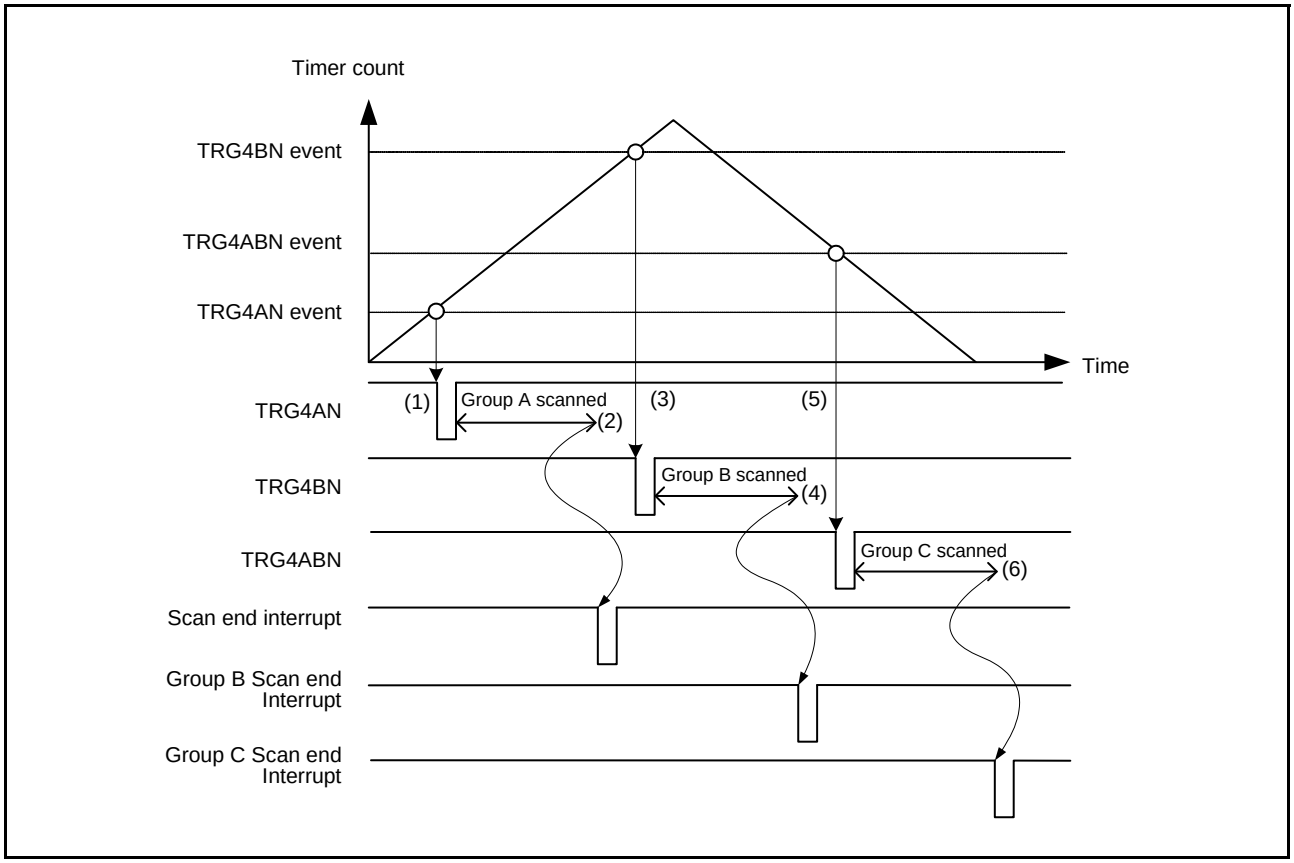


Figure 50.14 Example of Operation in Group Scan Mode (Basic Operation: Synchronous Triggers from MTU Used)

50.3.4.2 A/D Conversion in Double Trigger Mode

When double trigger mode is selected in group scan mode, two rounds of single scan operation started by a synchronous trigger are performed as a sequence for group A. For groups B and C, single scan operation started by a synchronous trigger is performed once.

In group scan mode, the synchronous triggers of groups A and B, or groups A, B, and C can be selected using the TRSA[5:0], TRSB[5:0], and TRSC[5:0] bits in ADSTRGR, respectively. Different triggers should be used for each group A, B, and C so that scanning of groups A, B, and C does not occur simultaneously. Software trigger and asynchronous trigger should not be used. When the TRG4AN or TRG4BN, TRG7AN or TRG7BN is selected in the ADSTRGR.TRSA[5:0] bits as the synchronous trigger of the group A, operation is in the extended double trigger mode. The channels to be scanned are selected using bits ADCSR.DBLANS[4:0] for group A, register ADANSB0 for group B, and register ADANSC0 for group C.

When double trigger mode is selected in group scan mode, the temperature sensor A/D conversion select bit (ADEXICR.TSSA) and internal reference voltage A/D conversion select bit (ADEXICR.OCSA) are set to 0 (not selected).

When double trigger mode is selected in group scan mode, self-diagnosis cannot be selected.

Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1.

The following describes operation in group scan mode with double trigger mode using a synchronous trigger from the MTU. The TRG4ABN, TRGA0N, and TRGA1N triggers from the MTU are assumed to be used to start conversion of groups A, B, and C, respectively.

(1) Scanning of group C is started by the TRGA1N trigger from the MTU.
(2) When group C scanning is completed, a group C scan interrupt is generated if the ADGCTRGR.GCADIE bit is 1 (interrupt generation upon group C scan completion enabled).
(3) Scanning of group B is started by the TRGA0N trigger from the MTU.
(4) When group B scanning is completed, a group B scan end interrupt is generated if the ADCSR.GBADIE bit is 1 (interrupt generation upon group B scan enabled).
(5) The first scanning of group A is started by the first TRG4ABN trigger from the MTU.
(6) When the first scanning of group A is completed, the conversion result is stored into the corresponding A/D data register (ADDRy); a scan end interrupt request is not generated irrespective of the ADIE bit setting in ADCSR.
(7) The second scanning of group A is started by the second TRG4ABN trigger from the MTU.
(8) When the second scanning of group A is completed, the conversion result is stored into ADDBLDR. a scan end interrupt is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).
(9) The second scanning of group B is started by the second TRGA0N trigger from the MTU.
(10) When the second scanning of group B is completed, a group B scan interrupt is generated if the ADCSR.GBADIE bit is 1 (interrupt generation upon group B scan completion enabled).
(11) The second scanning of group C is started by the second TRGA1N trigger from the MTU.
(12) When the second scanning of group C is completed, a group C scan interrupt is generated if the ADGCTRGR.GCADIE bit is 1 (interrupt generation upon group C scan completion enabled).

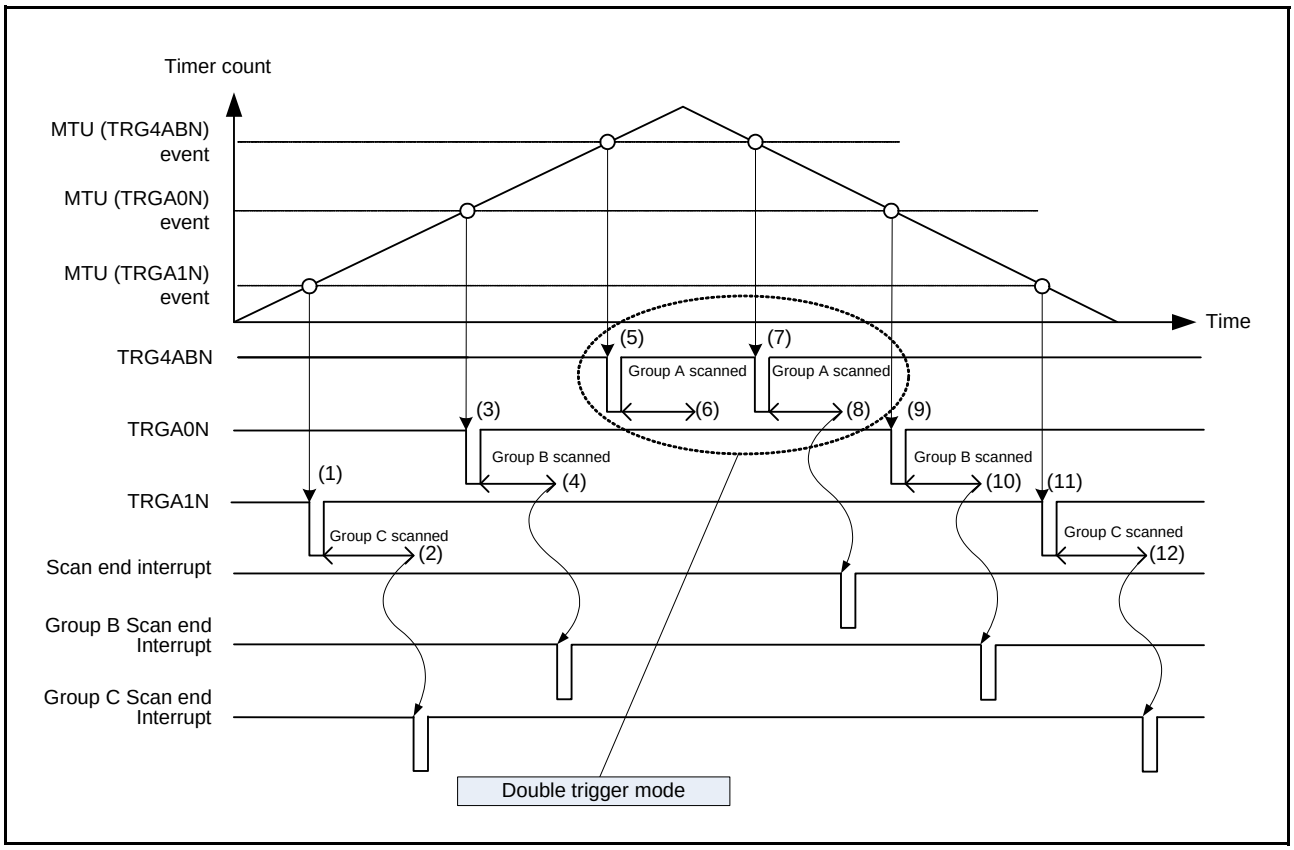


Figure 50.15 Example of Operation in Group Scan Mode with Double Trigger Mode (Basic Operation: Synchronous Triggers from MTU Used)

50.3.4.3 Operation under Group Priority Control

Setting the ADGSPCR.PGS bit to 1 in group scan mode makes operation proceed under group priority control. The group priority order is group A > group B > group C. Either two (groups A and B) or three (groups A, B, and C) can be selected as the number of the groups to be used in group scan mode. When setting the ADGSPCR.PGS bit to 1, follow the procedure described in Figure 50.16. If the procedure is not followed, proper scanning operation and data storage are not guaranteed.

In basic operation of group scan mode, if group A, B, or C is scanning, all other trigger inputs are ignored. Under group priority control, if a priority group trigger is input during A/D conversion for the low-priority group, A/D conversion for the low-priority group is discontinued and A/D conversion for the priority group proceeds.

If the ADGSPCR.GBRSCN bit is 0, the converter enters a wait state for the low-priority group on completion of the A/D conversion for the priority group.

The trigger input for the low-priority group during scanning is ignored.

If the ADGSPCR.GBRSCN bit is 1, the converter automatically restarts scanning for the low-priority group from the head of the group after A/D conversion for the priority group.

Also, the trigger input for the low-priority group generated during scanning for the priority group is enabled, and the converter automatically restarts scanning for the low-priority group after scanning for the priority group.

When the ADGSPCR.LGRRS bit is 0 while the ADGSPCR.GBRSCN bit is 1, the converter restarts scanning for the low-priority group from the head of the group. When the ADGSPCR.LGRRS bit is 1, the converter restarts scanning for the low-priority group from the channel on which scanning is discontinued.

However, when self-diagnosis is used, the converter restarts scanning from the channel on which scanning is discontinued after self-diagnosis is completed.

Table 50.15 summarizes operations in response to the input of a trigger during scanning with the settings of the ADGSPCR.GBRSCN bit.

When the ADGSPCR.GBRP bit is set to 1, scan operations in the lowest-priority group are continuously performed.

For the trigger settings in group scan mode, select a synchronous trigger for group A using the ADSTRGR.TRSA[5:0] bits, select a synchronous trigger different from that of group A for group B using the ADSTRGR.TRSB[5:0] bits, and select a synchronous trigger different from those of groups A and B for group C using the ADGCTRGR.TRSC[5:0] bits.

Set the ADSTRGR.TRSB[5:0] bits to 3Fh when setting group scan mode for two groups (the ADGCTRGR.GRCE bit to 0) and setting the ADGSPCR.GBRP bit to 1.

Set the ADGCTRGR.TRSC[5:0] bits to 3Fh when setting group scan mode for three groups (the ADGCTRGR.GRCE bit to 1) and setting the ADGSPCR.GBRP bit to 1.

Furthermore, as targets for scanning, select channels for group A by using the ADANSA0 register, and ADEXICR.TSSA and ADEXICR.OCSA bits; select the group B channels by using ADANSB0 register, and ADEXICR.TSSB and ADEXICR.OCSB bits; and select the group c by using ADANSC0 register, and ADGCEXCR.TSSC and ADGCEXCR.OCSC bits respectively.

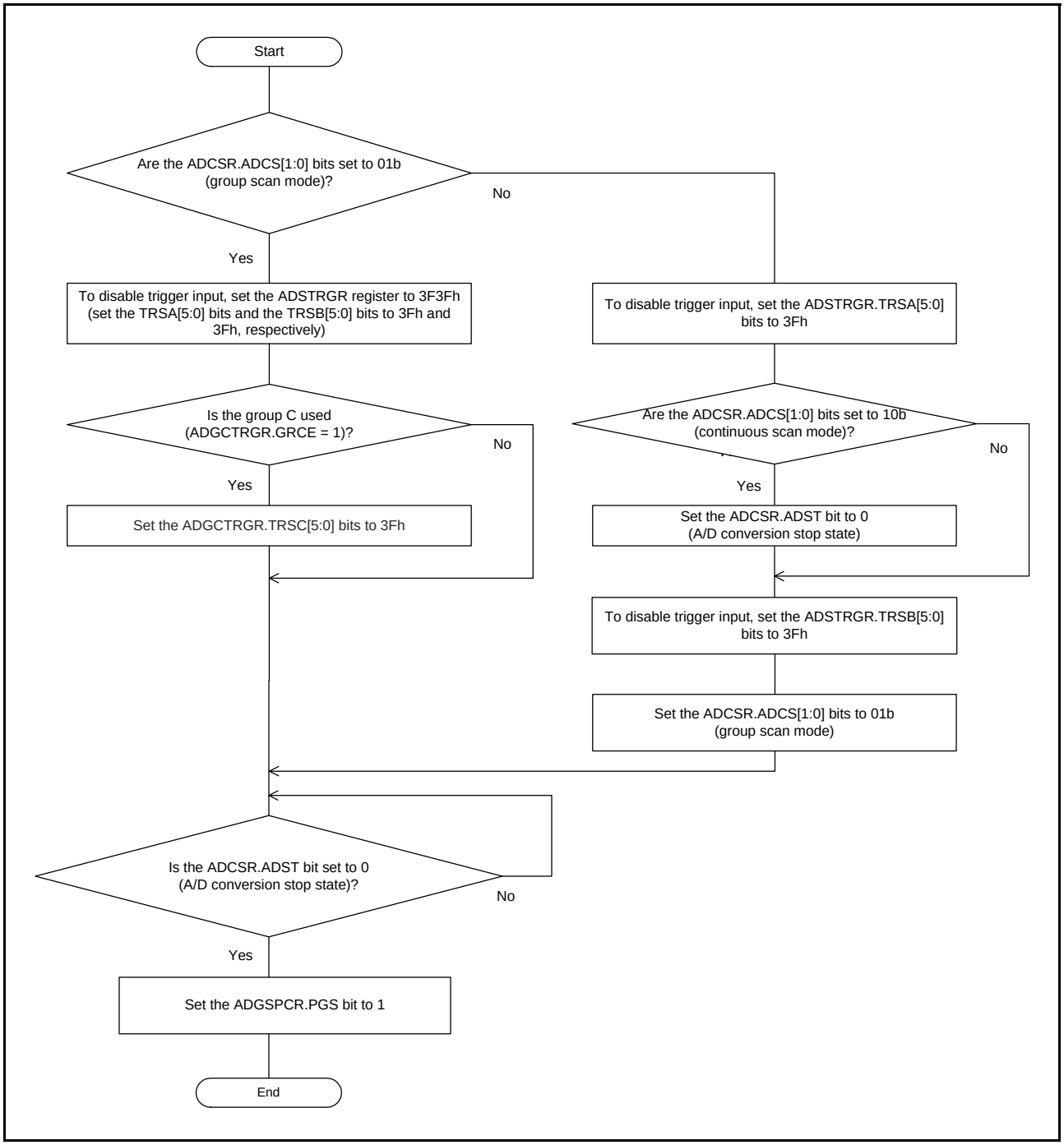


Figure 50.16 Flow of Setting the ADGSPCR.PGS Bit

Table 50.15 Control of Scanning Operations According to the Settings of the ADGSPCR.GBRSCN Bit

Scanning Operation	Trigger Input	ADGSPCR.GBRSCN = 0	ADGSPCR.GBRSCN = 1
When A/D conversion for group A is in progress	Input of trigger for group A	Trigger input is ineffective.	Trigger input is ineffective.
	Input of trigger for group B	Trigger input is ineffective.	A/D conversion is performed on group B after A/D conversion on group A is completed.
	Input of trigger for group C	Trigger input is ineffective.	A/D conversion is performed on group C after A/D conversion on group A is completed.
When A/D conversion for group B is in progress	Input of trigger for group A	A/D conversion in progress for group B discontinued and conversion for group A starts.	• A/D conversion in progress for group B is discontinued and conversion for group A starts. • A/D conversion for group B starts after conversion for group A is completed.
	Input of trigger for group B	Trigger input is ineffective.	Trigger input is ineffective.
	Input of trigger for group C	Trigger input is ineffective.	A/D conversion is performed on group C after A/D conversion on group B is completed.
When A/D conversion for group C is in progress	Input of trigger for group A	A/D conversion in progress for group C discontinued and conversion for group A starts.	• A/D conversion in progress for group C is discontinued and conversion for group A starts. • A/D conversion for group C starts after conversion for group A is completed.
	Input of trigger for group B	A/D conversion in progress for group C is discontinued and conversion for group B starts.	• A/D conversion in progress for group C is discontinued and conversion for group B starts. • A/D conversion for group C starts after conversion for group B is completed.
	Input of trigger for group C	Trigger input is ineffective.	Trigger input is ineffective.

When using group priority operation mode, refer to the following tables to select the desirable operating mode and set the registers.

Table 50.16 Group Priority Operation Setting and Operating Mode for Two Groups (ADGSPCR.PGS = 1, ADGCTRGR.GRCE = 0)

ADGSPCR			
GBRSCN	**LGRRS**	**GBRP**	**Operation**
0	x	0	Group priority operation for two groups (groups A and B) • When a group A trigger is input, group B scan is completed (not restarted)
1	0	0	Group priority operation for two groups (groups A and B) • After group B scan is discontinued, group B scan is restarted from the head of the channel selected with the ADANSB0 register after group A scan is completed.
1	1	0	Group priority operation for two groups (groups A and B) • After group B scan is discontinued, group B scan is restarted from the channel on which scan was discontinued*1, among the channels selected with the ADANSB0 register after group A scan is completed.
x	0	1	Group priority operation for two groups (groups A and B) • Single scan for group B is started continuously without start trigger input. After group B scan is discontinued, single scan is restarted from the head of the channel selected with the ADANSB0 register after scan for group A is completed.
1	1	1	Group priority operation for two groups (groups A and B) • Single scan for group B is started continuously without start trigger input. After group B scan was discontinued, single scan is restarted from the channel on which scan was discontinued*1, among the channels selected with the ADANSB0 register after group A scan is completed.

x = Don't care

Note 1. When self-diagnosis is used (ADCER.DIAGM = 1), A/D conversion of the discontinued channel is started after self-diagnosis.

Table 50.17 Group Priority Operation Setting and Operating Mode for Three Groups (ADGSPCR.PGS = 1, ADGCTRGR.GRCE = 1)

ADGSPCR			
GBRSCN	**LGRRS**	**GBRP**	**Operation**
0	x	0	Group priority operation for three groups (groups A, B, and C) • When a group A trigger is input, group B scan is completed (not restarted) • When a trigger for group A or B is input, group C scan is not completed (not restarted).
0	x	1	Group priority operation for three groups (groups A, B, and C) • When a group A trigger is input, group B scan is completed (not restarted) • Single scan for group C is started continuously without start trigger input. After group C scan is discontinued, scan is restarted from the head of the channel selected with the ADANSC0 register after scan for groups A and B is completed.
1	0	0	Group priority operation for three groups (groups A, B, and C) • After group B scan is discontinued, scan is restarted from the head of the channel selected with the ADANSB0 register after group A scan is completed. • After group C scan is discontinued, scan is restarted from the head of the channel selected with the ADANSC0 register after scan for groups A and B is completed.
1	1	0	Group priority operation for three groups (groups A, B, and C) • After group B scan is discontinued, scan is restarted from the head of the channel selected with the ADANSB0 register after group A scan is completed. • After group C scan is discontinued, scan is restarted from the channel on which scan was discontinued*1, among the channels selected with the ADANSC0 register after scan for groups A and B is completed.
1	0	1	Group priority operation for three groups (groups A, B, and C) • After group B scan is discontinued, scan is restarted from the head of the channel selected with the ADANSB0 register after group A scan is completed. • Single scan for group C is started continuously without start trigger input. After group C scan is discontinued, single scan is restarted from the head of the channel selected with the ADANSC0 register after scan for groups A and B is completed.
1	1	1	Group priority operation for three groups (groups A, B, and C) • After scanning for group B is discontinued, once scanning for group A completes, scanning is resumed starting from the channel on which scanning was discontinued*1, among those channels selected with the ADANSB0 register. • Single scan for group C is started continuously without start trigger input. After group C scan is discontinued, single scan is restarted from the channel on which scan was discontinued*1, among the channels selected with the ADANSC0 register after scan for groups A and B is completed.

x = Don’t care

Note 1. When self-diagnosis is used (ADCER.DIAGM = 1), A/D conversion of the discontinued channel is started after self-diagnosis.

(1) Group Priority Operation for Two Groups (ADGSPCR.PGS = 1, ADGCTRGR.GRCE = 0)

The following examples 1 to 5 show group priority operation in group scan mode (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0) when channel 0 is selected for group A and channels 1 to 3 are selected for group B.

Operation example 1: Group A trigger input during group B scan, with rescan setting

(1) When a group B trigger input sets the ADCSR.ADST bit to 1 (A/D conversion start), scan for the ANn channels selected in the ADANSB0 register, starts from the channel with the smallest number n.
(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(3) If a group A trigger is input during scan for group B, group B scan is discontinued while the ADCSR.ADST bit remains 1, and scan for the ANn channels of group A selected in the ADANSA0 register, starts from the channel with the smallest number n.
If A/D conversion is not completed when scan was discontinued, the result is not stored in the corresponding A/D data register (ADDRy).
(4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(5) A scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).
(6) If the ADGSPCR.GBRSCN bit is 1 (scan for the group is restarted after having been discontinued due to group priority operation), scan for the ANn channels of group B selected in the ADANSB0 register, restarts from the channel with the smallest number n while the ADCSR.ADST bit remains 1.
(7) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(8) A group B scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (interrupt generation upon group B scan completion enabled).
(9) The ADCSR.ADST bit is automatically cleared to 0 when scan of all selected channels is completed, and the 12-bit A/D converter enters a wait state.

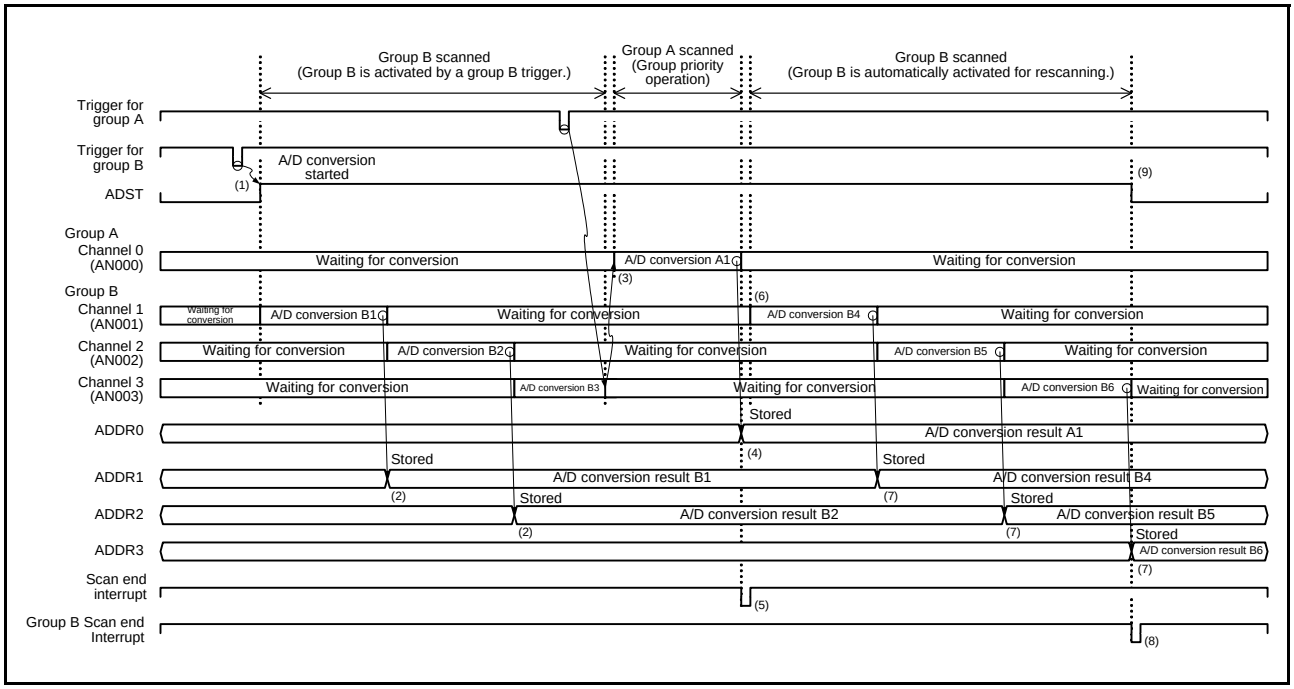


Figure 50.17 Example 1 of Group Priority Operation: Group A Trigger Input during Group B Scan, With Rescan Setting (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0)

Operation example 2: Group A trigger input during group B rescan, with rescan setting

Figure 50.18 shows an example when a group A trigger is input during rescan operation on group B.

If a group A trigger is input, scan for group A starts even while rescan operation is in progress. Scan for group B starts after scan for group A is completed.

Operations of the ADCSR.ADST bit, storing to the A/D conversion result in the A/D data register (ADDRy), and interrupt requests are the same as example 1.

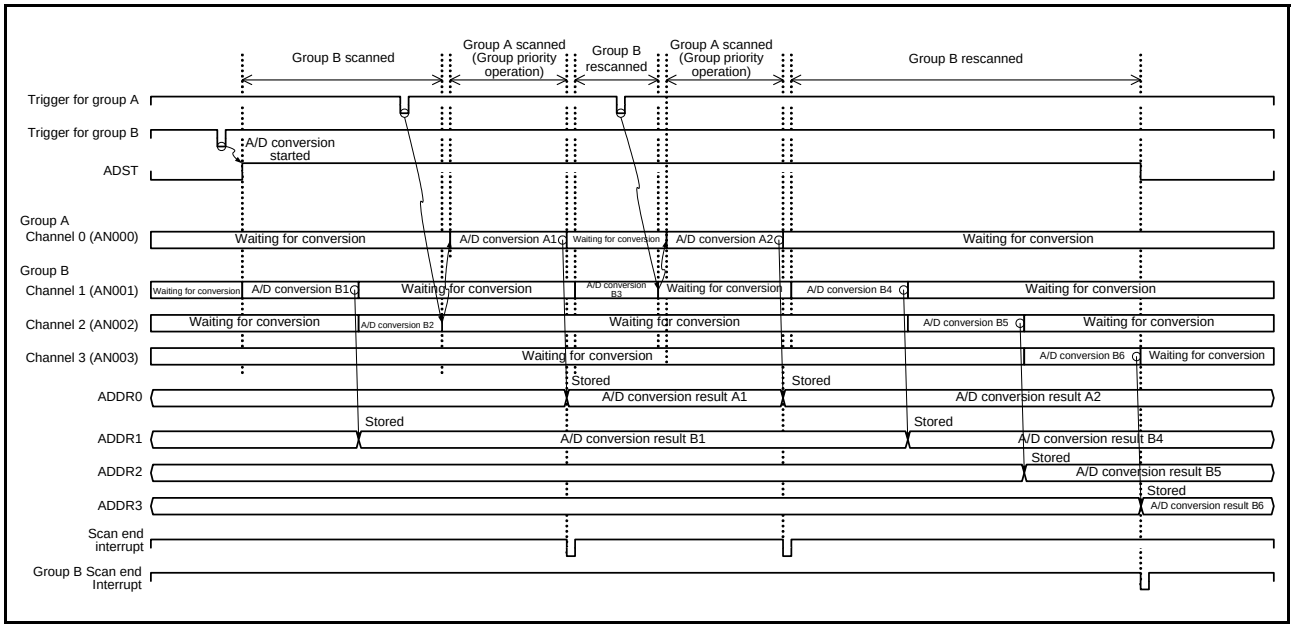


Figure 50.18 Example 2 of Group Priority Operation: Group A Trigger Input during Group B Rescan, With Rescan Setting (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0)

Operation example 3: Group B trigger input during group A scan, with rescan setting

The following describes an example when a group B trigger is input during scan operation on group A when the ADGPSCR.GBRSCN bit is 1 (scan for the group is restarted after having been discontinued due to group priority operation).

If the ADGPSCR.GBRSCN bit is 0, all group B triggers that are input during scan operation on group A are disabled.

(1) When a group A trigger input sets the ADCSR.ADST bit to 1 (A/D conversion start), scan for the ANn channels of group A selected in the ADANSA0 register, starts from the channel with the smallest number n.
(2) If a group B trigger is input during scan for group A, scan for group B can be started.
(3) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(4) After scan for group A is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).
(5) After scan for group A is completed, scan for the ANn channels of group B selected in the ADANSB0 register, starts from the channel with the smallest number n while the ADCSR.ADST bit remains 1. When a group A trigger is input during scan for group B, group A scan starts as in example 1, and group B scan starts after group A scan is completed.
(6) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(7) After scan for group B is completed, a group B scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (interrupt generation upon group B scan completion enabled).
(8) The ADCSR.ADST bit is automatically cleared to 0 when scan of all selected channels is completed, and the 12-bit A/D converter enters a wait state.

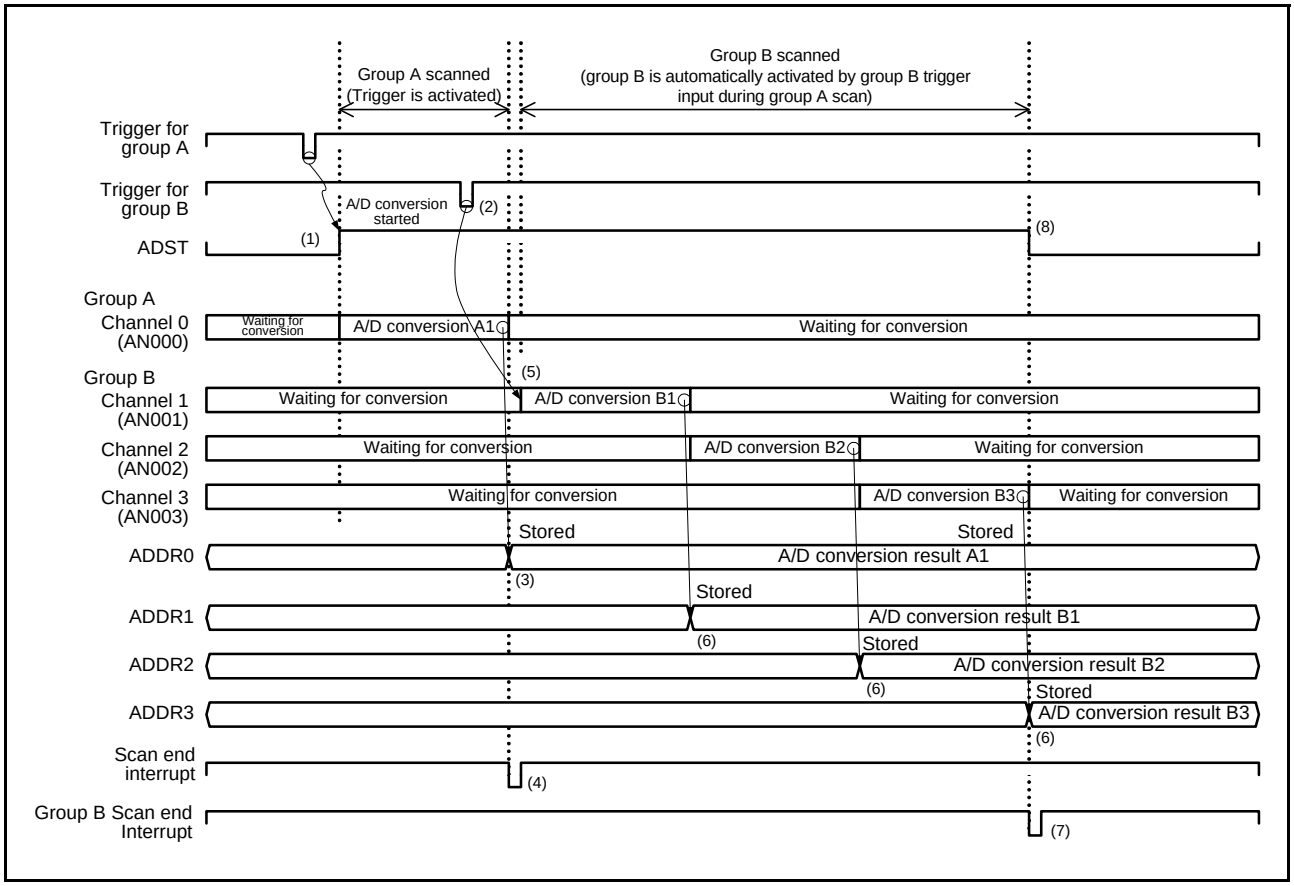


Figure 50.19 Example 3 of Group Priority Operation: Group B Trigger Input during Group A Scan, With Rescan Setting (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0)

Operation example 4 shows group priority operation in group scan mode (ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0) when channel 0 is selected for group A and channels 1 to 3 are selected for group B.

Operation example 4: Group A trigger input during group B scan, without rescan setting

(1) When a group B trigger input sets the ADCSR.ADST bit to 1 (A/D conversion start), scan for the ANn channels of group B selected in the ADANSB0 register, starts from the channel with the smallest number n.

(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).

(3) If a group A trigger is input during scan for group B, group B scan is discontinued while the ADCSR.ADST bit remains 1, and scan for the ANn channels of group A selected in the ADANSA0 register, starts from the channel with the smallest number n.
If A/D conversion is not completed when scan was discontinued, the result is not stored in the corresponding A/D data register (ADDRy).

(4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).

(5) After scan for group A is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).

(6) The ADST bit is automatically cleared to 0 when scan for group A is completed, and the 12-bit A/D converter enters a wait state. Scan for group B is not started until the next group B trigger is input.

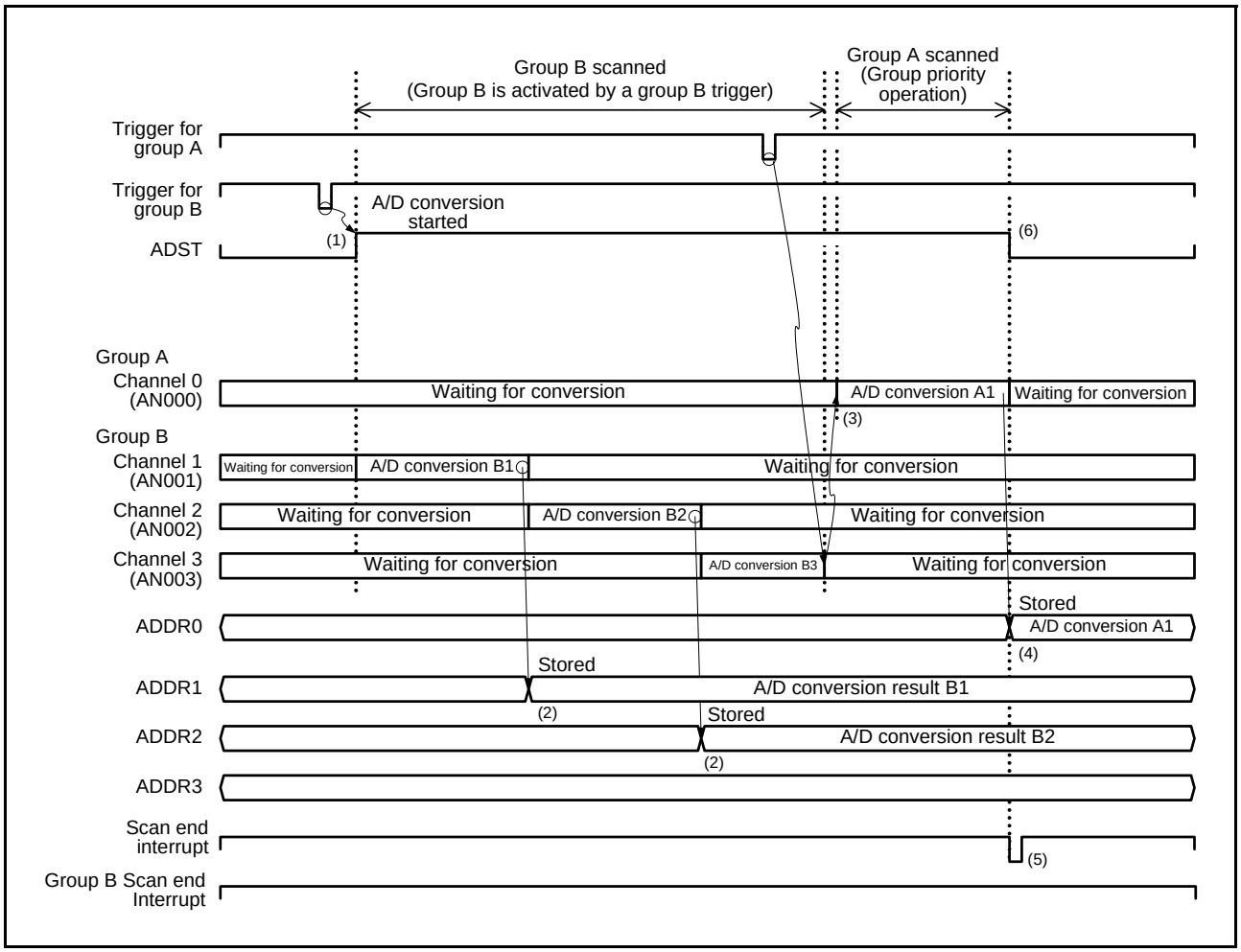


Figure 50.20 Example 4 of Group Priority Operation: Group A Trigger Input during Group B Scan, Without Rescan Setting (ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0)

Operation example 5 shows group priority operation in group scan mode (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 1, ADGSPCR.LGRRS = 0) when channel 0 is selected for group A and channels 1 and 2 are selected for group B. When the ADGCTRGR.GRCE bit is set to 1, single scan mode is continuously operated on group C and scan for group B is started by trigger input.

Operation example 5: Continuous single scan operation on group B

(1) When setting ADGSPCR.GBRP to 1 sets the ADCSR.ADST bit to 1 (A/D conversion start), scan for the ANn channels selected in the ADANSB0 register, starts from the channel with the smallest number n.
(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(3) If a group A trigger is input during scan for group B, group B scan is discontinued while the ADCSR.ADST bit remains 1, and scan for the ANn channels of group A selected in the ADANSA0 register, starts from the channel with the smallest number n. If A/D conversion is not completed when scan was discontinued, the result is not stored in the corresponding A/D data register (ADDRy).
(4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(5) After scan for group A is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).
(6) If the ADGSPCR.GBRP bit is set to 1 (single scan is continuously operated), scan for the ANn channels of group B selected in the ADANSB0 register, restarts from the channel with the smallest number n while the ADCSR.ADST bit remains 1.
(7) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(8) A group B scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (interrupt generation upon group B scan completion enabled).
(9) If the ADGSPCR.GBRP bit is set to 1 (single scan is continuously operated), scan for the ANn channels of group B selected in the ADANSB0 register, restarts from the channel with the smallest number n while the ADCSR.ADST bit remains 1.

To continuously operate single scan for group B, disable trigger input for group B.
Steps 6 to 9 are repeated as long as the ADGSPCR.GBRP bit remains 1. Clearing of the ADCSR.ADST bit to 0 is prohibited while the ADGSPCR.GBRP bit is set to 1. To forcibly stop scanning when ADGSPCR.GBRP = 1, follow the procedures for clear operation by software through the ADCSR.ADST bit shown in section 50.6.2, Notes on Stopping A/D Conversion.

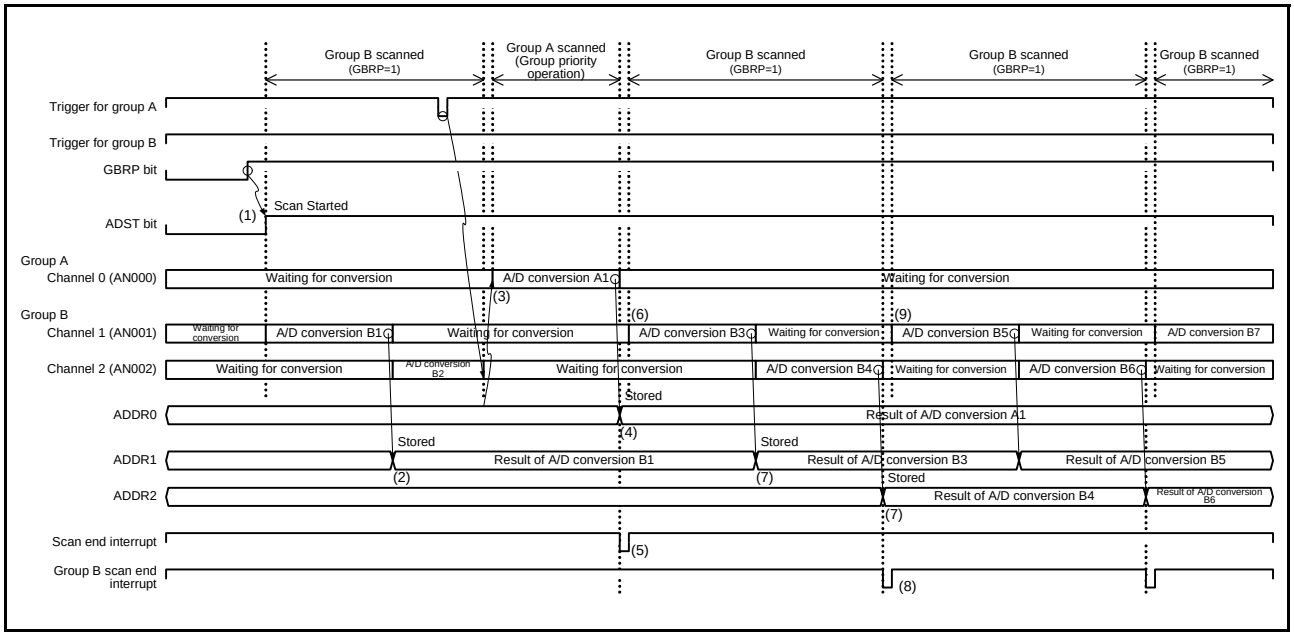


Figure 50.21 Example 5 for Group Priority Operation: Continuous Single Scan Operation on Group B (ADGSPCR.GBRP = 1, ADGSPCR.LGRRS = 1, ADGCTRGR.GRCE = 0)

(2) Group Priority Operation for Three Groups (ADGSPCR.PGS = 1, ADGCTRGR.GRCE = 1)

The following examples 1 to 5 show group priority operation in group scan mode (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 1) when channel 0 is selected for group A, channels 1 and 2 are selected for group B, and channels 3 and 4 are selected for group C.
The priority groups mean groups A and B for group C and group A for group B.

Operation example 1: Priority group trigger input during low-priority group scan, with rescan setting

(1) When a group C trigger input sets the ADCSR.ADST bit to 1 (A/D conversion start), scan for the ANn channels selected in the ADANSC0 register, starts from the channel with the smallest number n.
(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(3) If a group B trigger is input during scan for group C, group C scan is discontinued while the ADCSR.ADST bit remains 1, and scan for the ANn channels of group B selected in the ADANSB0 register, starts from the channel with the smallest number n. On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(5) If a group A trigger is input during scan for group B, group B scan is discontinued while the ADCSR.ADST bit remains 1, and scan for the ANn channels of group A selected in the ADANSA0 register, starts from the channel with the smallest number n. If A/D conversion is not completed when scan was discontinued, the result is not stored in the corresponding A/D data register (ADDRy).
(6) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(7) A scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).
(8) If the ADGSPCR.GBRSCN bit is 1 (scan for the group is restarted after having been discontinued due to group priority operation), scan for the ANn channels of group B selected in the ADANSB0 register, restarts from the channel with the smallest number n while the ADCSR.ADST bit remains 1. If the ADGSPCR.LGRRS bit is set to 1 at this time, scan for group B starts from the channel on which A/D conversion was discontinued.
(9) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(10) A group B scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (interrupt generation upon group B scan completion enabled).
(11) If the ADGSPCR.GBRSCN bit is 1, scan for the ANn channels of group C selected in the ADANSC0 register, restarts from the channel with the smallest number n while the ADCSR.ADST bit remains 1. If the ADGSPCR.LGRRS bit is set to 1 at this time, scan for group C starts from the channel on which A/D conversion was discontinued.
(12) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(13) A group C scan end interrupt is generated if the ADGCTRGR.GCADIE bit is 1 (interrupt generation upon group C scan completion enabled).
(14) The ADST bit is automatically cleared to 0 when scan for group A is completed, and the 12-bit A/D converter enters a wait state.

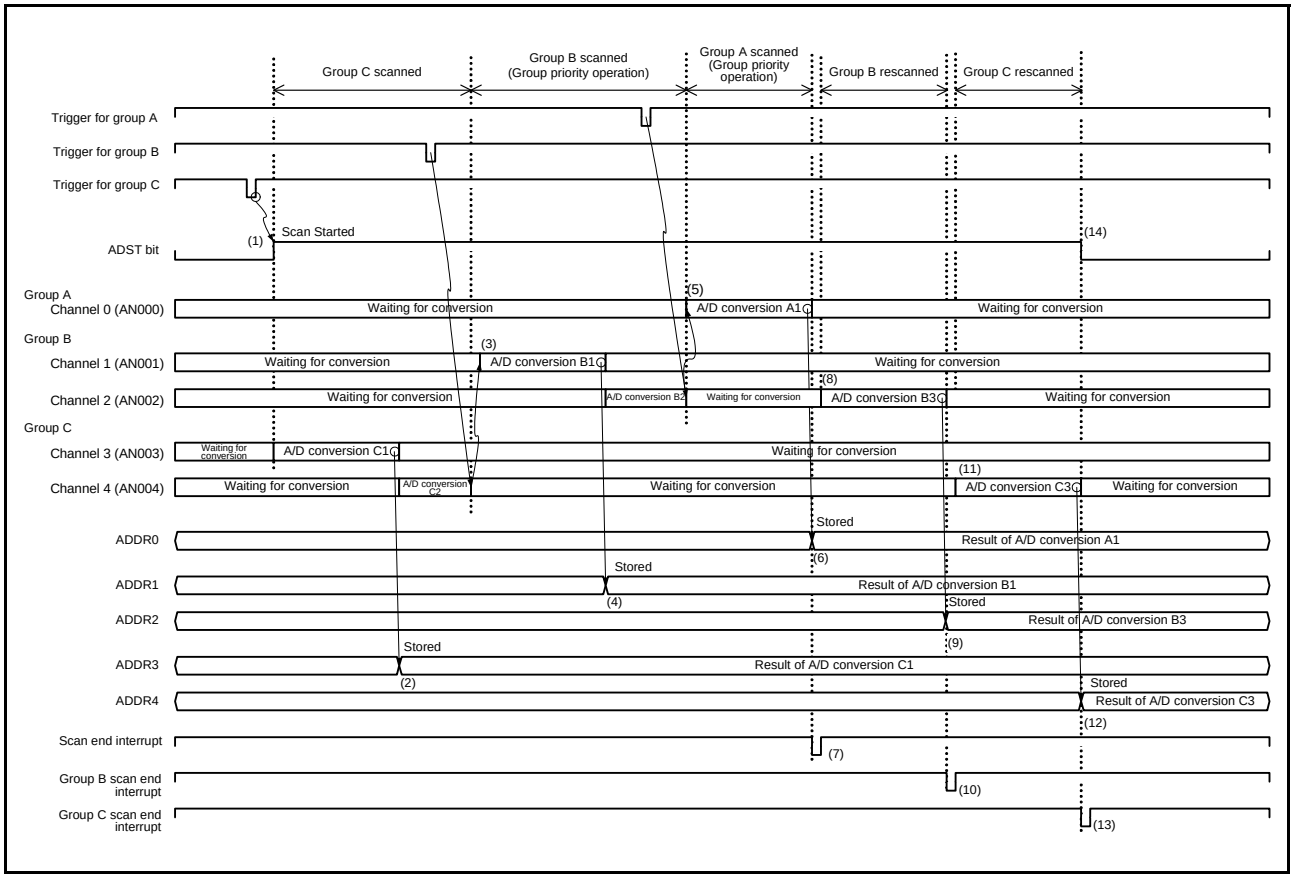


Figure 50.22 Example 1 of Group Priority Operation: Priority Group Trigger Input during Low-Priority Group Scan, With Rescan Setting (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 1)

Operation example 2: Priority group trigger input during low-priority group rescan, with rescan setting

Figure 50.23 shows an example when a group A trigger is input during rescan operation on group B.

If a trigger for the priority groups (groups A and B for group C and group A for group B) is input, scan for the priority group starts even while rescan operation on the low-priority group is in progress. After scan for the priority group is completed, scan for the low-priority group is restarted after having been discontinued.

Operations of the ADCSR.ADST bit, storing to the A/D conversion result in the A/D data register (ADDRy), and interrupt requests are the same as example 1.

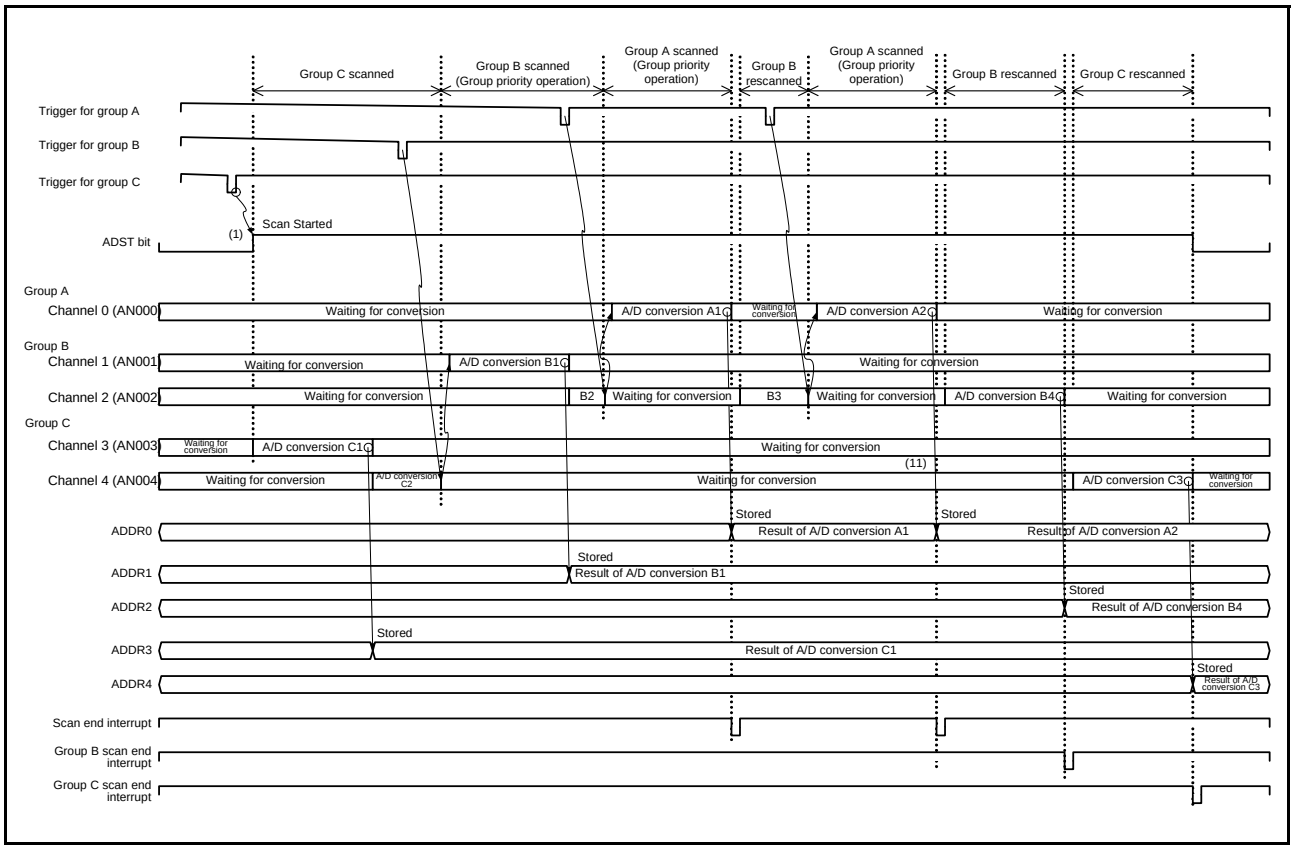


Figure 50.23 Example 2 of Group Priority Operation: Priority Group Trigger Input during Low-Priority Group Rescan, With Rescan Setting (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 1)

Operation example 3: Low-priority group trigger input during priority group scan, with rescan setting

The following describes an example when a trigger for the low-priority group is input during scan operation on the priority group when the ADGPSCR.GBRSCN bit is 1 (scan for the group is restarted after having been discontinued due to group priority operation). If the ADGPSCR.GBRSCN bit is 0, all triggers for the low-priority group that are input during scan operation on the priority group are disabled.

(1) When a group A trigger input sets the ADCSR.ADST bit to 1 (A/D conversion start), scan for the ANn channels of group A selected in the ADANSA0 register, starts from the channel with the smallest number n.
(2) If a group B trigger is input during scan for group A, scan for group B can be started.
(3) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(4) After scan for group A is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).
(5) After scan for group A is completed, scan for the ANn channels of group B selected in the ADANSB0 register, starts from the channel with the smallest number n while the ADCSR.ADST bit remains 1. If the ADGSPCR.LGRRS bit is set to 1 at this time, scan for group B starts from the channel on which A/D conversion was discontinued.
When a group A trigger is input during scan for group B, group A scan starts as in example 1, and group B scan starts after group A scan is completed.
(6) If a group C trigger is input during scan for group B, scan for group C can be started.
(7) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(8) After scan for group B is completed, a group B scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (interrupt generation upon group B scan completion enabled).
(9) After scan for group B is completed, scan for the ANn channels of group C selected in the ADANSC0 register, starts from the channel with the smallest number n while the ADCSR.ADST bit remains 1. If the ADGSPCR.LGRRS bit is set to 1 at this time, scan for group C starts from the channel on which A/D conversion was discontinued.
When a group A or B trigger is input during scan for group C, group A or B scan starts as in example 1, and group C scan starts after group A or B scan is completed.
(10) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(11) After scan for group C is completed, a group C scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (interrupt generation upon group C scan completion enabled).
(12) The ADCSR.ADST bit is automatically cleared to 0 when scan of all selected channels is completed, and the 12-bit A/D converter enters a wait state.

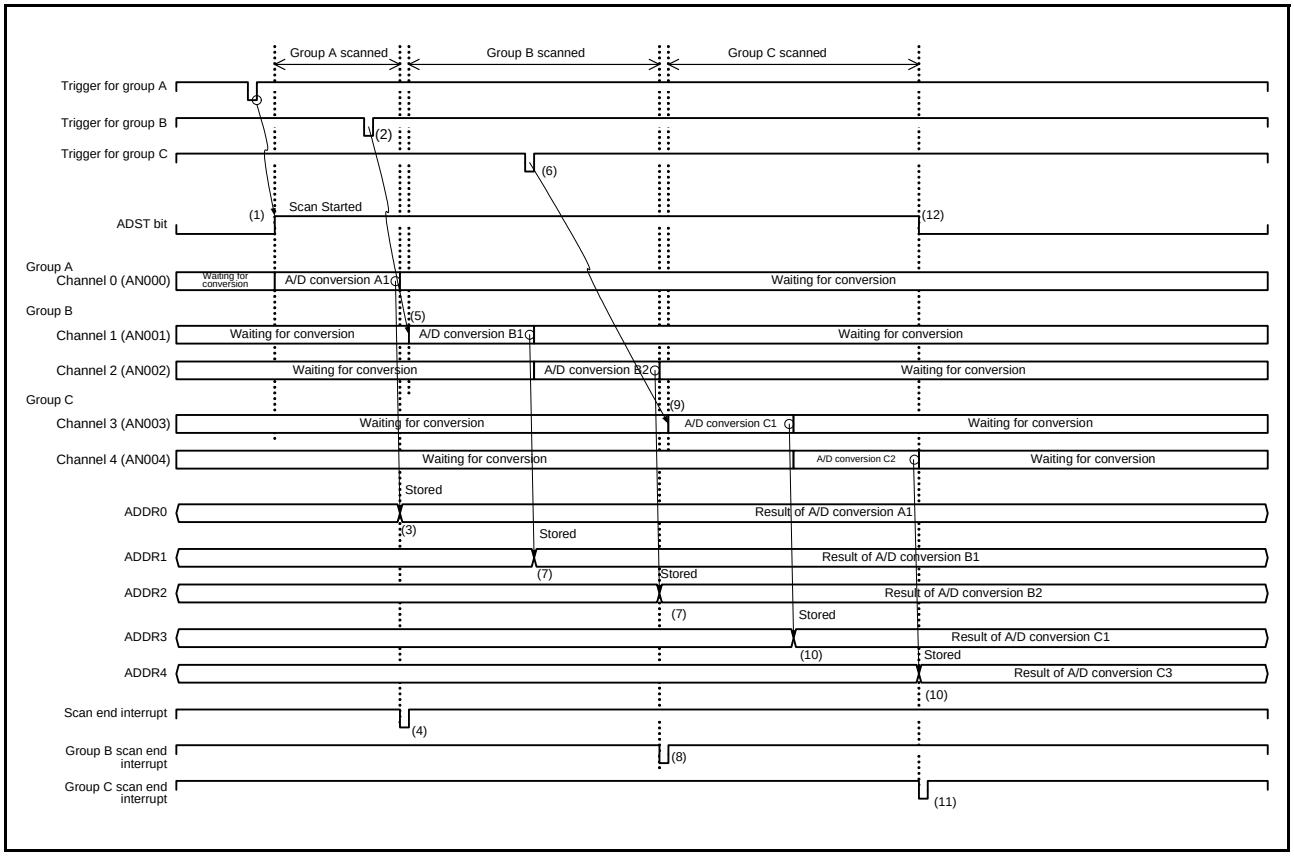


Figure 50.24 Example 3 of Group Priority Operation: Low-Priority Group Trigger Input during Priority Group Scan, With Rescan Setting (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 1, ADGCTRGR.GRCE = 1)

Operation example 4 shows group priority operation in group scan mode (ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0) when channel 0 is selected for group A, channels 1 and 2 are selected for group B, and channels 3 and 4 are selected for group C.

Operation example 4: Priority group trigger input during low-priority group scan, without rescan setting

(1) When a group C trigger input sets the ADCSR.ADST bit to 1 (A/D conversion start), scan for the ANn channels selected in the ADANSC0 register, starts from the channel with the smallest number n.

(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)

(3) If a group B trigger is input during scan for group C, group C scan is discontinued while the ADCSR.ADST bit remains 1, and scan for the ANn channels of group B selected in the ADANSB0 register, starts from the channel with the smallest number n. If A/D conversion is not completed when scan was discontinued, the result is not stored in the corresponding A/D data register (ADDRy).

(4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)

(5) If a group A trigger is input during scan for group B, group B scan is discontinued while the ADCSR.ADST bit remains 1, and scan for the ANn channels of group A selected in the ADANSB0 register, starts from the channel with the smallest number n. If A/D conversion is not completed when scan was discontinued, the result is not stored in the corresponding A/D data register (ADDRy).

(6) After scan for group A is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).

(7) The ADST bit is automatically cleared to 0 when scan for group A is completed, and the 12-bit A/D converter enters a wait state. Scan for groups C and B is not started until the next trigger corresponding to the group is input.

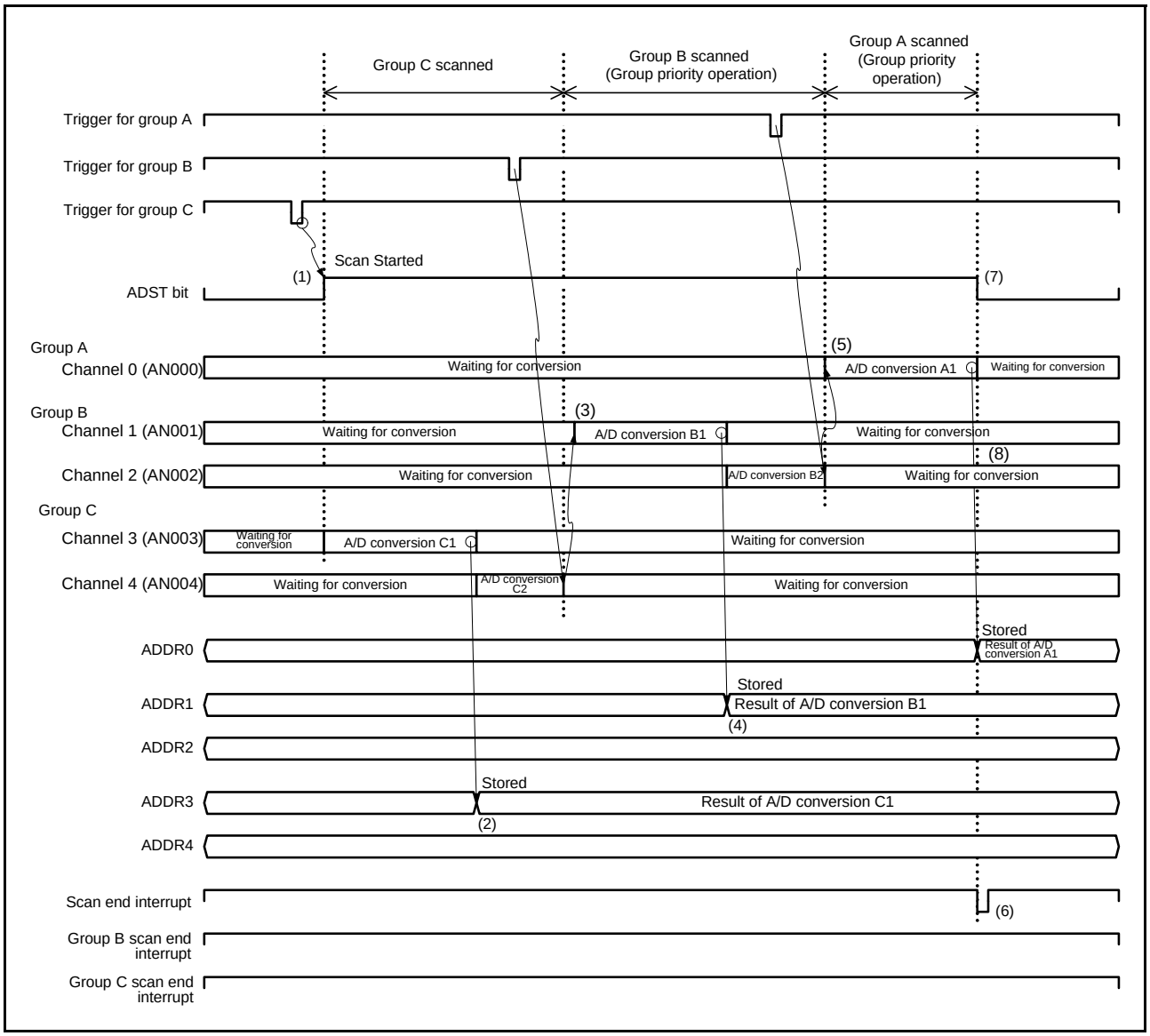


Figure 50.25 Example 4 of Group Priority Operation: Priority Group Trigger Input during Low-Priority Group Scan, Without Rescan Setting (ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 1)

Operation example 5 shows group priority operation in group scan mode (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 1, ADGSPCR.LGRRS = 1) when channel 0 is selected for group A, channel 1 is selected for group B, and channels 2 and 3 are selected for group C.
When the ADGCTRGR.GRCE bit is set to 1, single scan mode is continuously operated on group B and trigger input for group C is disabled.

Operation example 5: Continuous single scan operation on group C

(1) When setting ADGSPCR.GBRP to 1 sets the ADCSR.ADST bit to 1 (A/D conversion start), scan for the ANn channels selected in the ADANSC0 register, starts from the channel with the smallest number n.
(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(3) If a group B trigger is input during scan for group C, group C scan is discontinued while the ADCSR.ADST bit remains 1, and scan for the ANn channels of group B selected in the ADANSB0 register, starts from the channel with the smallest number n. If A/D conversion is not completed when scan was discontinued, the result is not stored in the corresponding A/D data register (ADDRy).
(4) If a group A trigger is input during scan for group B, group B scan is discontinued while the ADCSR.ADST bit remains 1, and scan for the ANn channels of group A selected in the ADANSA0 register, starts from the channel with the smallest number n. If A/D conversion is not completed when scan was discontinued, the result is not stored in the corresponding A/D data register (ADDRy).
(5) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(6) A scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).
(7) If the ADGSPCR.GBRSCN bit is 1 (scan for the group is restarted after having been discontinued due to group priority operation), scan for the ANn channels of group B selected in the ADANSB0 register, restarts from the channel with the smallest number n while the ADCSR.ADST bit remains 1. If the ADGSPCR.LGRRS bit is set to 1 at this time, scan for group B starts from the channel on which A/D conversion was discontinued.
(8) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(9) A group B scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (interrupt generation upon group B scan completion enabled).
(10) If the ADGSPCR.GBRSCN bit is 1 (scan for the group is restarted after having been discontinued due to group priority operation), scan for the ANn channels of group C selected in the ADANSC0 register, restarts from the channel with the smallest number n while the ADCSR.ADST bit remains 1. If the ADGSPCR.LGRRS bit is set to 1 at this time, scan for group C starts from the channel on which A/D conversion was discontinued.
(11) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(12) A group C scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (interrupt generation upon group C scan completion enabled).
(13) If the ADGSPCR.GBRP bit is set to 1 (single scan is continuously operated), scan for the ANn channels of group C selected in the ADANSC0 register, restarts from the channel with the smallest number n while the ADCSR.ADST bit remains 1.

To continuously operate single scan for group C, disable trigger input for group B.
Steps 13, 11, 12, and then 13 are repeated as long as the ADGSPCR.GBRP bit remains 1.
Clearing of the ADCSR.ADST bit to 0 is prohibited while the ADGSPCR.GBRP bit is set to 1.
To forcibly stop scanning when ADGSPCR.GBRP = 1, follow the procedures for clear operation by software through the ADCSR.ADST bit shown in section 50.6.2, Notes on Stopping A/D Conversion.

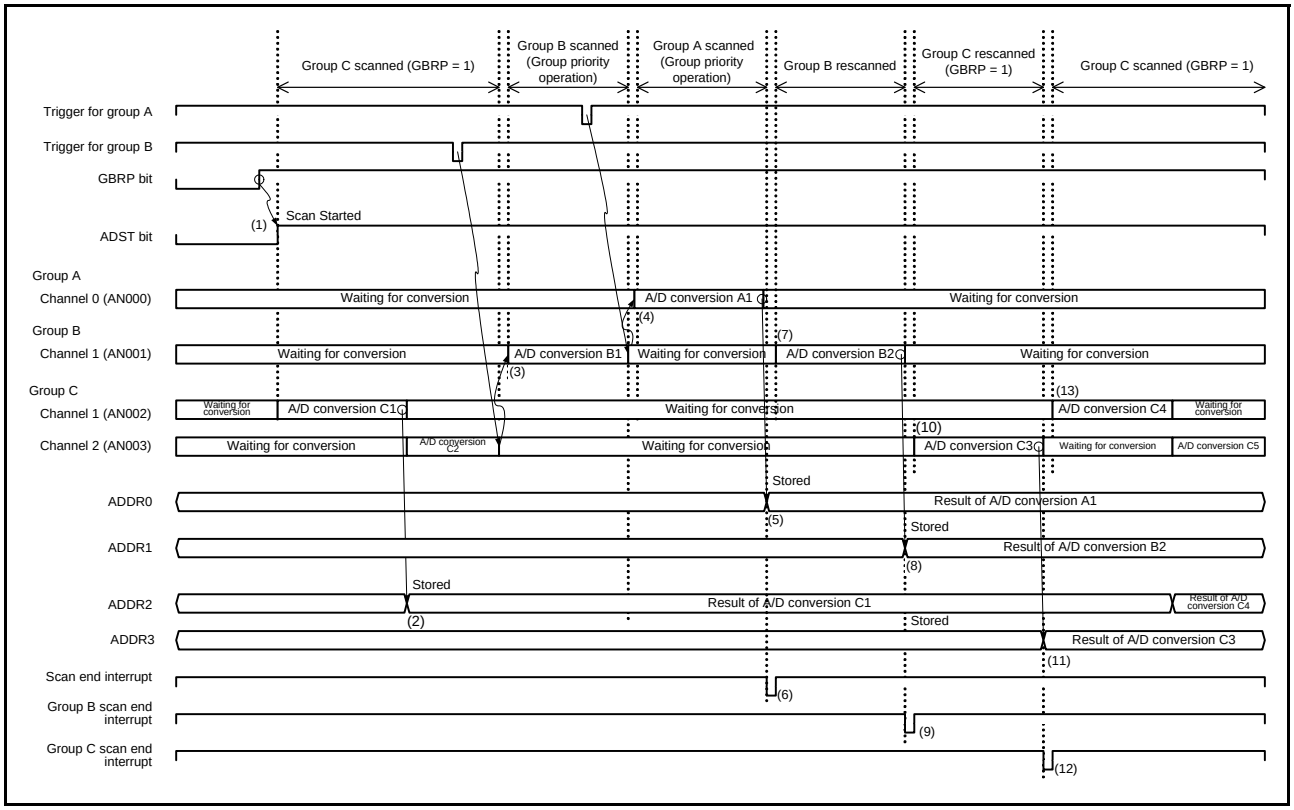


Figure 50.26 Example 5 for Group Priority Operation: Continuous Single Scan Operation on Group C (ADGSPCR.GBRP = 1, ADGSPCR.LGRRS = 1)

50.3.5 Extended Analog Input

Extended Analog Input (ANEX1) is used when an external operational amplifier is connected to the MCU and multiple analog values are A/D converted. When the extended analog input is selected, only AN100 to AN107 are selectable. Do not select AN108 to AN111, temperature sensor output, or internal reference voltage. Also, when the extended analog input is selected, self-diagnosis or disconnection detection assist function cannot be used.

50.3.5.1 Usage of ANEX1

To perform A/D conversion of multiple analog values through an operational amplifier, analog values are input by using analog input channels of the MCU (AN100 to AN107), the time-divided analog values are retrieved, and an operational amplifier is connected between ANEX0 and ANEX1.
When ANEX1 is connected, set the ADEXICR.EXSEL[1:0] bits to 01b (ANEX1 selected) and ADEXICR.EXOEN bit to 1 (ANEX0: output enabled) and then select single scan mode or sequence scan mode.
Do not set in group scan mode. Figure 50.27 shows an example of structure of the extended analog input when ANEX1 is in use, and Figure 50.28 shows operation when three channels (AN100, AN101, AN102) are selected and single scan mode is selected.

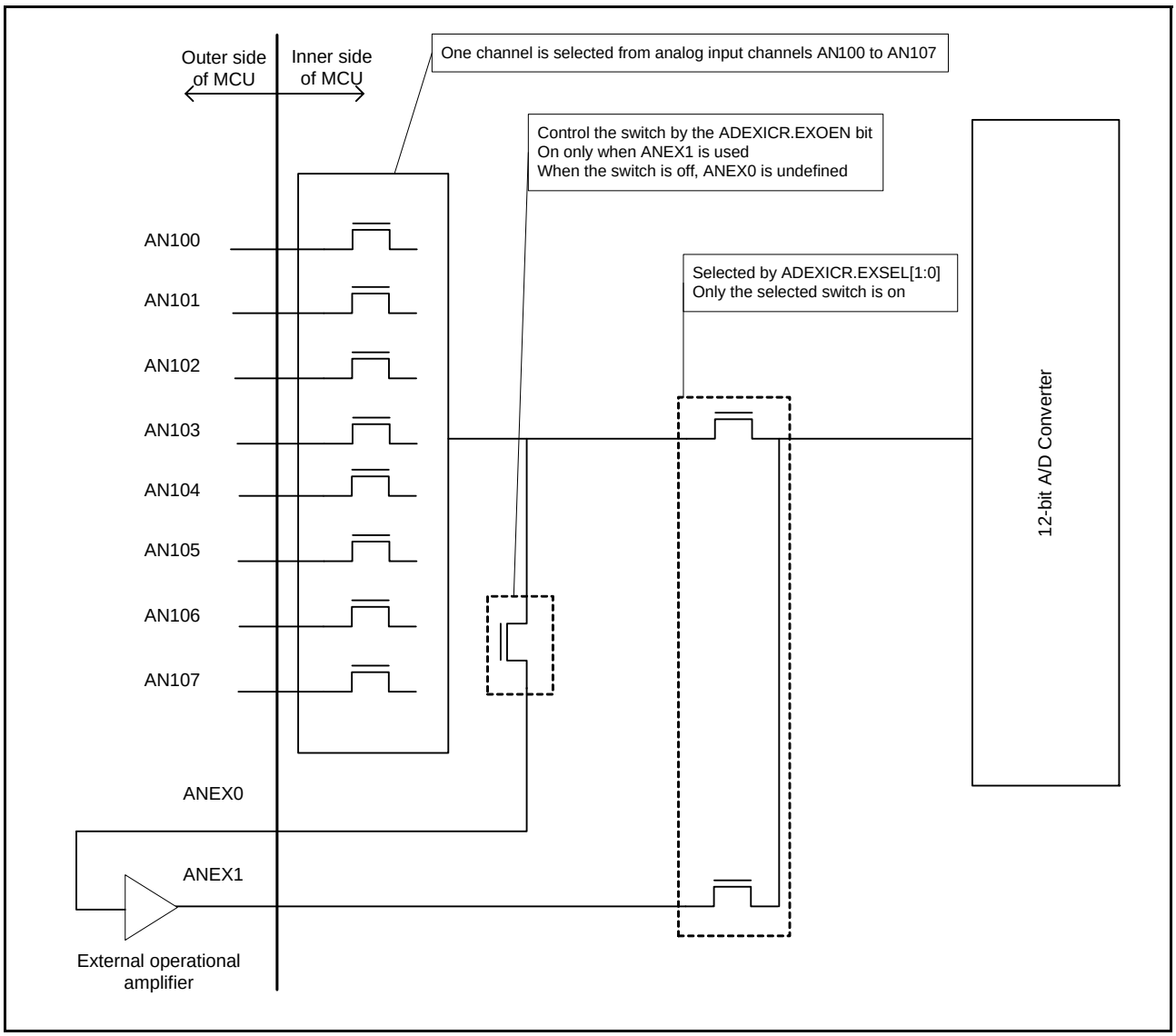


Figure 50.27 Example of Configuration of Extended Analog Input When ANEX1 is Used

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, or synchronous or asynchronous trigger input, A/D conversion is performed for the selected channels starting from the channel with the smallest number.
(2) Each time A/D conversion of a single channels is completed, the A/D converted value is stored in the corresponding A/D data register (ADDRy).
(3) After completion of A/D conversion of all selected channels, if the ADCSR.ADIE bit is set to 1 (when interrupt generation after scanning is enabled), a scan end interrupt request is generated.
(4) The ADCSR.ADST bit holds 1 during A/D conversion and is automatically cleared upon completion of A/D conversion of all selected channels, and the 12-bit A/D converter enters into a wait state.

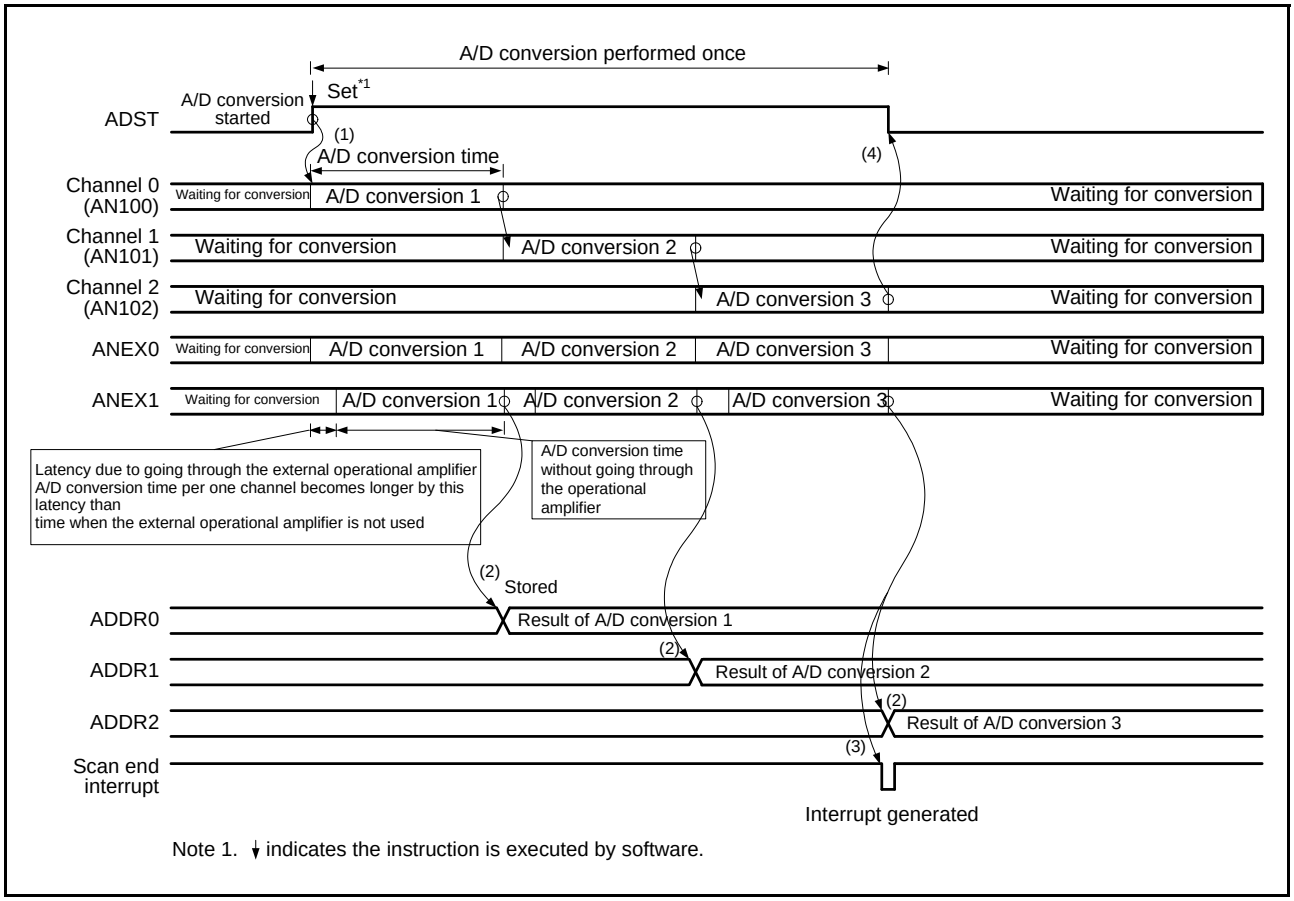


Figure 50.28 Example of Operation of ANEX1 Input (Single Scan Mode)

When the extended analog input is selected, each period for A/D conversion will be relatively longer than that of the case when analog input channels are directly A/D converted due to latency of the operational amplifier.

50.3.6 Comparison Function (Windows A/B)

50.3.6.1 Comparison Function Windows A/B

The comparison function compares the reference values set in registers ADCMPDR0, ADCMPDR1, ADWINLLB, and ADWINULB with the A/D conversion results. When this function is used, self-diagnosis or double trigger mode cannot be used. A window comparison function can also be used (when ADCMPCR.WCMPE = 1), enabling comparison of two values. Two sets of voltage level ranges can be set in the window comparison function, one for window A and one for window B.

Operation when window comparison is enabled (ADCMPCR.WCMPE = 1) in combination with continuous scan mode is described below.

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, or synchronous or asynchronous trigger input, A/D conversion is performed for selected channels, temperature sensor output, and internal reference voltage in this order.

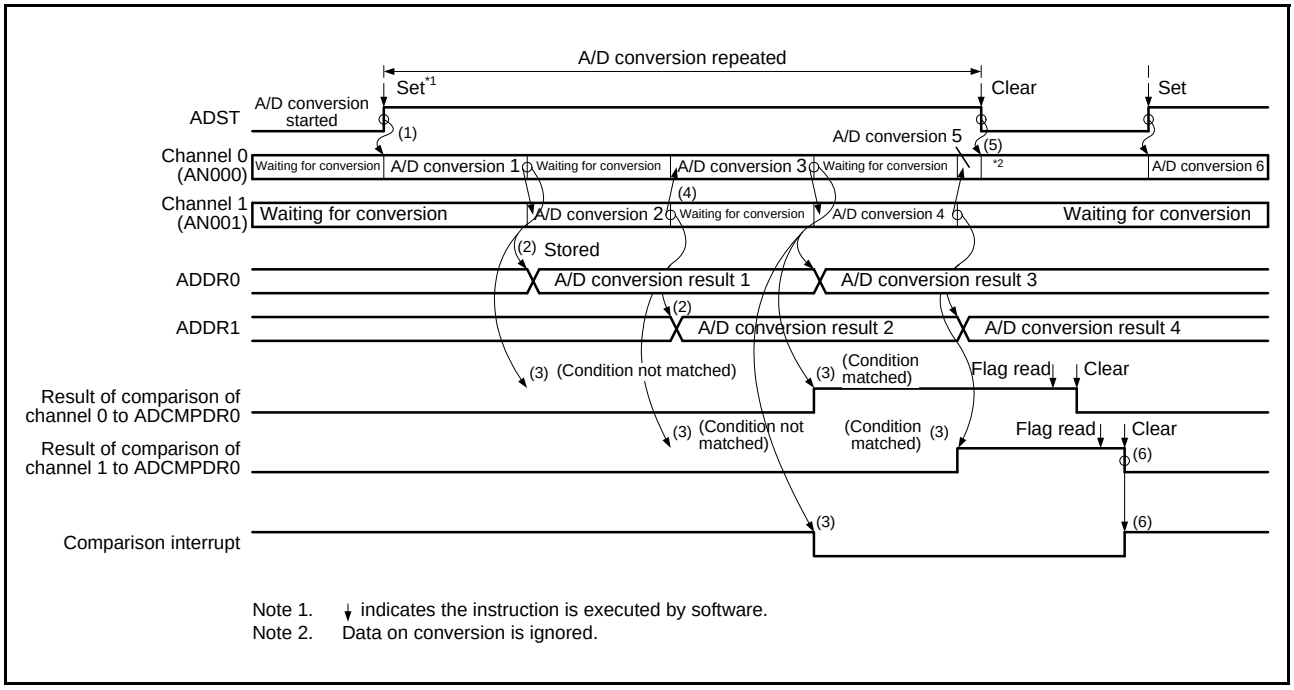
(2) Each time A/D conversion of a single channels is completed, the A/D converted value is stored in the corresponding A/D data register (ADDRy, ADTSDR, or ADOCDR). When ADCMPCR.CMPAE = 1 and window A is selected in the ADCMPANSRy and ADCMPANSER registers, the A/D converted value is compared with the setting values of the ADCMPDR0 and ADCMPDR1 registers.
When ADCMPCR.CMPBE = 1 and window B is selected in the ADCMPBNSR register, the A/D converted value is compare with the setting values of the ADWINULB and ADWINLLB registers.

(3) Upon a comparison match with the conditions specified in the ADCMPLR0 and ADCMPLER registers, the flags of the comparison window A (ADCMPSR0.CMPSTCHA0n, ADCMPSER.CMPFTS, and ADCMPSER.CMPFOC) are set to 1 for window A.
In this case, if the ADCMPCR.CMPAIE bit is set to 1, an interrupt request S12CMPAI is generated.
Similarly, for window B, upon a match with the conditions set in the ADCMPBNSR.CMPLB register, the comparison window B flag (ADCMPBSR.CMPSTB) is set to 1. In this case, if the ADCMPCR.CMPBIE bit is set to 1, an interrupt request S12CMPBI is generated.

(4) When all selected A/D conversion and comparison are completed, scanning is performed again.

(5) Set the ADCSR.ADST to 0 (A/D conversion stop) and execute the processing for the channels in which the comparison flag is 1.

(6) After the processing above, clear all comparison flags. When comparison is re-executed, start A/D conversion again.



Note 1. ↓ indicates the instruction is executed by software.
Note 2. Data on conversion is ignored.

Figure 50.29 Example of Operation of Comparison (AN000 and AN001 for Targets for Comparison)

50.3.6.2 Restrictions on Comparison Function

The Comparison function has the following restrictions.

1. Use of self-diagnosis and double trigger mode is prohibited.
   (ADRD and ADDBLDR are not targeted for the comparison function)
2. When temperature sensor or internal reference voltage is selected for window A, operation of window B is prohibited.
3. When temperature sensor or internal reference voltage is selected for window B, operation of window A is prohibited.
4. The same channel cannot be set for window A and window B.
5. Satisfy the condition of [reference value (high side) ≥ reference value (low side)].

50.3.7 Analog Input Sampling Time and Scan Conversion Time

Figure 50.30 shows the scan conversion timing in single scan mode, in which scan conversion is activated by software or a synchronous trigger. Figure 50.31 shows the scan conversion timing in single scan mode, in which scan conversion is activated by an asynchronous trigger. The scan conversion time (t_{SCAN}) includes the start-of-scanning-delay time (t_D), disconnection detection assistance processing time (t_{DIS})*1, self-diagnosis A/D conversion processing time (t_{DIAG})*2, A/D conversion processing time (t_{CONV}), and end-of-scanning-delay time (t_{ED}).
The A/D conversion processing time (t_{CONV}) consists of sampling time (t_{SPL}) and time for conversion by successive approximation (t_{SAM}). The sampling time (t_{SPL}) is used to charge sample-and-hold circuits in the A/D converter. If there is not sufficient sampling time due to the high impedance of an analog input signal source, or if the A/D conversion clock (ADCLK) is slow, sampling time can be adjusted using the ADSSTR register.
The time for conversion by successive approximation (t_{SAM}) for unit 0 is at 13 states (ADCLK) with 12-bit resolution selected, 11 states (ADCLK) with 10-bit resolution selected, and 9 states (ADCLK) with 8-bit resolution selected; and for unit 1, 15 states (ADCLK) with 12-bit resolution selected, 13 states (ADCLK) with 10-bit resolution selected, and 11 states (ADCLK) with 8-bit resolution selected. Table 50.18 shows the scan conversion time.

The scan conversion time (t_{SCAN}) in single scan mode for which the number of selected channels is n can be determined as follows:

$$t_{SCAN} = t_D + (t_{DIS} \times n) + t_{DIAG} + (t_{CONV} \times n)^{*3} + t_{ED}$$

The scan conversion time for the first cycle in continuous scan mode is t_{SCAN} for single scan minus t_{ED}.
The scan conversion time for the second and subsequent cycles in continuous scan mode is fixed to $(t_{DIS} \times n) + t_{DIAG} + t_{DSD} + (t_{CONV} \times n)^{*3}$.

Note 1. When disconnection detection assistance is not selected, $t_{DIS} = 0$.
Note 2. When the self-diagnosis function is not used, $t_{DIAG} = 0$, $t_{DSD} = 0$.
Note 3. $t_{CONV} \times n$ when the sampling time (t_{SPL}) of selected channels is the same, but it is the total of the sampling time of each channel and time for conversion by successive approximation (t_{SAM}).

Table 50.18 Times for Conversion during Scanning (in Numbers of Cycles of ADCLK and PCLK)

Item			Symbol		Type/Conditions: Synchronous Trigger (MTU)	Type/Conditions: Synchronous Trigger (TMR, TPU, ELC)	Type/Conditions: Asynchronous Trigger	Type/Conditions: Software Trigger	Unit
Scan start processing time*1, *2	A/D conversion on group under group priority control.	The low-priority group is to be stopped. (The priority group is activated after low-priority group B is stopped due to an A/D conversion source of the priority group.)	t_D		1 PCLKA + 4 PCLKB + 6 ADCLK	2PCLKB + 6 ADCLK	—	—	Cycle
		The low-priority group is not to be stopped. (Activation by an A/D conversion source of the priority group.)			1 PCLKA + 3 PCLKB + 4 ADCLK	2 PCLKB + 4 ADCLK	—	—	
	A/D conversion when self-diagnosis is enabled	A/D conversion for self-diagnosis is to be started.			1 PCLKA + 3 PCLKB + 6 ADCLK	2 PCLKB + 6 ADCLK	4 PCLKB+ 6 ADCLK	6 ADCLK	
	Other than above				1 PCLKA + 3 PCLKB + 4 ADCLK	2 PCLKB + 4 ADCLK	2 PCLKB + 4 ADCLK	4 ADCLK	
Disconnection detection assistance processing time			t_{DIS}		The setting of ADNDIS[3:0] (initial value = 00h) × ADCLK				
Self-diagnosis conversion processing time*1	Sampling time		t_{DIAG}	t_{SPL}	The setting of ADSSTR0 (initial value = 0Bh) × ADCLK				
	Time for conversion by successive approximation			t_{SAM}	Unit 0		Unit 1		
					13 ADCLK (12-bit conversion resolution) 11 ADCLK (10-bit conversion resolution) 9 ADCLK (8-bit conversion resolution)		15 ADCLK (12-bit conversion resolution) 13 ADCLK (10-bit conversion resolution) 11 ADCLK (8-bit conversion resolution)		
	Normal A/D conversion is to be started after completion of self-diagnosis conversion.		t_{DED}		2 ADCLK				
	A/D conversion for self-diagnosis is to be started after completion of conversion for continuous scan on the last channel specified.		t_{DSD}		2 ADCLK				
A/D conversion processing time*1	Sampling time		t_{CONV}	t_{SPL}	The setting of ADSSTRn (n = 0 to 11, T, O) (initial value = 0Bh) × ADCLK				
	Time for conversion by successive approximation			t_{SAM}	Unit 0		Unit 1		
					13 ADCLK (12-bit conversion resolution) 11 ADCLK (10-bit conversion resolution) 9 ADCLK (8-bit conversion resolution)		15 ADCLK (12-bit conversion resolution) 13 ADCLK (10-bit conversion resolution) 11 ADCLK (8-bit conversion resolution)		
Scan end processing time*1			t_{ED}		1 PCLKB + 3 ADCLK				

Note 1. For t_D, t_{DIAG}, t_{CONV}, and t_{ED}, see Figure 50.30 and Figure 50.31.
Note 2. This is the maximum time required from software writing or trigger input to A/D conversion start.

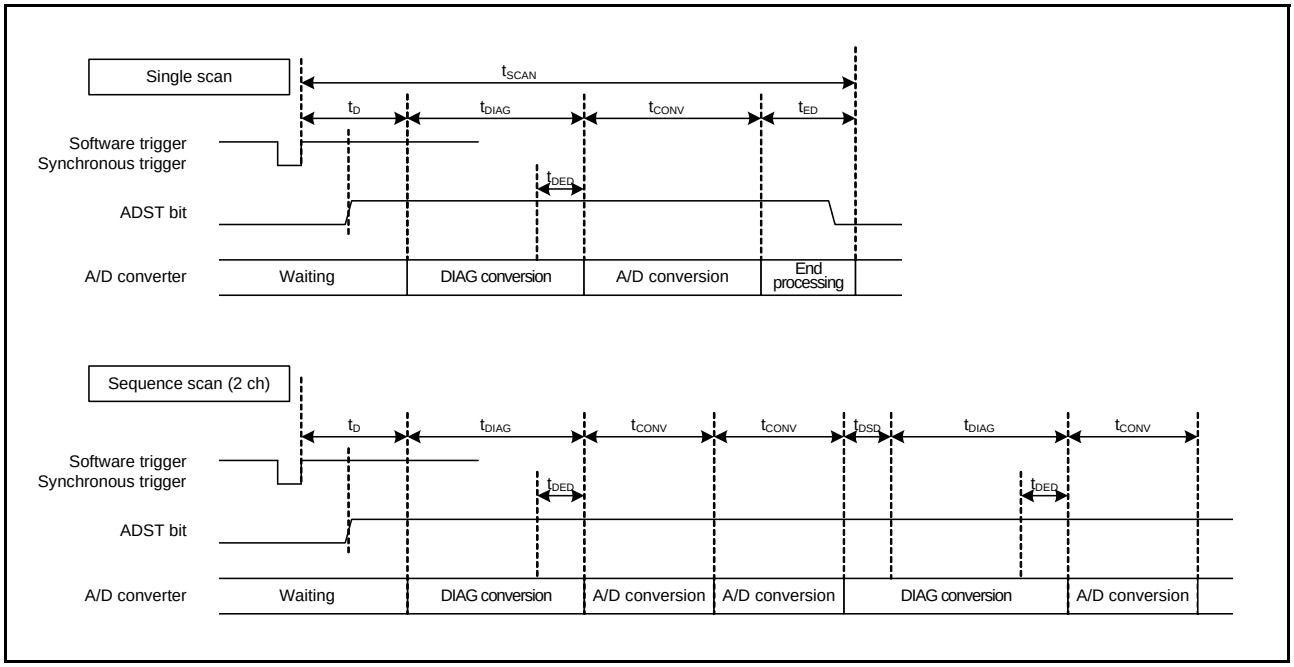


Figure 50.30 Scan Conversion Timing (Activated by Software or Synchronous Trigger)

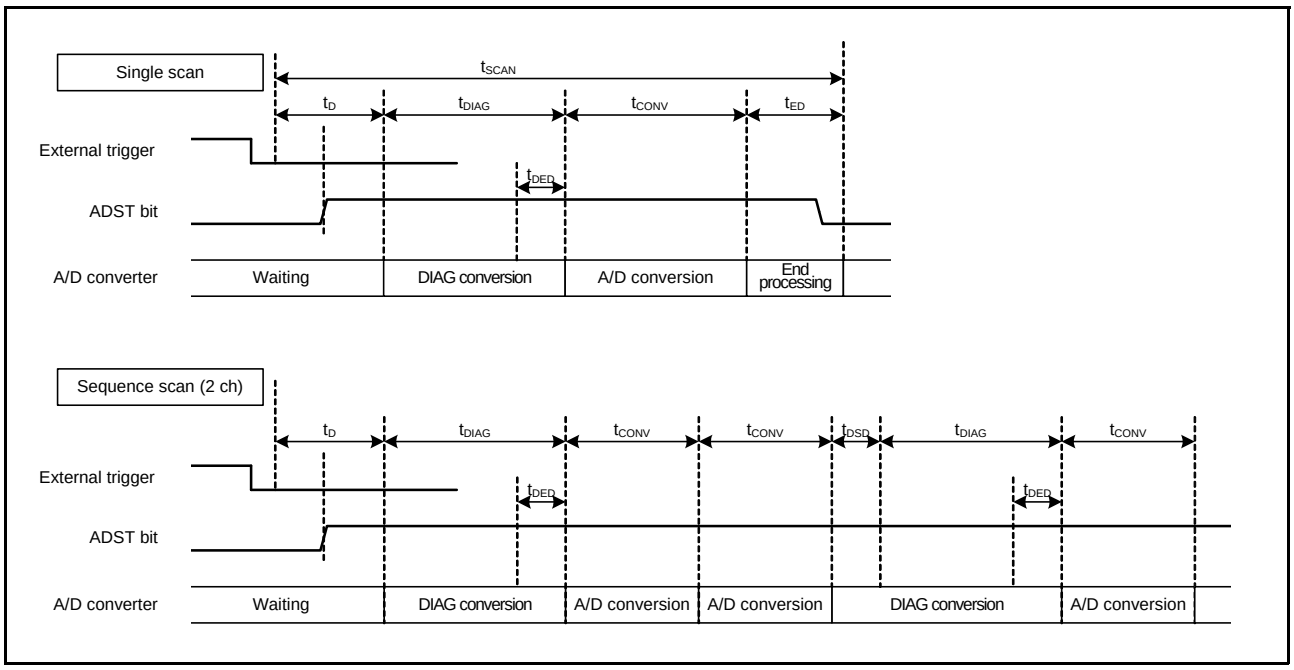


Figure 50.31 Scan Conversion Timing (Activated by Asynchronous Trigger)

50.3.7.1 Timing of Suspension and Starting of Scanning in Operation under Group Priority Control

The timings for suspension and starting of scanning in operation under group priority control that must considered are listed below.

1. The timing for suspending a scan of a group with a lower-priority and the timing for starting a scan of a group with a higher-priority.
2. The time at which scanning by the group with a lower-priority is resumed on completion of scanning by the higher-priority group when the trigger for scanning by the lower-priority group is accepted during scanning by the higher-priority group.
3. The timing for performing sequential single scans by a lower-priority group.

Figure 50.32 shows the timing diagram of each of the above cases.
The times at which scanning by group A is completed and scanning by group C resumes or scanning by group B is completed and scanning by group C resumes are the same as those for group A and group B in Figure 50.32.
The timing for consecutive single scans is the same for group B and group C.

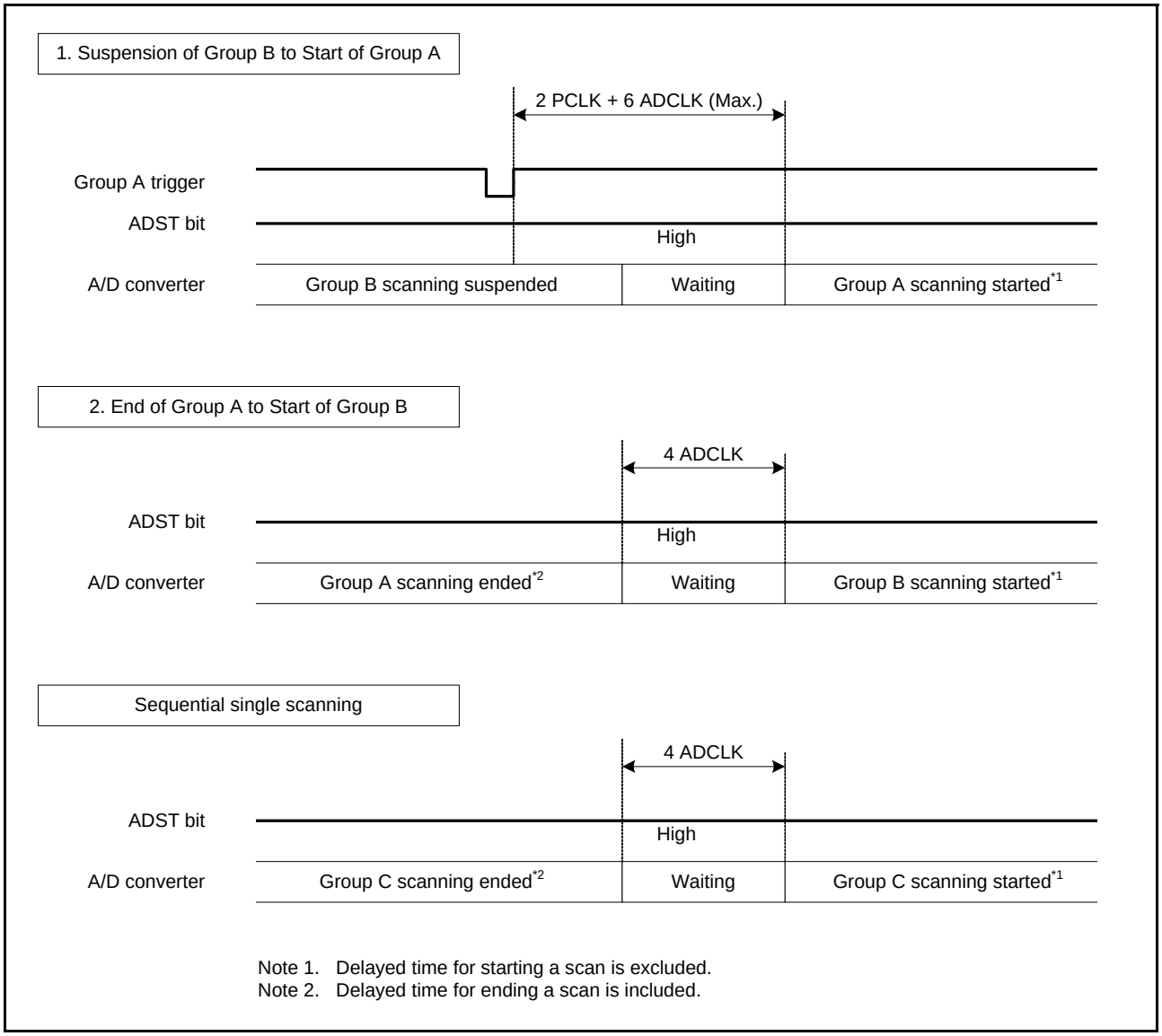


Figure 50.32 Timing of Stop/Start of Scanning in Group Priority Mode

50.3.8 Usage Example of A/D Data Register Automatic Clearing Function

Setting the ADCER.ACE bit to 1 automatically clears the A/D data registers (ADDRy, ADRD, ADTSDR, ADOCDR, ADDBLDR, ADDBLDRA, ADDBLDRB) to 0000h when the A/D data registers (ADDRy, ADRD, ADTSDR, ADOCDR, ADDBLDR, ADDBLDRA, ADDBLDRB) are read by the CPU, DTC, or DMAC.
This function enables detection of update failures of the A/D data registers (ADDRy, ADRD, ADTSDR, ADOCDR, ADDBLDR, ADDBLDRA, ADDBLDRB). The following describes the examples in which the function to automatically clear the ADDRy register is enabled and disabled.
In a case where the ADCER.ACE bit is 0 (automatic clearing disabled), if the A/D conversion result (0222h) is not written to the ADDRy register for some reason, the old data (0111h) will be the ADDRy value. Furthermore, if this ADDRy value is read into a general register using an A/D scan end interrupt, the old data (0111h) can be saved in the general register. When checking whether there is an update failure, it is necessary to frequently save the old data in the RAM or a general register.
In a case where the ADCER.ACE bit is 1 (automatic clearing 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 0222h cannot be transferred to ADDRy for some reason, the cleared data (0000h) remains as the ADDRy value. If this ADDRy value is read into a general register using an A/D scan end interrupt at this point, 0000h will be saved in the general register. Occurrence of an ADDRy update failure can be determined by simply checking that the read data value is 0000h.

50.3.9 A/D-Converted Value Addition/Average Mode

In A/D-converted value addition mode, the same channel is A/D-converted 2, 3, 4, or 16 consecutive times and the sum of the converted values is stored in the data register. In A/D-converted value average mode, the same channel is A/D-converted two or four consecutive times and the mean of the converted values is stored in the data register. The use of the average of these results can improve the accuracy of A/D conversion, depending on the types of noise components that are present. This function, however, cannot always guarantee an improvement in A/D conversion accuracy.
The A/D-converted value addition/average mode can be used when A/D conversion of the channel select analog input, temperature sensor output (for S12AD1 only), or internal reference voltage (for S12AD1 only) is selected.

50.3.10 Disconnection Detection Assist Function

This converter incorporates the function to fix the charge for sampling capacitance to the specified state before start of A/D conversion. This function enables disconnection detection in wiring of analog inputs.
Figure 50.33 illustrates the A/D conversion operation when the disconnection detection assist function is used. Figure 50.34 shows an example of disconnection detection when precharge is selected. Figure 50.35 shows an example of disconnection detection when discharge is selected.

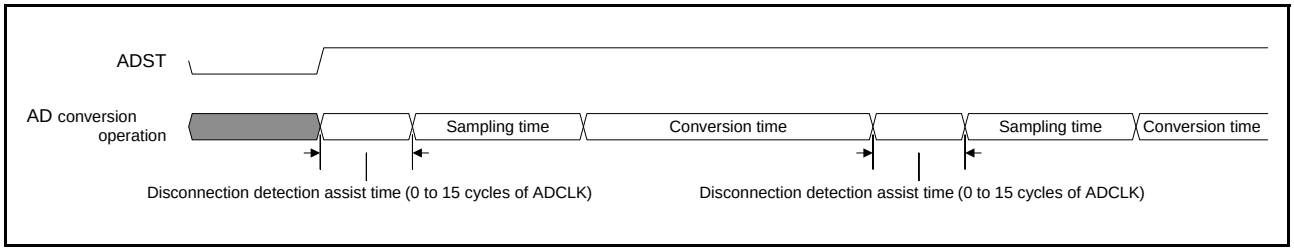
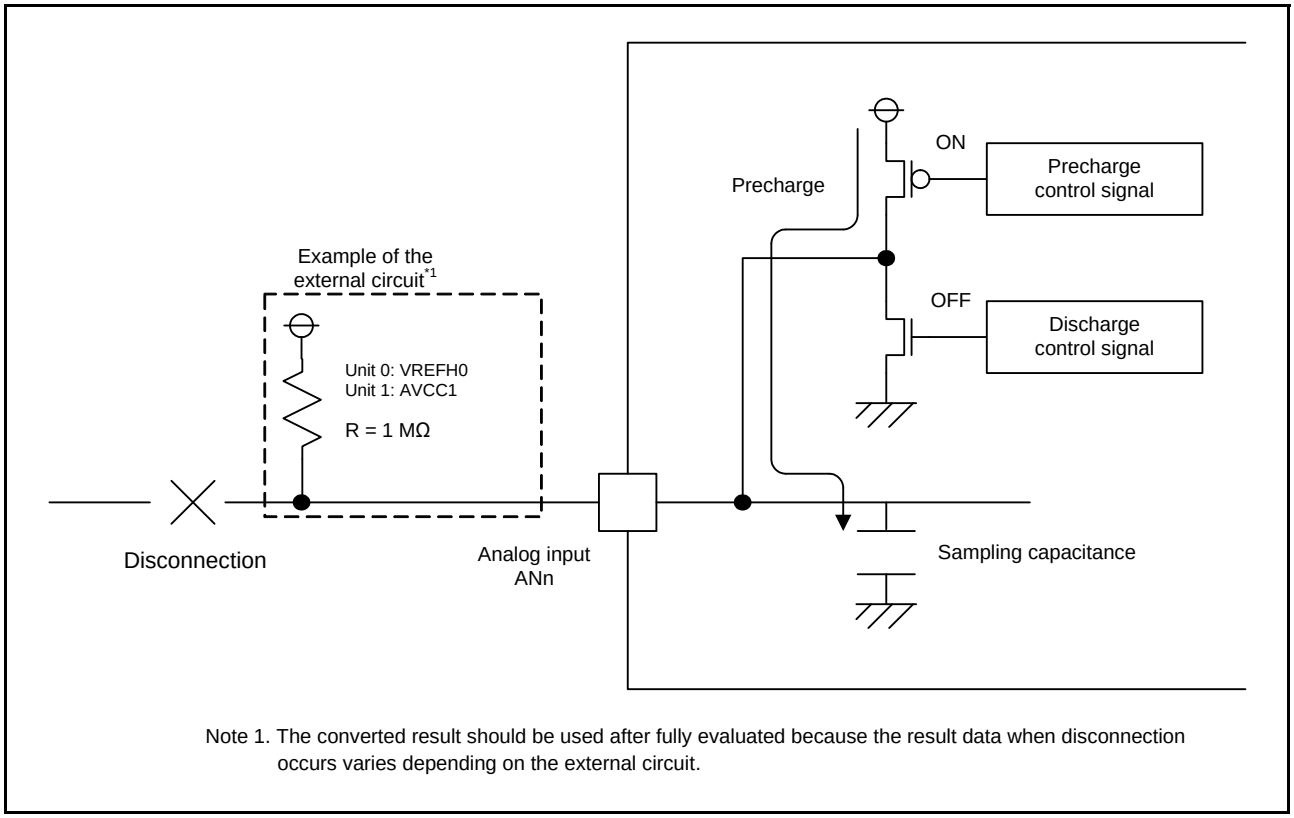
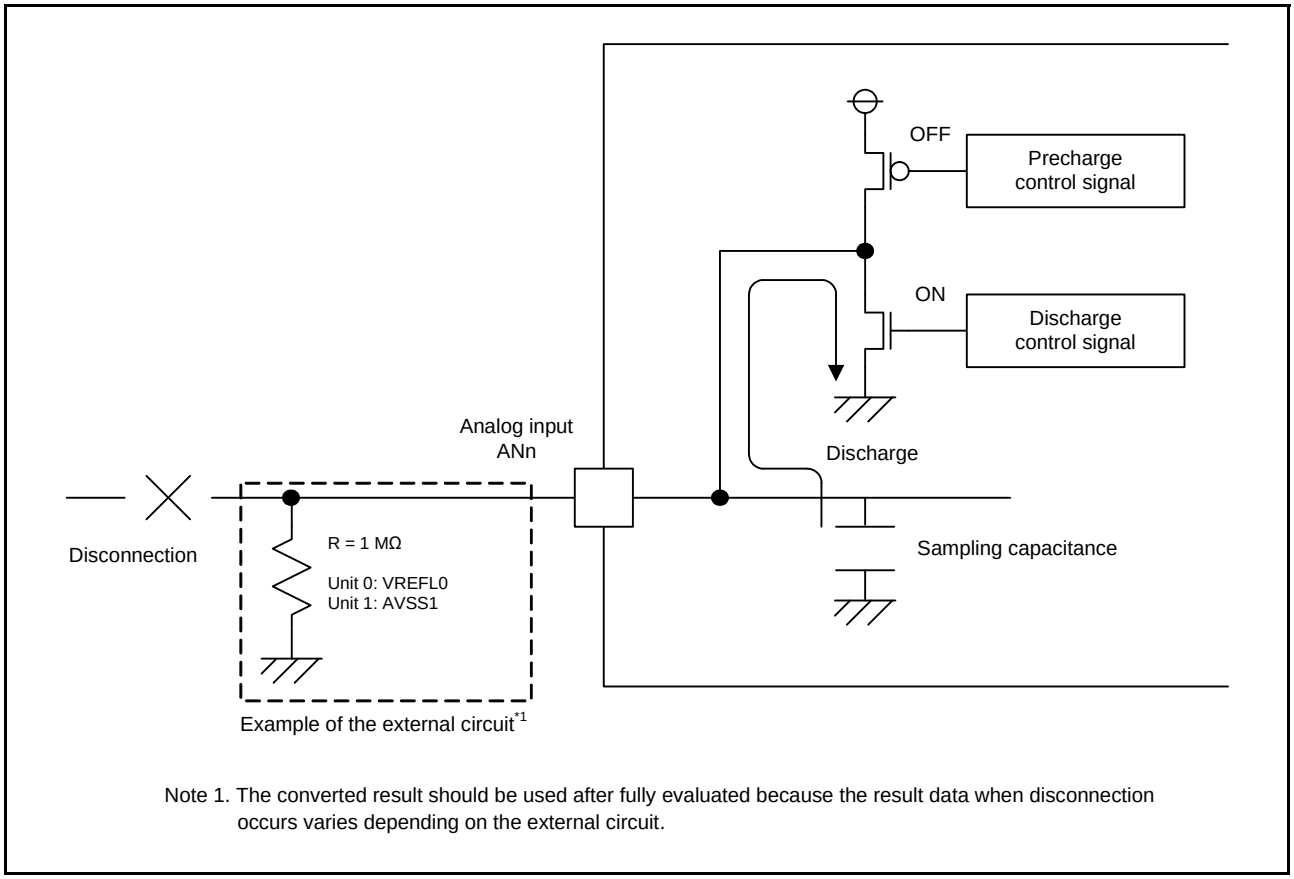


Figure 50.33 Operation of A/D Conversion When the Disconnection Detection Assist Function is Used



Note 1. The converted result should be used after fully evaluated because the result data when disconnection occurs varies depending on the external circuit.

Figure 50.34 Example of Disconnection Detection When Precharge is Selected



Note 1. The converted result should be used after fully evaluated because the result data when disconnection occurs varies depending on the external circuit.

Figure 50.35 Example of Disconnection Detection When Discharge is Selected

50.3.11 Starting A/D Conversion with Asynchronous Trigger

The A/D conversion can be started by the input of an asynchronous trigger. To start up the A/D converter by an asynchronous trigger, the A/D conversion start trigger select bits (ADSTRGR.TRSA[5:0]) should be set to 000000b, and a high-level signal should be input to the asynchronous trigger (ADTRG0# pin). Then, the ADCSR.TRGE and ADCSR.EXTRG bits should be set to 1. Figure 50.36 shows a timing of the asynchronous trigger input.
For the time from when the ADST bit is set to 1 until conversion starts, refer to section 50.6.3, A/D Conversion Restarting Timing and Termination Timing. An asynchronous trigger cannot be selected for group B or group C in group scan mode.

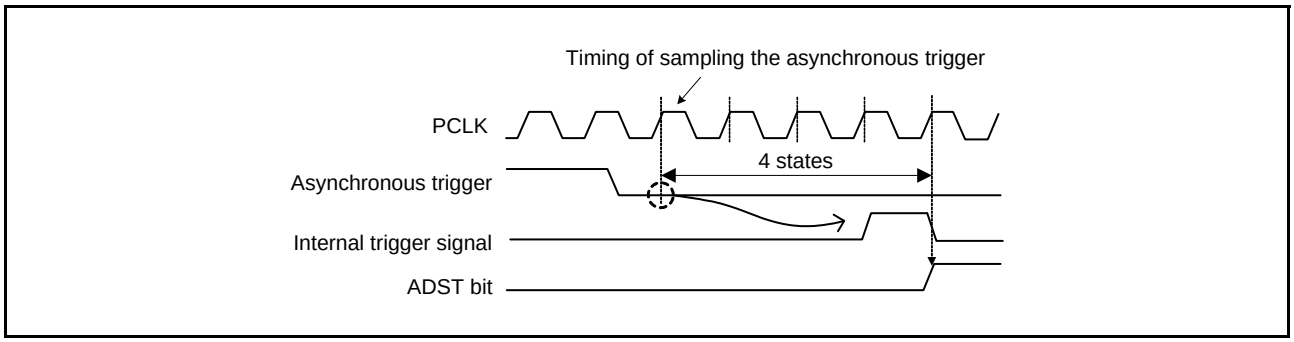


Figure 50.36 Timing of Sampling Asynchronous Trigger

50.3.12 Starting A/D Conversion with Synchronous Trigger from Peripheral Module

The A/D conversion can be started by a synchronous trigger. To start the A/D conversion by a synchronous trigger, the ADCSR.TRGE bit should be set to 1, the ADCSR.EXTRG bit should be cleared to 0, and the relevant sources should be selected by the ADSTRGR.TRSA[5:0] and ADSTRGR.TRSB[5:0] bits.

50.4 Interrupt Sources and DTC/DMAC Transfer Requests

50.4.1 Interrupt Requests

The 12-bit A/D converter can send scan end interrupt requests S12ADI/S12ADI1, S12GBADI/S12GBADI1, and S12GCADI/S12GCADI1 to the CPU.
This module can also generate interrupt requests S12CMPAI/S12CMPAI1 and S12CMPBI/S12CMPBI1 to the CPU in response to matches with a condition for comparison.
Setting the ADCSR.ADIE bit to 1 and 0 enables and disables an S12ADI/S12ADI1 interrupt, respectively; similarly, setting the ADCSR.GBADIE bit to 1 and 0 enables and disables a S12GBADI/S12GBADI1 interrupt, respectively; similarly, setting the ADGCTRGR.GCADIE bit to 1 and 0 enables and disables a S12GCADI/S12GCADI1 interrupt, respectively.
Setting the ADCMPCR.CMPAIE bit to 1 enables an S12CMPAI interrupt and setting the ADCMPCR.CMPAIE bit to 0 disables an S12CMPAI interrupt.
Setting the ADCMPCR.CMPBIE bit to 1 enables an S12CMPBI interrupt and setting the ADCMPCR.CMPBIE bit to 0 disables an S12CMPBI interrupt.
In addition, the DTC or DMAC can be activated when an S12ADI/S12ADI1, a S12GBADI/S12GBADI1, or S12GCADI/S12GCADI1 interrupt is generated. Using an S12ADI/S12ADI1, a S12GBADI/S12GBADI1, or S12GCADI/S12GCADI1 interrupt to allow the DTC or DMAC to read the converted data enables sequence conversion without burden on software.
For details on DTC and DMAC settings, see section 20, Data Transfer Controller (DTCb) and section 18, DMA Controller (DMACAb).

50.4.2 Scan Complete Event Output to ELC

The ELC can set up linked operation of a module specified in advance by using the S12ADI or S12ADI1 interrupt request signal as an event signal.
The S12GBADI or S12GBADI1 interrupt, S12GCADI or S12GCADI1 interrupt, S12CMPAI or S12CMPAI1 interrupt, and S12CMPBI or S12CMPBI1 interrupt request signals cannot be used as event signals. An event signal is output regardless of the setting of the corresponding interrupt request enable bit. The 12-bit A/D converter outputs the A/D conversion completed signals as event signals.

50.5 Allowable Impedance of Signal Source

To achieve high-speed conversion of 0.48 µs, the analog input pins of this MCU are designed so that the conversion accuracy is guaranteed if the impedance of the input signal source is 1.0 kΩ or less. If an external capacitor of large capacitance is attached in the application in which only a single pin input is converted in single scan mode, the only load on input is virtually 1.2 kΩ of the internal input resistor; therefore, the impedance of the signal source can be ignored. Being a low-pass filter, however, an analog input circuit may not follow the analog signal with a large differential coefficient. When high-speed analog signals are to be converted or multiple pins are to be converted in scan mode, a low-impedance buffer should be used.
Figure 50.37 shows an equivalent circuit of an analog input pin and an external sensor.
To perform A/D conversion accurately, charging of the internal capacitor C shown in Figure 50.37 must be completed within the specified period of time. This specified period is referred to as sampling time.

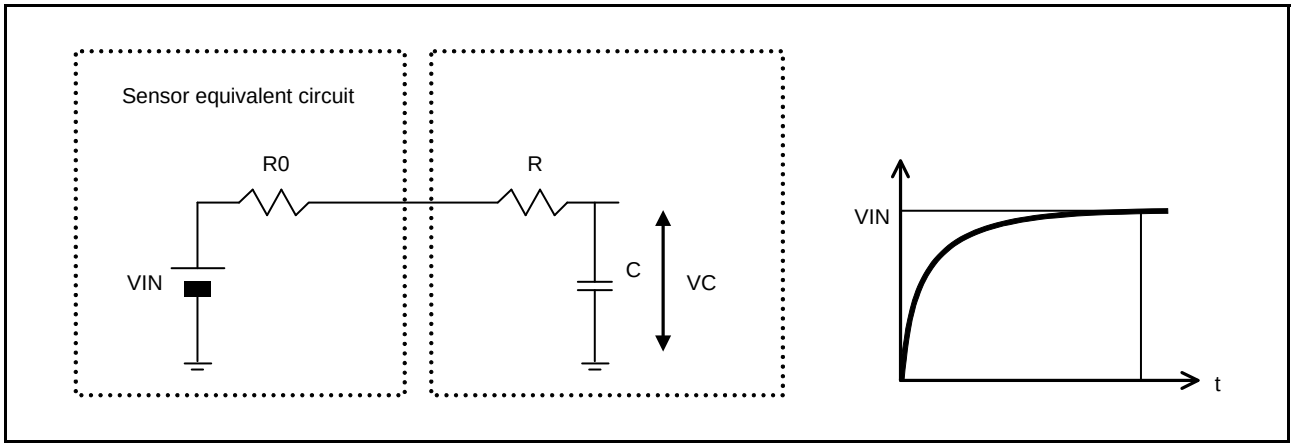


Figure 50.37 Equivalent Circuit of Analog Input Pin and External Sensor

50.6 Usage Notes

50.6.1 Notes on Reading Data Registers

The A/D data registers, A/D data duplication registers, A/D data duplication register A, A/D data duplication register B, A/D temperature sensor data register, A/D internal reference voltage data register, and A/D self-diagnosis data register should be read in 16-bit units. If a register is read twice in 8-bit units, that is, the higher-order byte and lower-order byte are separately read, the A/D-converted value having been read first may disagree with the A/D-converted value having been read for the second time. To prevent this, the data registers should never be read in 8-bit units.

50.6.2 Notes on Stopping A/D Conversion

50.6.2.1 Procedure of Stopping A/D Conversion

To stop A/D conversion when an asynchronous trigger or a synchronous trigger has been selected as the condition for starting A/D conversion, follow the procedure in Figure 50.38.

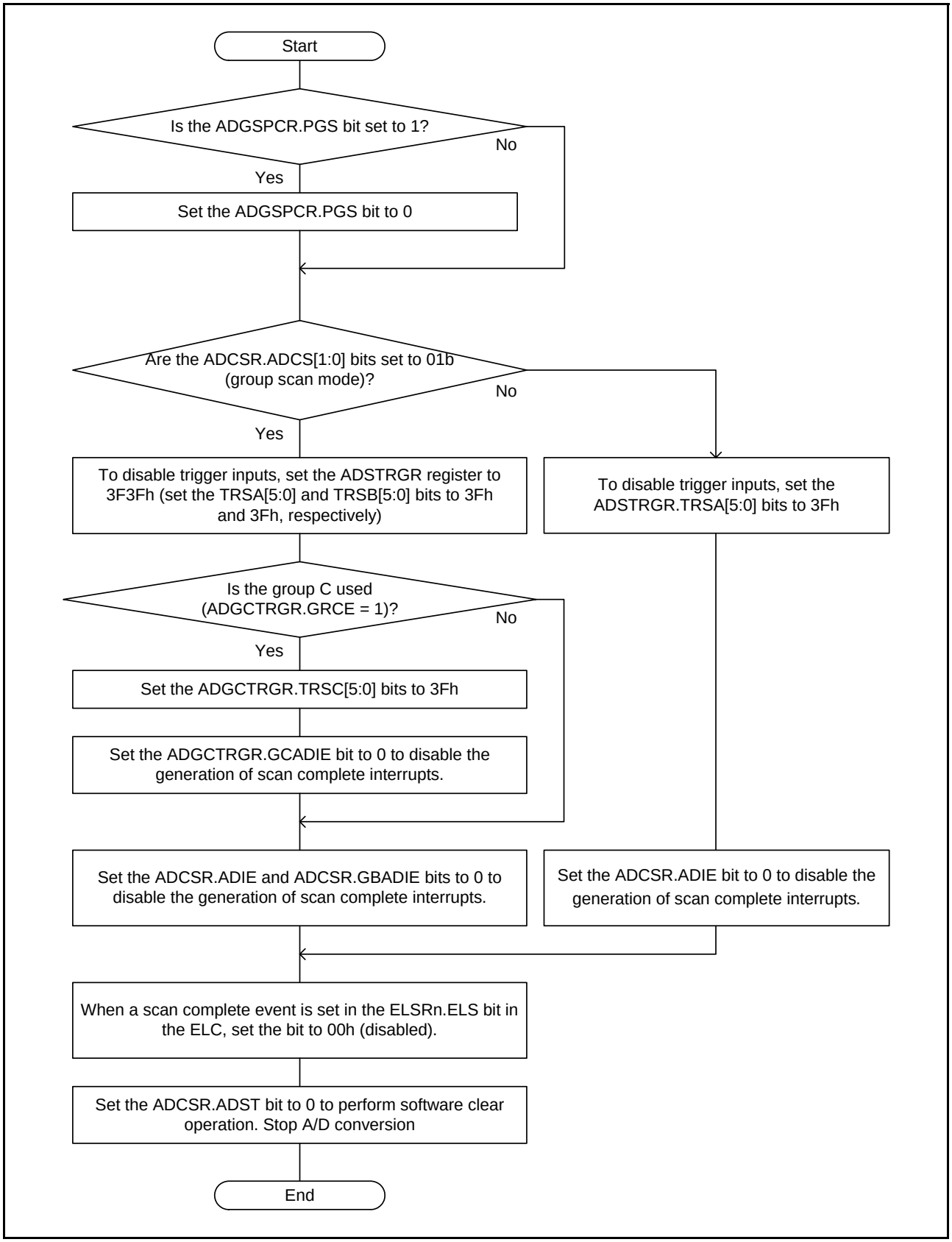


Figure 50.38 Procedure for Clear Operation by Software through the ADCSR.ADST Bit

50.6.2.2 Notes on Modes and Status Bits

The following values may be cleared or reset as described below: the voltage status used in self-diagnosis, the odd-even determination for the double-trigger mode, the bits for monitoring comparison for the comparison function.

- Re-set the settings for the voltage for use in self-diagnosis by setting the ADCER.DIAGLD bit to 1 and setting the ADCER.DIAGVAL[1:0] bits.
- Make double-trigger mode effective starting from the first scan by changing the setting of the ADCSR.DBLE bit from 0 to 1.
- Set the ADCMPCR.CMPAE and ADCMPCR.CMPBE bits to 0, which initializes the bits for monitoring comparison (MONCMPA, MONCMPB, MONCOMB).

50.6.3 A/D Conversion Restarting Timing and Termination Timing

It takes a maximum of six ADCLK cycles for the idle analog unit of the 12-bit A/D converter to be restarted by setting the ADCSR.ADST bit to 1. It takes a maximum of two ADCLK cycles for the operating analog unit of the 12-bit A/D converter to be terminated by setting the ADCSR.ADST bit to 0.

50.6.4 Notes on Scan End Interrupt Handling

When scanning the same analog input twice using any trigger, the first A/D-converted data is overwritten with the second A/D-converted data in the case that the CPU does not complete reading the A/D-converted data by the time the A/D conversion of the first analog input for the second scan ends after the first scan end interrupt is generated.

50.6.5 Module Stop Function Setting

Operation of the 12-bit A/D converter can be disabled or enabled by setting module stop control register. The initial setting is for operation of the 12-bit A/D converter to be halted. Register access is enabled by releasing the module stop state.
After the module stop state is released, wait for 1 µs to start A/D conversion. For details, refer to section 11, Low Power Consumption.

50.6.6 Notes on Entering Low Power Consumption States

Before entering the module stop state or software standby mode, make sure to stop A/D conversion. Here, set the ADCSR.ADST bit to 0, and secure certain period of time until the analog unit of the 12-bit A/D converter is stopped. Follow the procedure given below to secure this time.
Set the ADCSR.ADST bit to 0 by following the procedure in Figure 50.38 Procedure for Clear Operation by Software through the ADCSR.ADST Bit. Then wait for two clock cycles of ADCLK before entering the module stop mode or software standby mode.

50.6.7 Notes on Canceling Software Standby Mode

After software standby mode is canceled, wait until the crystal oscillation stabilization time or the PLL circuit stabilization time elapses, and then wait for 1 µs before starting A/D conversion. For details, refer to section 11, Low Power Consumption.

50.6.8 Pin Setting When Using the 12-bit A/D Converter

When using the 12-bit A/D converter unit 0, do not use the P40 to P47, P03, P05, and P07 pins as output pins. We also recommend not using the P00 to P02, P90, PD0 to PD7, PE0, and PE1 pins as output pins. If any of the P00 to P02, P90, PD0 to PD7, PE0, and PE1 pins is used for an output pin, perform A/D conversion several times, eliminate the maximum and minimum values, and obtain the average of the other results.
When using the 12-bit A/D converter unit 1, we recommend not using the P00 to P02, P90, PD0 to PD7, PE0, and PE1 pins as output pins. If any of the P00 to P02, P90, PD0 to PD7, PE0, and PE1 pins is used for an output pins, perform A/D conversion several times, eliminate the maximum and minimum values, and obtain the average of the other results.

50.6.9 Caution When Using an External Bus

A/D conversion at the same time as access to an external bus may produce poor results.
In this case, use a software approach, such as performing A/D conversion several times, then obtaining the average after excluding the highest and lowest values.
Deterioration in A/D conversion resolution can be reduced by changing the assignment of the external data bus. For details of the setting, refer to section 23, Multi-Function Pin Controller (MPC).

50.6.10 Error in Absolute Accuracy When Disconnection Detection Assistance is in Use

Using disconnection detection assistance leads to an error in absolute accuracy of the A/D converter. This is because an error voltage is input to the analog input pins due to the resistive voltage division between the pull-up or pull-down resistor (Rp) and the resistance of the signal source (Rs). This error in absolute accuracy is calculated from the following formula. Only use disconnection detection assistance after thorough evaluation.

Maximum error in absolute accuracy (LSB) = 4095 × Rs / Rp

50.6.11 Voltage Range of Analog Power Supply Pins

If this MCU is used with the voltages outside the following ranges, the reliability of the MCU may be affected.

- Analog input voltage range
  Voltage applied to analog input pins AN000 to AN007, AN100 to AN111: AVSSn ≤ VANn ≤ AVCCn (n = 0, 1)
- Relationship between power supply pin pairs (AVCCn–AVSSn, VCC–VSS)

  A 0.1-µF capacitor should be connected between each pair of power supply pins to create a closed loop with the shortest route possible as shown in Figure 50.39, and connection should be made so that the following condition is satisfied at the supply side: AVSSn = VSS
  When the 12-bit A/D converter is not used, the following conditions should be satisfied:
  AVCCn = VCC and AVSSn = VSS

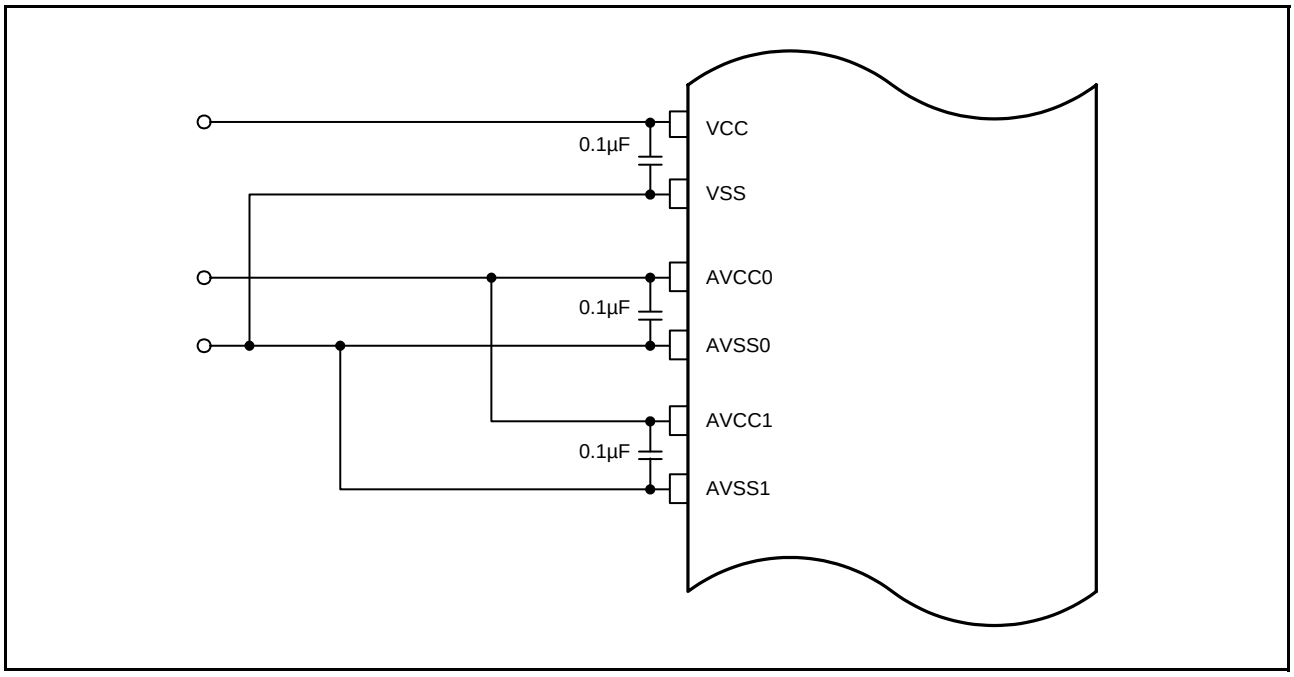


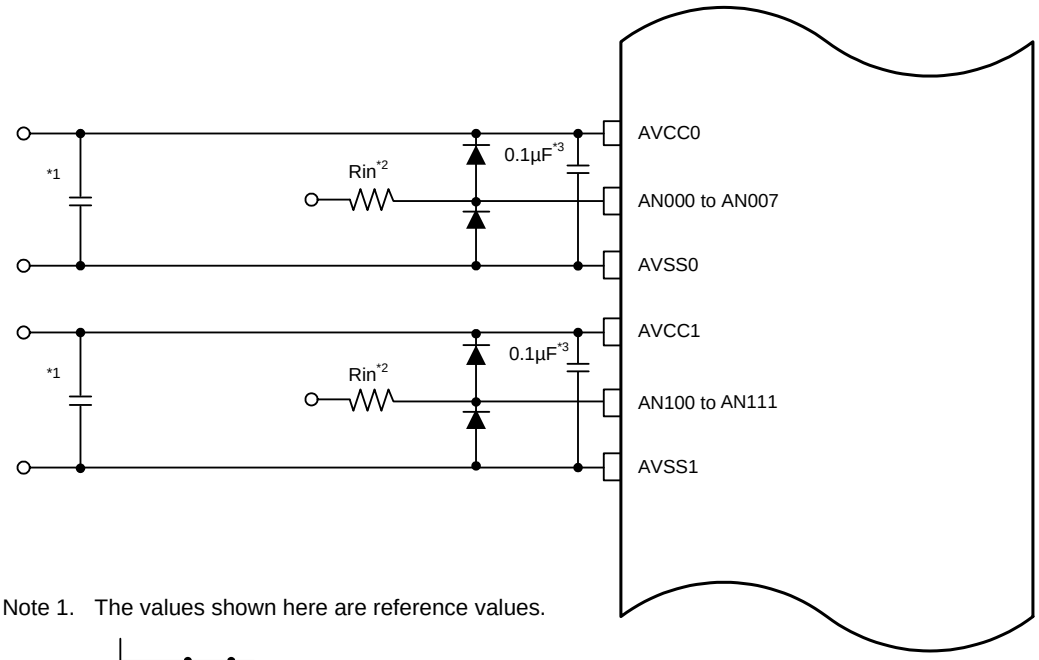
Figure 50.39 Power Supply Pin Connection Example

50.6.12 Notes on Board Design

The board should be designed so that digital circuits and analog circuits are separated from each other as far as possible. In addition, digital circuit signal lines and analog circuit signal lines should not intersect or placed near each other. If these rules are not followed, noise will be produced on analog signals and A/D conversion accuracy will be affected. The analog input pins (AN000 to AN007, AN100 to AN111), and analog power supply (AVCCn) should be separated from digital circuits using the analog ground (AVSSn). The analog ground (AVSSn) should be connected to a stable digital ground (VSS) on the board (single-point ground plane connection).

50.6.13 Notes on Noise Prevention

To prevent the analog input pins (AN000 to AN007, AN100 to AN111) from being destroyed by abnormal voltage such as excessive surge, a capacitor should be inserted between AVCCn and AVSSn, and a protection circuit should be connected to protect the above analog input pins as shown Figure 50.40.



Note 1. The values shown here are reference values.

Note 2. Rin: Signal source impedance

Note 3. Place capacitors to be inserted between the power source of AVCC0, AVSS0, AVCC1, and AVSS1 as close as possible to the power supply pins to prevent deterioration in A/D conversion resolution.

Figure 50.40 Sample Protection Circuit for Analog Inputs

51. Temperature Sensor (TEMPS)

51.1 Overview

This MCU includes a temperature sensor. The temperature sensor outputs a voltage which varies with the temperature. The 12-bit A/D converter unit 1 can convert the voltage from the sensor into a digital value. The temperature around the MCU can be obtained by converting the value into the temperature.

Table 51.1 lists the specifications of the temperature sensor, and Figure 51.1 shows a block diagram of the temperature sensor.

Table 51.1 Specifications of Temperature Sensor

Item	Description
Temperature sensor voltage output	Temperature sensor outputs a voltage to the 12-bit A/D converter unit 1.
Low-power consumption function	Module stop state can be set.
Temperature Sensor Calibration Data	Reference data measured for each chip at factory shipment is stored.

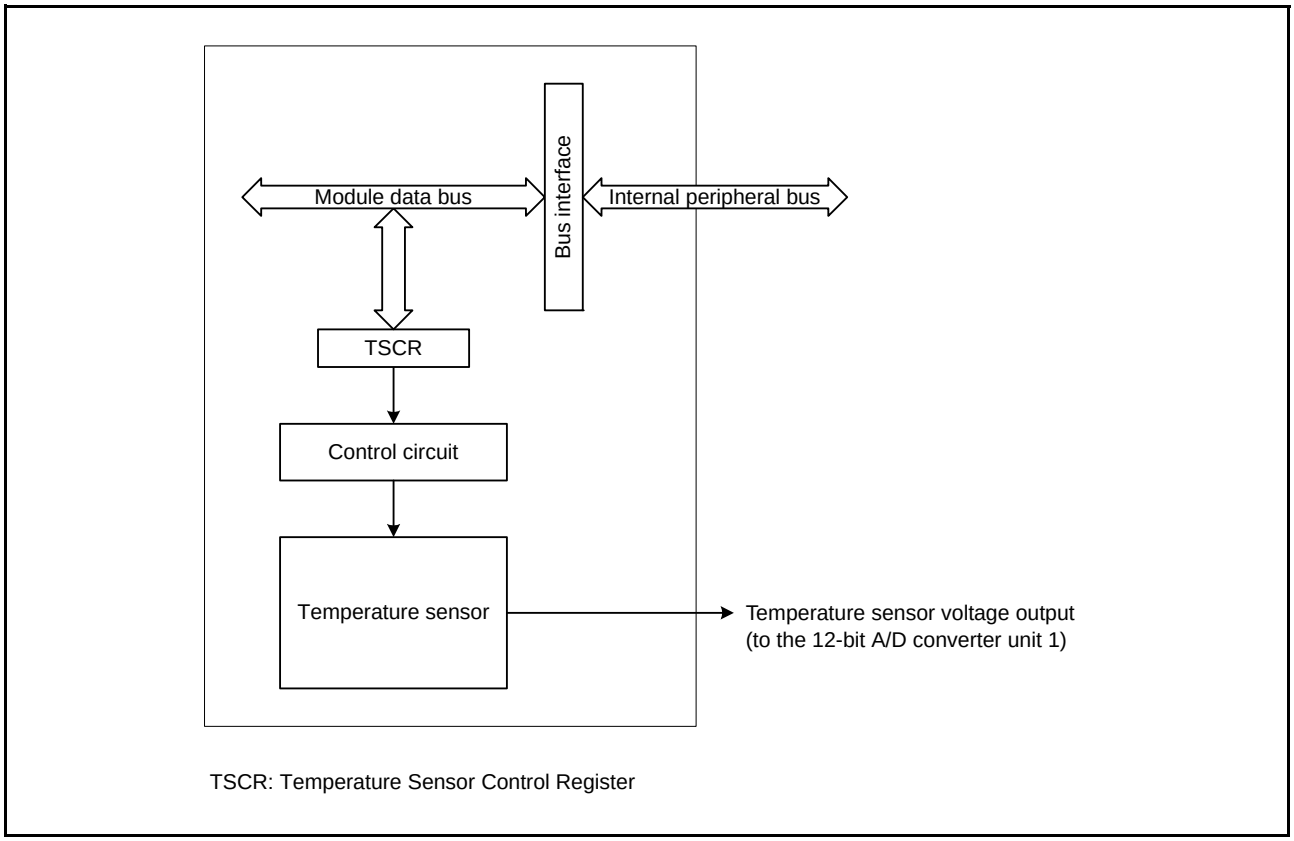


Figure 51.1 Block Diagram of Temperature Sensor

51.2 Register Descriptions

51.2.1 Temperature Sensor Control Register (TSCR)

Address(es): TEMPS.TSCR 0008 C500h

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

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	TSOE	Temperature Sensor Output Enable	0: Disables output from the temperature sensor to the 12-bit A/D converter unit 1. 1: Enables output from the temperature sensor to the 12-bit A/D converter unit 1.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	TSEN	Temperature Sensor Enable	0: Stops the temperature sensor. 1: Starts the temperature sensor.	R/W

The settings of TSCR register have the timing restrictions shown in Figure 51.4.

51.2.2 Temperature Sensor Calibration Data Register (TSCDR)

Address(es): TEMPSCONST.TSCDR FE7F 7D7Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—												
Value after reset:	0	0	0	0	Unique value for each chip											

Note: This register is only readable when the SYSCR0.ROME bit is 1 (the on-chip ROM is enabled).

The TSCDR register stores temperature sensor calibration data measured for each chip at factory shipment. The TSCDR register is a 32-bit read-only register and should be read in 32-bit units.
Temperature sensor calibration data is a digital value obtained using the 12-bit A/D converter unit 1 to convert the voltage output by the temperature sensor under the condition of Ta = Tj = 128°C and AVCC1 = 3.3 V.
The voltage V1 output by the temperature sensor under the condition of Ta = Tj = 128°C can be calculated from the value of the TSCDR register according to the formula below.

V1 = 3.3 × value of the TSCDR register/4096 [V]

Note that the AVCC1 voltage does not affect voltage V1.

51.3 Using the Temperature Sensor

The temperature sensor outputs a voltage which varies with the temperature.
This voltage is converted to a digital value by the 12-bit A/D converter unit 1. The temperature around the MCU can be obtained by converting the value into the temperature.

51.3.1 Preparation for Using the Temperature Sensor

Perform a calibration of the temperature sensor as shown below. The voltage output by the temperature sensor is proportional to temperature, which can be calculated according to the following formula.

Formula for the temperature characteristic:
T = (Vs – V1)/Slope + T1

T: Measured temperature (°C)
Vs: Voltage output by the temperature sensor when the temperature is measured (V)
T1: Sample temperature measurement at first point (°C)
V1: Voltage output by the temperature sensor when T1 is measured (V)
T2: Sample temperature measurement at second point (°C)
V2: Voltage output by the temperature sensor when T2 is measured (V)
Slope: Temperature slope of the temperature sensor (V/°C); Slope = (V2 – V1)/(T2 – T1)

Characteristics of the temperature sensor vary from MCU to MCU. Therefore, a two-point calibration (the following experimental measurement at two different temperatures) is recommended.
Use the 12-bit A/D converter unit 1 to measure the voltage V1 output by the temperature sensor at temperature T1. Again, using the 12-bit A/D converter unit 1, measure the voltage V2 output by the temperature sensor at a different temperature T2. Obtain the temperature slope (Slope = (V2 – V1)/(T2 – T1)) from these results.
Subsequently, obtain temperatures by substituting the slope into the formula for the temperature characteristic (T = (Vs – V1)/Slope + T1).

If you are using the temperature slope given in Table 56.56 of section 56, Electrical Characteristics, use the 12-bit A/D converter unit 1 to measure the voltage V1 output by the temperature sensor at temperature T1, and then calculate the temperature characteristic by using the formula below.
However, this calibration gives less accurate temperatures than two-point calibration.

T = (Vs – V1)/Slope + T1

T: Measured temperature (°C)
Vs: Voltage output by the temperature sensor when the temperature is measured (V)
T1: Sample temperature measurement at first point (°C)
V1: Voltage output by the temperature sensor when T1 is measured (V)
Slope: Temperature slope given in Table 56.56 ÷ 1000 (V/°C)

In this MCU, the TSCDR register stores the temperature value (CAL_{128}) of the temperature sensor measured under the condition of Ta = Tj = 128°C and AVCC1 = 3.3 V. By using this value as the sample measurement result at the first point, preparation before using the temperature sensor can be omitted.
If V1 is calculated from CAL_{128},

$V1 = 3.3 \times CAL_{128}/4096$ [V]
Using this, the measured temperature can be calculated according to the formula below.

T = (Vs – V1)/Slope + 128 [°C]

T: Measured temperature (°C)
Vs: Voltage output by the temperature sensor when the temperature is measured (V)
V1: Voltage output by the temperature sensor when Ta = Tj = 128°C and AVCC1 = 3.3 V (V)
Slope: Temperature slope given in Table 56.56 ÷ 1000 (V/°C)

Error in the measured temperature (the range of variation is 3σ) is shown in Figure 51.2.

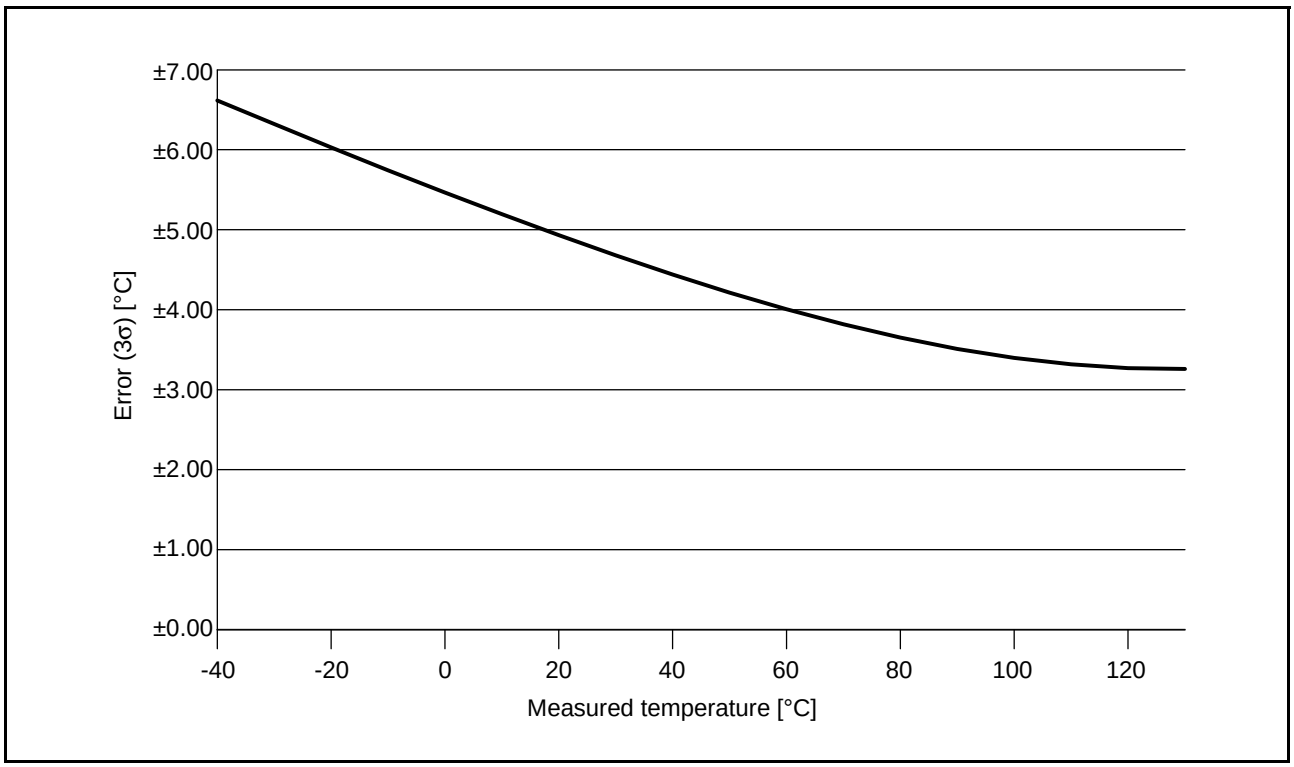


Figure 51.2 Error in the Measured Temperature

51.3.2 Setting of 12-Bit A/D Converter Unit 1

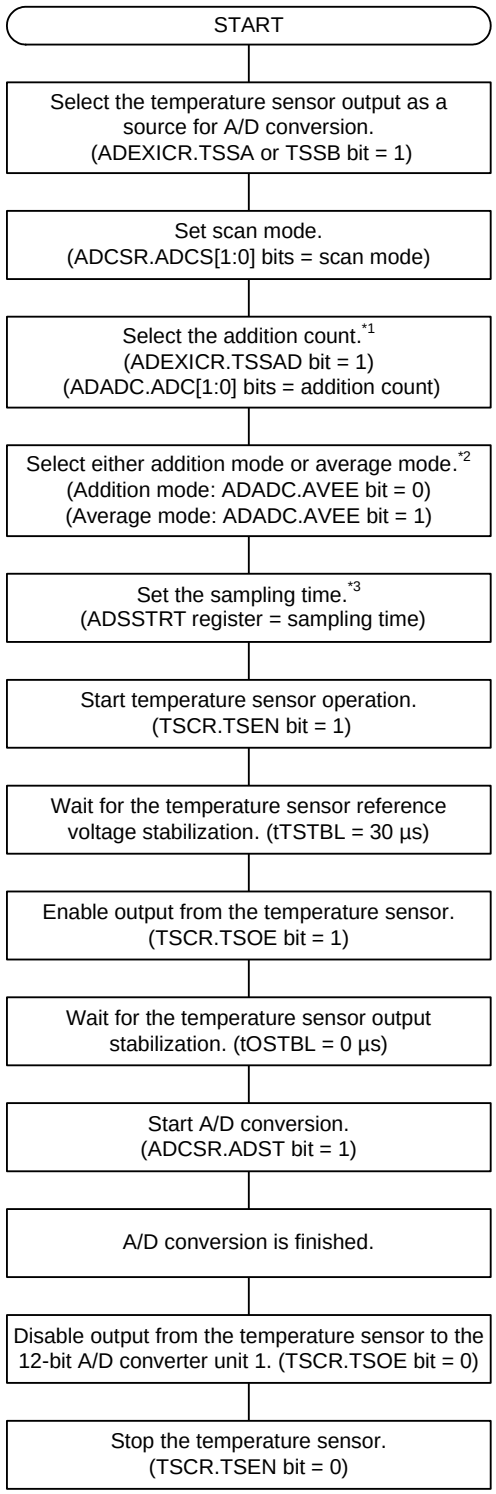
For A/D conversion of temperature sensor output voltages, 12-bit A/D converter unit 1 registers should be set as follows.

- Selecting the Temperature Sensor Voltage as a Source for A/D Conversion
  Select the temperature sensor voltage as a source for A/D conversion by setting the temperature sensor output A/D conversion select bit in the A/D conversion extended input control register (TSSA or TSSB in the ADEXICR register) to 1.
- Setting Scan Mode
  Select scan mode by setting the scan mode select bits in the A/D control register (ADCSR.ADCS[1:0]).
- Setting Addition/Average Mode
  For A/D conversion of the temperature sensor output, additional or average mode is selectable. To use either additional or average mode, set the temperature sensor output A/D converted value addition mode select bit in the A/D conversion extended input control register (ADEXICR.TSSAD) to 1, and the addition count select bits in the A/D converted value addition count select register (ADADC.ADC[1:0]) to the desired number of addition. Furthermore, clear the AVEE bit in ADADC to 0 to select addition mode; set the AVEE bit in ADADC to 1 to select average mode. In average mode, however, the ADC[1:0] bits in ADADC should not be set to 10b.
- Setting the Sampling Time of the 12-bit A/D converter Unit 1
  The sampling time of the 12-bit A/D converter can be changed at converting the output from the temperature sensor. The initial sampling time is 11 cycles of ADCLK. To change the setting of sampling time from 11 cycles of ADCLK, set the A/D sampling state register T (ADSSTRT) while the ADCSR.ADST bit is 0.

Setting the A/D conversion start bit in the A/D control register (ADCSR.ADST) to 1 starts A/D conversion, and the result is stored in the A/D temperature sensor data register (ADTSDR). If you will be using A/D conversion of the output from the temperature sensor, do so in accord with section 51.3.3, Procedure for Using the Temperature Sensor.

51.3.3 Procedure for Using the Temperature Sensor

Figure 51.3 shows the procedure for using the temperature sensor.



Note 1. When not using the addition/average mode, this setting is not required. To select average mode, do not set the ADADC.ADC[1:0] bits to 10b.
Note 2. When not using the addition/average mode, this setting is not required.
Note 3. The initial sampling time is 11 cycles of ADCLK. If the sampling time is not to be changed, this setting is not required.

Figure 51.3 Procedure for Using the Temperature Sensor

51.3.4 Timing of A/D Conversion of Temperature Sensor Output

Figure 51.4 shows the timing from the start of temperature-sensor operation until the completion of A/D conversion when only the output from the temperature sensor is to be A/D converted and conversion is in single-scan mode. The times shown in the figure are described in Table 51.2.

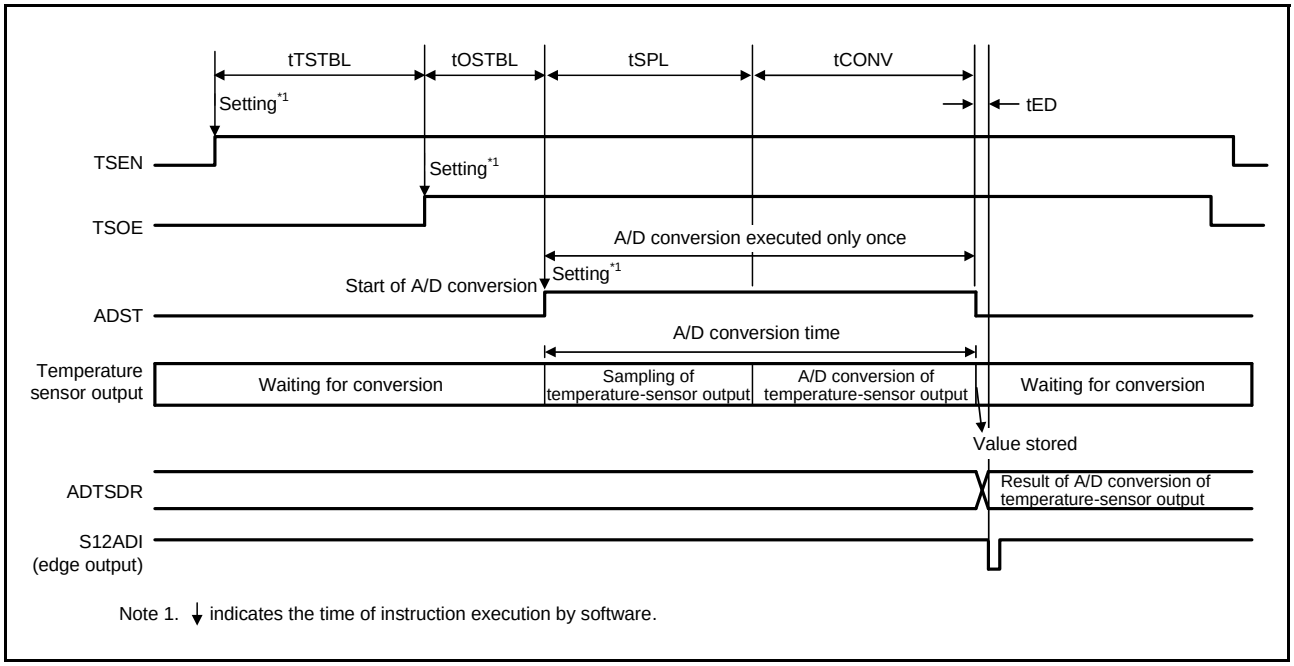


Figure 51.4 Timing from the Start of Temperature-Sensor Operation until Completion of A/D Conversion

Table 51.2 Time until Completion of A/D Conversion after the Start of Temperature-Sensor Operation

Item	Symbol	Time
Temperature-sensor reference-voltage stabilization wait time	tTSTBL	30 µs (min)
Temperature-sensor output stabilization wait time	tOSTBL	0 µs (min)
12-bit A/D converter unit 1 input sampling time	tSPL	ADSSTRT setting × tC(ADCLK)
A/D conversion time	tCONV	Refer to Table 50.18, Times for Conversion during Scanning (in Numbers of Cycles of ADCLK and PCLK) in section 50.3.7, Analog Input Sampling Time and Scan Conversion Time.
Scan conversion end delay time	tED	Refer to Table 50.18, Times for Conversion during Scanning (in Numbers of Cycles of ADCLK and PCLK) in section 50.3.7, Analog Input Sampling Time and Scan Conversion Time.

51.4 Usage Note

51.4.1 Module-Stop Function Setting

The corresponding bit in module stop control register B (MSTPCRB) can be used to enable and disable the temperature sensor. The initial setting is for the temperature sensor to be stopped. The register becomes accessible by releasing from the module-stop state. For details, refer to section 11, Low Power Consumption.

52. Data Operation Circuit (DOCA)

52.1 Overview

The data operation circuit (DOC) is used to compare, add, or subtract 16- or 32-bit values.

Table 52.1 lists the specifications of the DOC and Figure 52.1 is a block diagram of the DOC.

An interrupt can be generated if the result of 16- or 32-bit comparison meets one of the set interrupt conditions or if the result of the addition or subtraction result of 16- or 32-bit values is an overflow or underflow.

Table 52.1 DOC Specifications

Item	Description
Data operation function	• Comparison of 16- or 32-bit values (equal or not equal, greater or less than, or within or beyond a range) • Addition or subtraction of 16- or 32-bit values
Lower power consumption function	The DOC can be placed in a module-stop state.
Interrupts	• The result of data comparison meets the detection condition. • The result of data addition is greater than FFFFh (DOCR.DOPSZ = 0) or FFFF FFFFh (DOCR.DOPSZ = 1), which is an overflow. • The result of data subtraction is less than 0000h (DOCR.DOPSZ = 0) or 0000 0000h (DOCR.DOPSZ = 1), which is an underflow.
Event link function (output)	• The result of data comparison meets the detection condition. • The result of data addition is greater than FFFFh (DOCR.DOPSZ = 0) or FFFF FFFFh (DOCR.DOPSZ = 1), which is an overflow. • The result of data subtraction is less than 0000h (DOCR.DOPSZ = 0) or 0000 0000h (DOCR.DOPSZ = 1), which is an underflow.

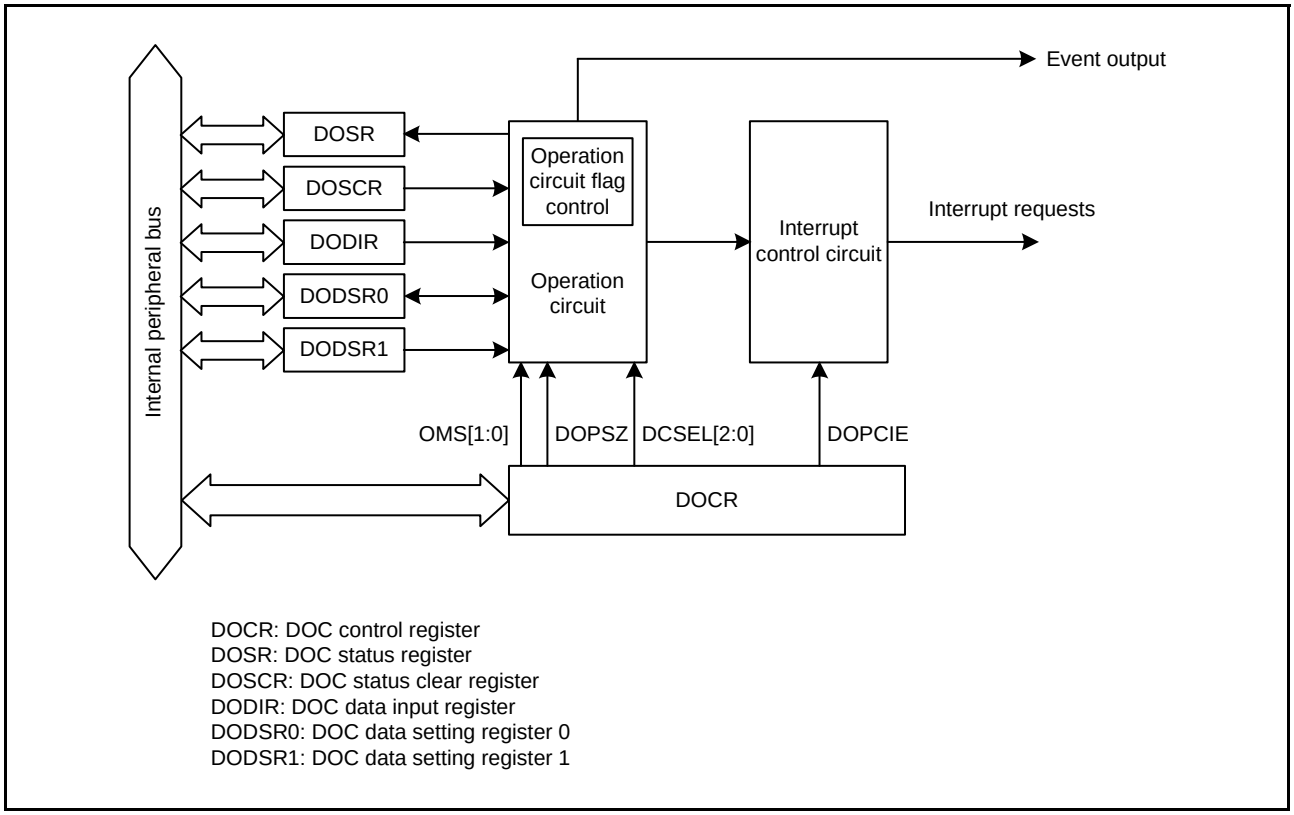
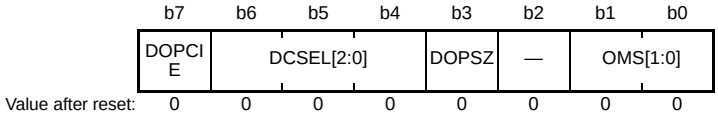


Figure 52.1 DOC Block Diagram

52.2 Register Descriptions

52.2.1 DOC Control Register (DOCR)

Address(es): DOC.DOCR 000A 0580h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	OMS[1:0]	Operating Mode Select	b1 b0 0 0: Data comparison mode 0 1: Data addition mode 1 0: Data subtraction mode 1 1: Setting prohibited	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	DOPSZ	Data Operation Size Select	0: 16-bit width 1: 32-bit width	R/W
b6-b4	DCSEL[2:0]	Detection Condition Select*1	b6 b4 0 0 0: Not equal to (DODIR ≠ DODSR0) 0 0 1: Equal to (DODIR = DODSR0) 0 1 0: Less than (DODIR < DODSR0) 0 1 1: Greater than (DODIR > DODSR0) 1 0 0: Within the range (DODSR0 < DODIR < DODSR1) 1 0 1: Beyond the range (DODIR < DODSR0, DODSR1 < DODIR) Settings other than above are prohibited.	R/W
b7	DOPCIE	Data Operation Circuit Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/W

Note 1. Valid only when data comparison mode is selected.

The DOCR register specifies the operation of DOC, or enabling or disabling of the interrupt.

OMS[1:0] Bits (Operating Mode Select)

These bits select the operating mode of the DOC.

DOPSZ Bit (Data Operation Size Select)

This bit selects the size of the data operation.

DCSEL[2:0] Bits (Detection Condition Select)

This bit is valid only when data comparison mode is selected.
This bit selects the condition for detection in data comparison mode.

DOPCIE Bit (Data Operation Circuit Interrupt Enable)

Setting this bit to 1 enables interrupts from the DOC.

52.2.2 DOC Status Register (DOSR)

Address(es): DOC.DOSR 000A 0584h

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

Bit	Symbol	Bit Name	Description	R/W
b0	DOPCF	Data Operation Result Flag	Indicates the result of an operation.	R
b7-b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The DOSR register indicates the results of data operation.

DOPCF Flag (Data Operation Result Flag)

[Setting conditions]

- The setting of the DOCR.OMS[1:0] bits is 00b (data comparison mode) and the result of data comparison meets the condition selected by the DOCR.DCSEL[2:0] bits.
- The setting of the DOCR.OMS[1:0] bits is 01b (data addition mode) and the result of addition was greater than FFFFh (DOCR.DOPSZ = 0) or FFFF FFFFh (DOCR.DOPSZ = 1).
- The setting of the DOCR.OMS[1:0] bits is 10b (data subtraction mode) and the result of subtraction is below 0000h (DOCR.DOPSZ = 0) or 0000 0000h (DOCR.DOPSZ = 1).

[Clearing condition]

- Writing 1 to the DOSCR.DOPCFCL bit

52.2.3 DOC Status Clear Register (DOSCR)

Address(es): DOC.DOSCR 000A 0588h

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

Bit	Symbol	Bit Name	Description	R/W
b0	DOPCFCL	Data Operation Result Clear	0: Retain the value of the DOPCF flag. 1: Clears the DOPCF flag.	W
b7-b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

The DOSCR register is for clearing the DOPCF flag. It is always read as 00h.

DOPCFCL Bit (Data Operation Result Clear)

Writing 1 to this bit clears the DOSR.DOPCF flag.

52.2.4 DOC Data Input Register (DODIR)

Address(es): DOC.DODIR 000A 058Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

The DODIR register is a readable and writable register that holds values for use in operations.
Access the DODIR register with the data operation size selected by the DOCR.DOPSZ bit.

52.2.5 DOC Data Setting Register 0 (DODSR0)

Address(es): DOC.DODSR0 000A 0590h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

The DODSR0 register is a readable and writable register that holds values for use in comparison or the results of other operations.
Access the DODSR0 register with the data operation size selected by the DOCR.DOPSZ bit.
In data comparison mode, store the standard value for use in comparison in this register. When either ‘within the range’ (DOCR.DCSEL[2:0] = 100b) or ‘beyond the range’ (DOCR.DCSEL[2:0] = 101b) is selected, specify the lower boundary of the range.
In data addition or data subtraction mode, this register holds the results of operations.

52.2.6 DOC Data Setting Register 1 (DODSR1)

Address(es): DOC.DODSR1 000A 0594h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

The DODSR1 register is a readable and writable register that holds a value for use in range comparison.
Access the DODSR1 register with the data operation size selected by the DOCR.DOPSZ bit.
When either ‘within the range’ (DOCR.DCSEL[2:0] = 100b) or ‘beyond the range’ (DOCR.DCSEL[2:0] = 101b) is selected in data comparison mode, specify the upper boundary of the range.
This register is only used when either ‘within the range’ or ‘beyond the range’ is selected.

52.3 Operation

52.3.1 Data Comparison Mode

Figure 52.2 to Figure 52.7 show an example of the steps involved in data comparison mode operation by the DOC*1. An example of operation with the operation size of 32 bits is shown below.

(1) Writing 00b to the DOCR.OMS[1:0] bits places the DOC in the data comparison mode. At the same time, write to the DOCR.DCSEL[2:0] bits to select the condition to be detected.
(2) Specify the standard values for comparison in the DODSR0 and DODSR1 registers.*2
(3) Write the value for comparison in the DODIR register.
(4) If the value written to the DODIR register satisfies the condition set in the DOCR.DCSEL[2:0] bits, the DOSR.DOPCF flag becomes 1, and an ELC event is generated. If the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also issued.

Note 1. Comparison is made to proceed at the same time a value is written to the DODIR register. Writing values to the DODSR0 and DODSR1 registers does not make comparison proceed.
Note 2. Setting of the DODSR1 register is only required when either ‘within the range’ or ‘beyond the range’ is selected. Set a greater value for the DODSR1 register than that of the DODSR0 register.

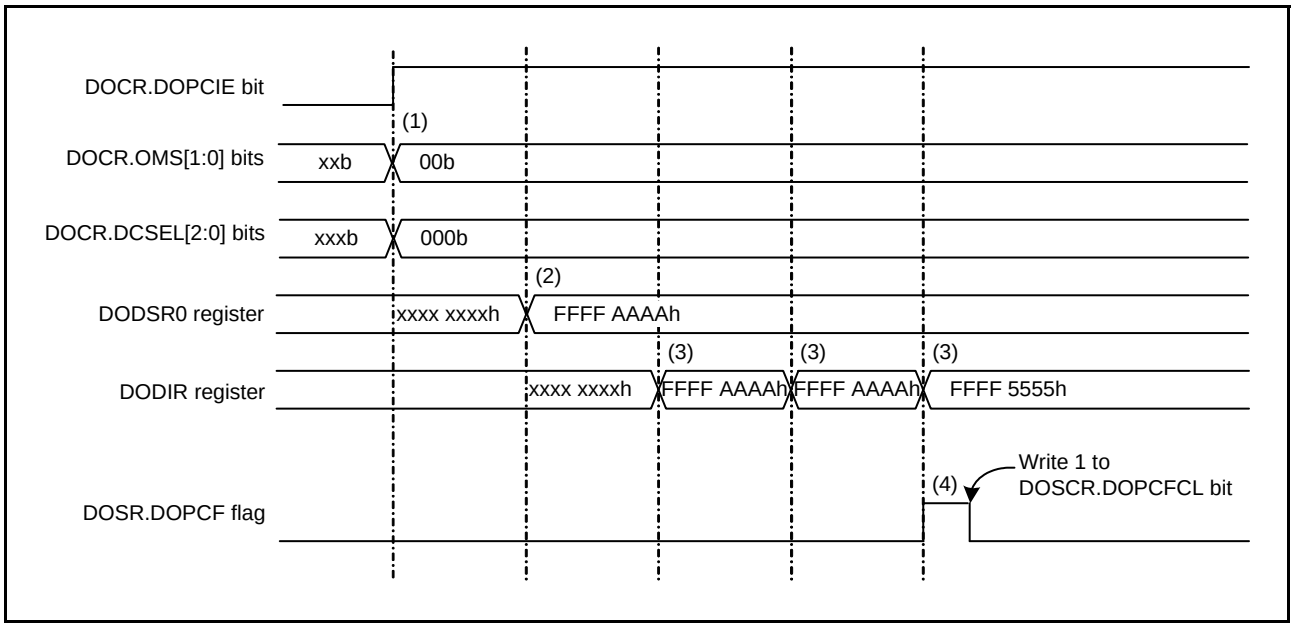


Figure 52.2 Example of the Operation of ‘Not Equal to’ as the Detection Condition in Data Comparison Mode

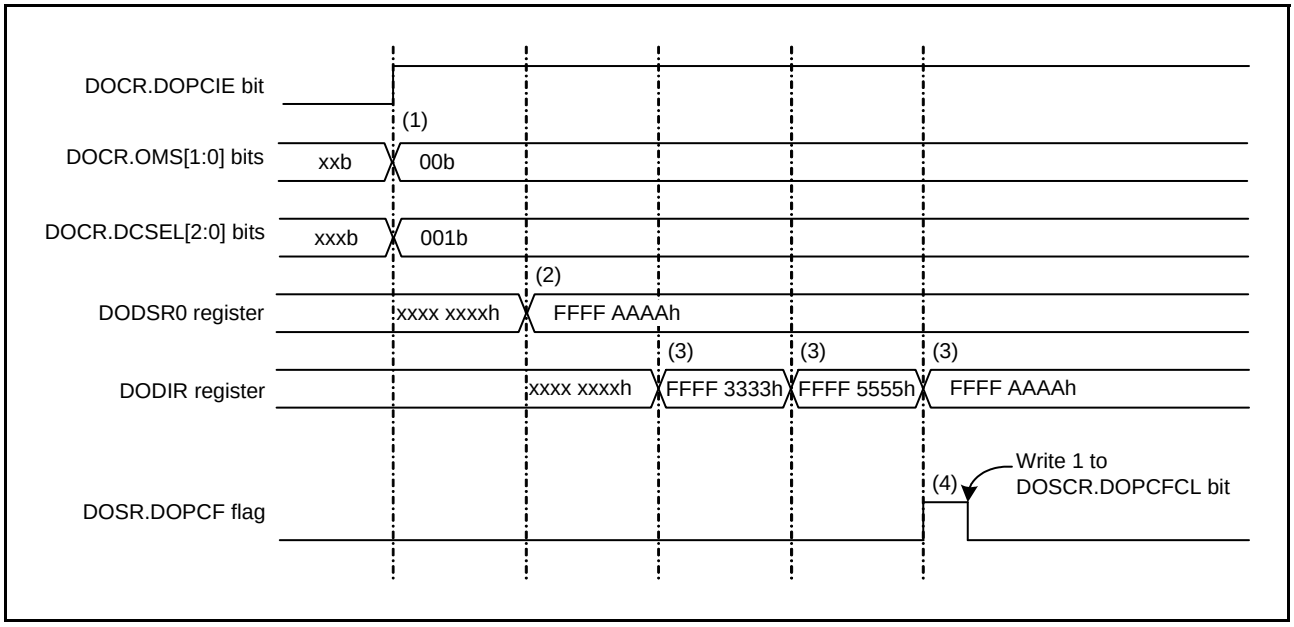


Figure 52.3 Example of the Operation of 'Equal to' as the Detection Condition in Data Comparison Mode

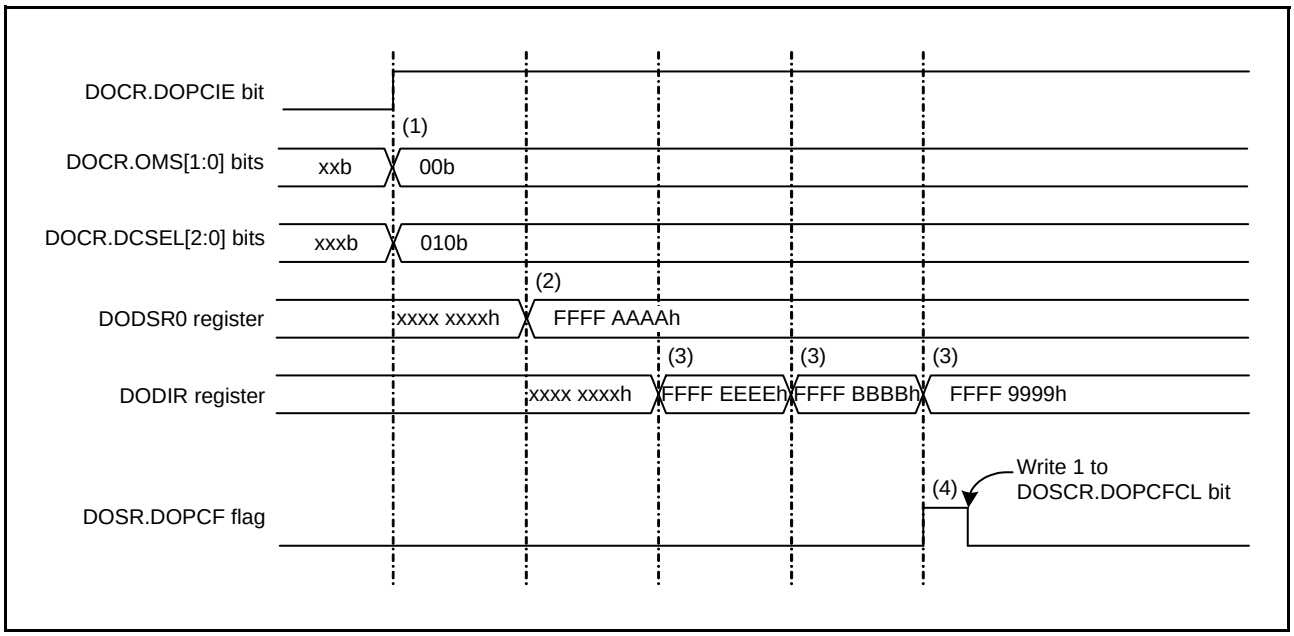


Figure 52.4 Example of the Operation of 'Less Than' as the Detection Condition in Data Comparison Mode

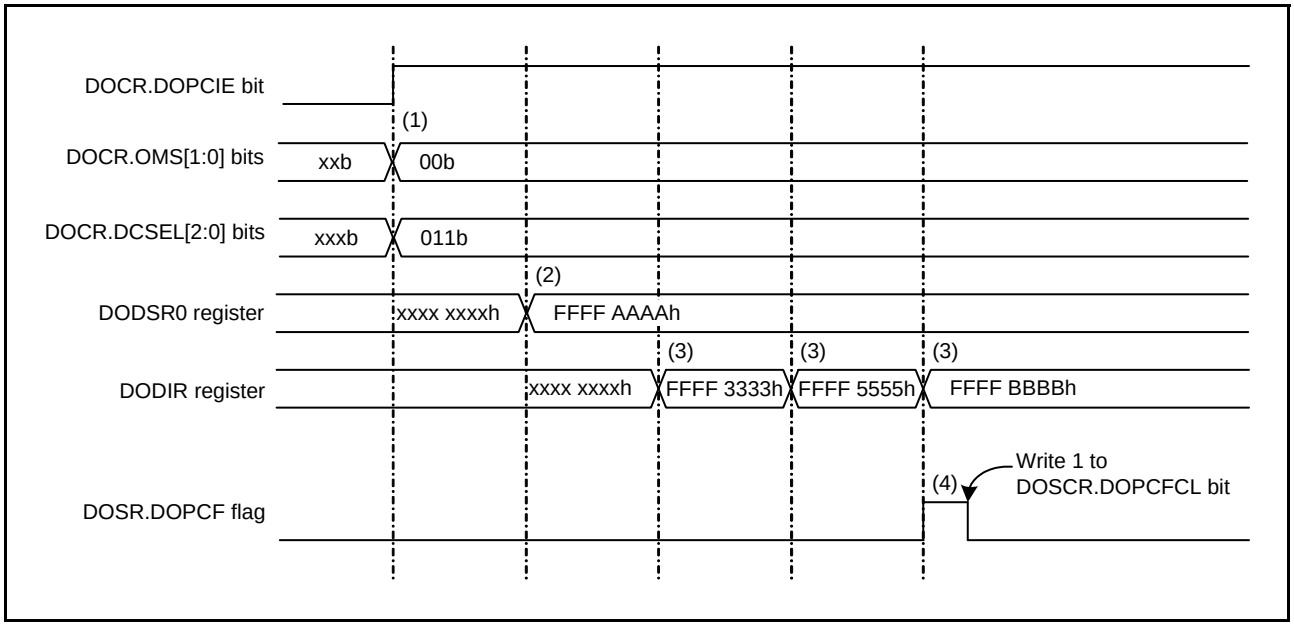


Figure 52.5 Example of the Operation of 'Greater Than' as the Detection Condition in Data Comparison Mode

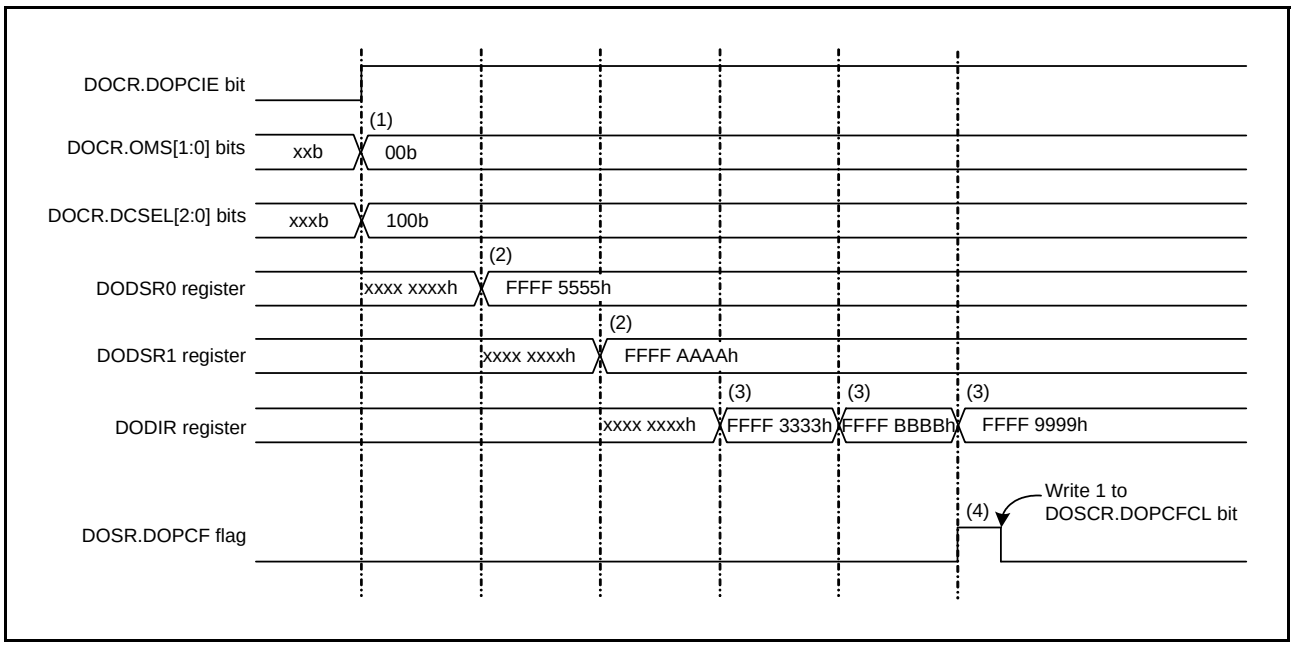


Figure 52.6 Example of the Operation of 'Within the Range' as the Detection Condition in Data Comparison Mode

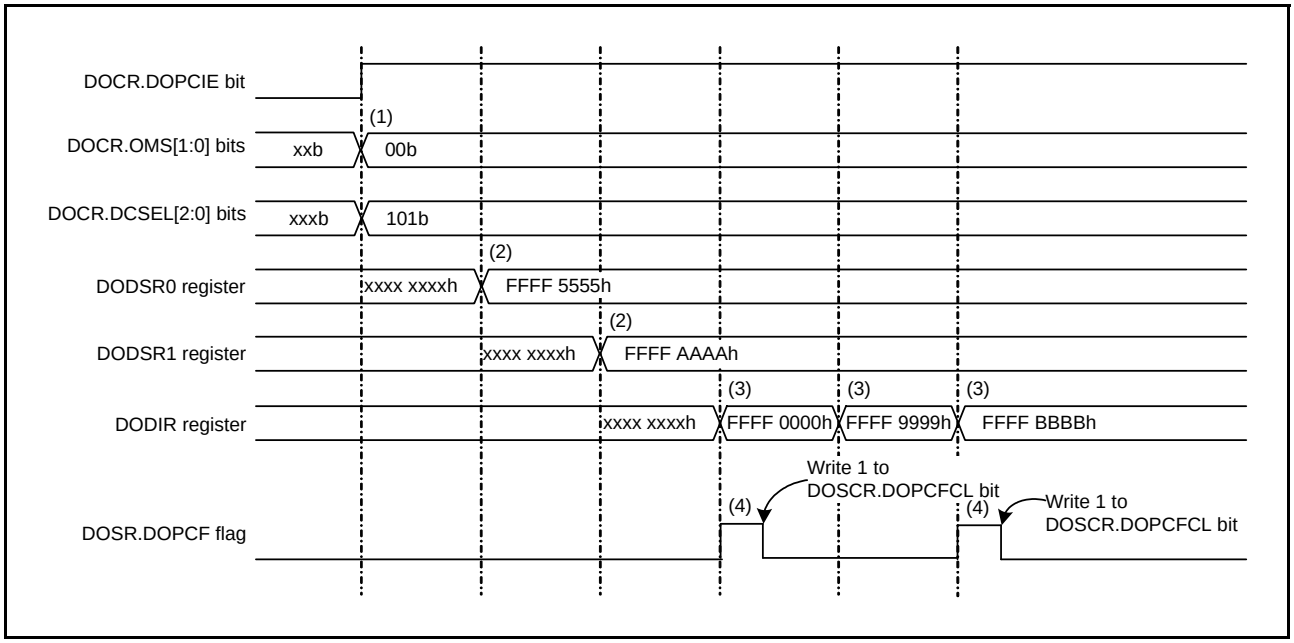


Figure 52.7 Example of the Operation of 'Beyond the Range' as the Detection Condition in Data Comparison Mode

52.3.2 Data Addition Mode

Figure 52.8 shows an example of the steps involved in data addition mode operation*1 by the DOC.
An example of operation when the data operation size is 32 bits is shown below.

(1) Writing 01b to the DOCR.OMS[1:0] bits selects data addition mode.
(2) Set the initial value in the DODSR0 register.
(3) Write the value for addition in the DODIR register. The result of the operation is stored in DODSR0.
(4) Write all values for use in addition to the DODIR register.
(5) If the result of the operation is greater than FFFF FFFFh, the DOSR.DOPCF flag becomes 1, and an ELC event is generated. If the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also issued.

Note 1. Addition is made to proceed at the same time as a value is written to the DODIR register. Writing a value to the DODSR0 register does not make addition proceed.

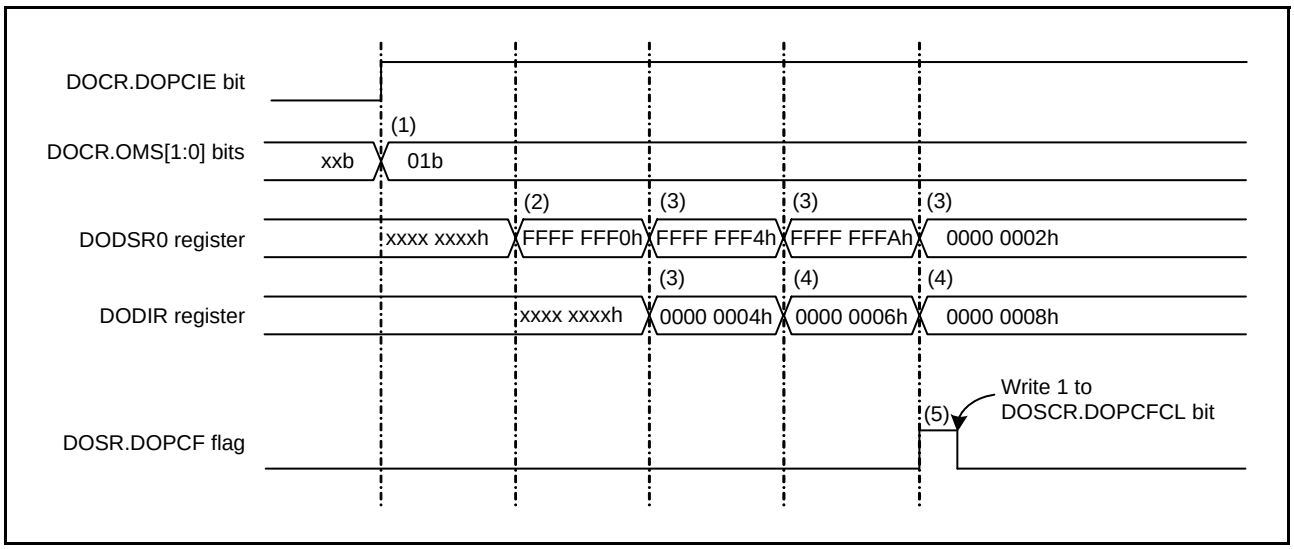


Figure 52.8 Example of Operation in Data Addition Mode

52.3.3 Data Subtraction Mode

Figure 52.9 shows an example of the steps involved in data subtraction mode operation*1 by the DOC.
An example of operation when the data operation size is 32 bits is shown below.

(1) Writing 10b to the DOCR.OMS[1:0] bits selects data subtraction mode.
(2) Set the initial value in the DODSR0 register.
(3) Write the value for subtraction in the DODIR register. The result of the operation is stored in DODSR0.
(4) Write all values for use in subtraction to the DODIR register.
(5) If the result of the operation is less than 0000 0000h, the DOSR.DOPCF flag becomes 1, and an ELC event is generated. If the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also issued.

Note 1. Subtraction is made to proceed at the same time a value is written to the DODIR register. Writing a value to the DODSR0 register does not make subtraction proceed.

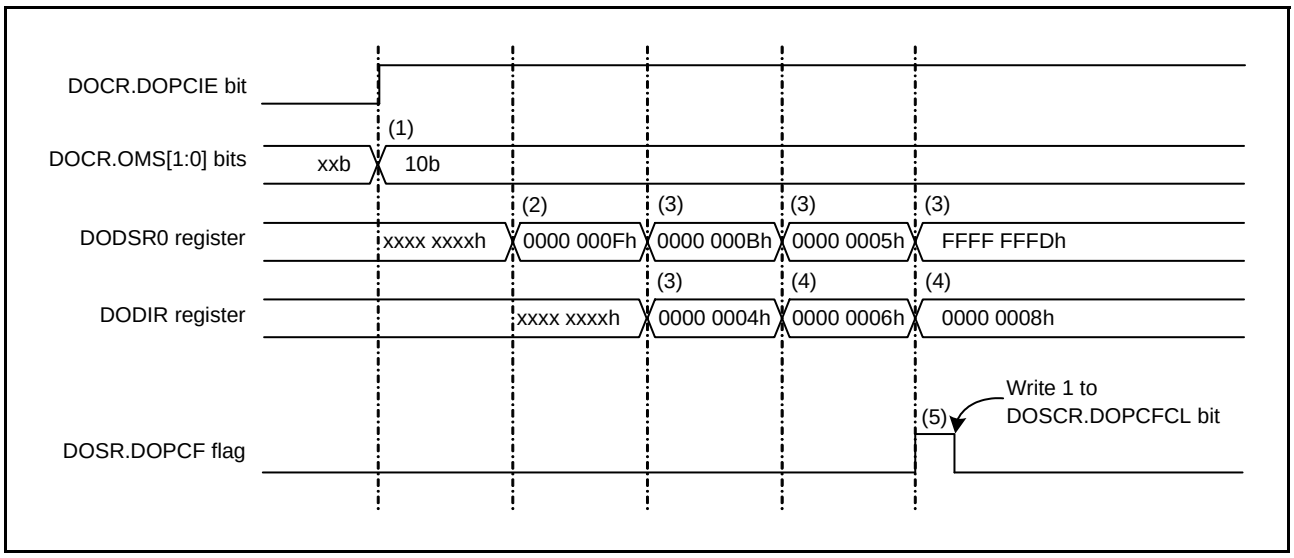


Figure 52.9 Example of Operation in Data Subtraction Mode

52.4 Interrupt Requests

The data operation circuit interrupt (DOPCI) is the interrupt request generated by the DOC. The DOSR.DOPCF flag becomes 1 when the interrupt source condition is satisfied, and an interrupt request is also issued if the DOCR.DOPCIE bit is 1.
Table 52.2 lists the details of the interrupt request.

Table 52.2 Interrupt Request from DOC

Interrupt Request	Data Operation Result Flag	Interrupt Generation Timing
Data operation circuit interrupt (DOPCI)	DOPCF	• The result of data comparison meets the detection condition. • The result of data addition is greater than FFFFh (DOCR.DOPSZ = 0) or FFFF FFFFh (DOCR.DOPSZ = 1). • The result of data subtraction is less than 0000h (DOCR.DOPSZ = 0) or 0000 0000h (DOCR.DOPSZ = 1).

52.5 Event Link Output

The DOC outputs event signals for the event link controller (ELC) under the following conditions, and these can be used to initiate operations by other modules selected in advance.

- The result of data comparison meets the detection condition.
- The result of data addition is greater than FFFFh (DOCR.DOPSZ = 0) or FFFF FFFFh (DOCR.DOPSZ = 1).
- The result of data subtraction is less than 0000h (DOCR.DOPSZ = 0) or 0000 0000h (DOCR.DOPSZ = 1).

52.5.1 Interrupt Handling and Event Linking

The DOC has a bit to enable or disable interrupts. When an interrupt source condition is satisfied while the interrupt is enabled, the interrupt request signal is issued to the CPU.
In contrast, an event link output signal is sent to other modules as an event signal via the ELC when an interrupt source is generated, regardless of the setting of the corresponding interrupt enable bit.

52.6 Usage Note

52.6.1 Module Stop Function Setting

Operation of the DOC can be enabled or disabled by setting the MSTPB6 bit in module stop control register B (MSTPCRB). The DOC is initially disabled after a reset. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

53. RAM

This MCU has a 384-Kbyte high-speed static RAM (RAM), which runs at the frequency of 120 MHz in the no-wait state.

53.1 Overview

Table 53.1 lists the specifications of the RAM.

Table 53.1 Specifications of RAM

Item	RAM
Capacity	384 Kbytes
Address	0000 0000h to 0005 FFFFh
Memory bus	Memory bus 1
Access	• Single-cycle access is possible for both reading and writing.*1 • Enabling or disabling of the RAM is selectable.*2
Data retention function	Not available in deep software standby mode
Low power consumption function	Transitions to the module stop state are possible.
Error checking	• Parity check: Detection of 1-bit errors • A non-maskable interrupt or interrupt is generated in response to an error.

Note 1. When accessing across the 8-byte boundary, the number of cycles is doubled.
Note 2. Selectable by the RAME bit in SYSCR1. For details on SYSCR1, refer to section 3.2.3, System Control Register 1 (SYSCR1).

53.2 Register Descriptions

53.2.1 RAM Operating Mode Control Register (RAMMODE)

Address(es): 0008 1200h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	RAMMODE[1:0]	RAM Operating Mode Select	b1 b0 0 0: Parity checking is disabled. 0 1: Parity checking is enabled. Settings other than above are prohibited.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R

The RAMMODE register is write-protected by the RAM protection register (RAMPRCR). Before writing to the RAMMODE register, set the RAMPRCR.RAMPRCR bit to 1 to enable writing to it. Set the RAMMODE register before starting access to the RAM. If this register is modified after accessing to the RAM, RAM operation is not guaranteed.

53.2.2 RAM Error Status Register (RAMSTS)

Address(es): 0008 1201h

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

Bit	Symbol	Bit Name	Description	R/W
b0	RAMERR	RAM Error Status Flag	0: A parity check error has not occurred. 1: A parity check error has occurred.	R/(W)*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

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

When parity checking is enabled, the RAMERR flag is set to 1 if a parity check error is detected. The RAM error interrupt request is also generated at this time.
When parity checking is disabled, the RAMERR flag is not set to 1 because no parity check error is detected.
Writing 0 to the RAMERR flag clears the RAM error interrupt request corresponding to the parity check error.

53.2.3 RAM Error Address Capture Register (RAMECAD)

Address(es): 0008 1208h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	READ		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b3	READ	Error Address	The address where an error is found is read.	R
b31 to b19	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

When parity checking is enabled, this register will hold the address where a parity check error was found.
The address of the 8-byte boundary below the location where the error was found is stored in this register at the same time the RAMSTS.RAMERR flag is set to 1.
The error address is not updated when the RAMERR flag is 1 (error has occurred). Its value does not change when parity checking is disabled because no parity check error is detected.
The RAMECAD register is initialized only by a reset.

53.2.4 RAM Protection Register (RAMPRCR)

Address(es): 0008 1204h

	b7	b6	b5	b4	b3	b2	b1	b0
	KW[6:0]							RAMP RCR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RAMPRCR	RAMMODE Register Write Control	0: Disables writing to the RAMMODE register. 1: Enables writing to the RAMMODE register.	R/W
b7 to b1	KW[6:0]	Write Key Word	Enables or disables the rewriting of the RAMPRCR register. When rewriting the RAMPRCR register, write 1111000b to the KW[6:0] bits.	R/W

Writing 1 to the RAMPRCR bit is possible when KW[6:0] = 1111000b. Otherwise writing to RAMPRCR clears the bit to 0. The value of KW[6:0] is read as 0000000b.
The targets for write protection by the RAMPRCR register is the RAM operating mode control register (RAMMODE). Once the RAMPRCR bit is set to 1, writing to RAMMODE register is enabled until the RAMPRCR bit is cleared to 0. Clear the RAMPRCR bit to 0 after writing to RAMMODE register.

53.3 Operation

53.3.1 Parity Checking

Enabling and disabling of parity checking for the RAM can be selected through the RAMMODE register setting. In the initial state, parity checking is disabled. Even parity checking is used in this device.
1-bit parity check code is added to each 1-byte data for writing, and the parity is checked for reading.
If a 1-bit error is detected in the 1 byte when the parity is checked for reading, a RAM error interrupt can be generated. If a 2-bit error or more is detected in the 1 byte, errors cannot be correctly detected.
After power-on, parity check code is undefined until data is written. To use parity checking, write the initial value to all areas while parity checking is enabled before accessing to the RAM immediately after a reset.
Operation cannot be guaranteed if access is made to an area where the initial value is not written.

53.3.2 RAM Error Interrupt Function

A RAM error interrupt is generated when the RAMSTS.RAMERR bit that indicates a parity check error has been changed to 1 while parity checking is enabled.
Writing 0 to the bit clears the RAM interrupt.

53.3.3 Interrupt Source

Of the RAM interrupt sources, that due to the detection of an error through parity checking can be used as either a non-maskable interrupt or a maskable interrupt. For details, refer to section 15, Interrupt Controller (ICUE).

Table 53.2 RAM Interrupt Source

Name	Interrupt Source	DTC Activation	DMAC Activation
RAMERR	RAM error	Not possible	Not possible

53.4 Usage Notes

53.4.1 Low Power Consumption Function

Power consumption can be reduced by setting module stop control register C (MSTPCRC) to stop supply of the clock signal to the RAM.
Setting the MSTPCRC.MSTPC0 bit to 1 stops supply of the clock signal to the RAM.
Stopping supply of the clock signal places the RAM in the module stop state.
The RAM operates after a reset.
The RAM is not accessible in the module stop state.
Do not allow transitions to the module stop state while accessing to the RAM.
Access to the RAM in the module stop state is prohibited. If access is attempted, correct operation is not guaranteed.
For details on the MSTPCRC register, refer to section 11, Low Power Consumption.

53.4.2 Notes on Using Error Checking of RAM

Data in RAM are undefined when the power is turned on. Therefore, parity check errors occur if the data are read before initialization. The RAM is read in 8-byte (64-bit) units. Initialize it on 8-byte boundaries.
When a program is executed in the RAM with the parity check enabled, initialize the RAM in consideration of possible instruction prefetching by the CPU. Instruction prefetching can be performed up to 32 bytes. The initialization must thus cover extra 24 to 31 bytes from the last address of the program.

53.4.3 Notes on Self-Diagnosis of the RAM

A write buffer is mounted for the RAM. When the same address is read after a write operation, data in the write buffer, rather than in the memory cell of the RAM may be read. When the RAM is self-diagnosed, confirm that the data have been written by following the procedure below so that data will not be read from the write buffer.

(1) Write data to the address targeted for diagnosis.
(2) Write data to an address which is at least 4 addresses away from the that in (1).
(3) Read the data from the address in (1).

54. Standby RAM

This MCU provides an on-chip static RAM that can retain data in deep software standby (standby RAM).

54.1 Overview

Table 54.1 lists the specifications of the standby RAM.

Table 54.1 Specifications of Standby RAM

Item	Description
RAM capacity	4 Kbytes
RAM address	000A 4000h to 000A 4FFFh
Access	• Both read and write operations take 2 or 3 cycles of PCLKB when ICLK ≥ PCLKB; two cycles of ICLK are needed when ICLK < PCLKB. • Enabling or disabling of RAM access is selectable.*1 • Endian conforms to the endian setting of the chip. • Non-aligned access is prohibited. If non-aligned access is attempted, correct operation is not guaranteed.
Data retention function	Data can be retained in deep software standby mode
Low-power consumption function	The module-stop state is selectable.

Note 1. Selectable by the SBYRAME bit in SYSCR1. For details on SYSCR1, refer to section 3.2.3, System Control Register 1 (SYSCR1).

54.2 Operation

54.2.1 Data Retention

Whether or not the supply of internal power to the standby RAM continues in deep software standby mode is selectable by the DPSBYCR.DEEPCUT[1:0] bits.

If continuation of the supply of internal power is selected, data in the standby RAM are retained in deep software standby mode.

Refer to section 11, Low Power Consumption, for details on the DPSBYCR.DEEPCUT[1:0] bits.

54.2.2 Low Power Consumption Function

Power consumption can be reduced by setting module stop control register C (MSTPCRC) to stop supply of the clock signal to the standby RAM.

If the MSTPC7 bit in MSTPCRC is set to 1, supply of the clock signal to the standby RAM is stopped.

The standby RAM is thus placed in the module-stop state by stopping supply of the clock signals. The standby RAM operates after a reset.

The standby RAM is not accessible if it is in the module-stop state. A transition to the module-stop state should not be made while access to the standby RAM is in progress.

For details on the MSTPCRC register, refer to section 11, Low Power Consumption.

55. Flash Memory (FLASH)

This MCU incorporates code flash memory, data flash memory, and option-setting memory. The code flash memory stores instructions and operands, and the data flash memory stores only data. For option-setting memory, refer to section 7, Option-Setting Memory (OFSM).

55.1 Overview

Table 55.1 lists the specifications of the flash memory, and Figure 55.1 is a block diagram of the flash memory related modules.
The I/O pins used in boot mode, see Table 55.17.
The FCU (flash control unit) controls programming and erasure of the flash memory. The FACI (flash application command interface) controls the FCU according to the specified FACI commands.
Regarding the configuration of the code flash memory, see Figure 55.2 to Figure 55.5, and for the configuration of the data flash memory, see Figure 55.6.

Table 55.1 Specifications of Flash Memory (1/2)

Item	Code Flash Memory	Data Flash Memory
Memory capacity	• User area: 2 Mbytes max.	Data area: 8 Kbytes
ROM cache	• Capacity: 8 Kbytes • Mapping method: direct mapping • Line size: 16 bytes	Not available
Read cycle	• While ROM cache operation is enabled: When the cache is hit, one cycle; When the cache is missed, one to two cycles if ICLK ≤ 60 MHz, two to three cycles if ICLK > 60 MHz.*1 • When ROM cache operation is disabled: One cycle if ICLK ≤ 60 MHz Two cycles if ICLK > 60 MHz*1	Reading proceeds in every cycle of FCLK*2
Value after erasure	FFh	Undefined
Programming/erasing method	• Programming and erasing the code flash memory and data flash memory, and programming the option-setting memory are handled by the FACI commands specified in the FACI command issuing area (007E 0000h) (self-programming). • Programming/erasure through transfer by a serial-programmer via a serial interface (serial programming)	
Security function	Protects against illicit tampering with or reading out of data in flash memory	
Protection	Protects against erroneous rewriting of the flash memory	
Dual bank function	The dual-bank structure makes a safe update possible in cases where programming is suspended. • Linear mode: the code flash memory is used as one area • Dual mode: the code flash memory is divided into two areas	Not available
Trusted memory (TM) function	Protects against illicit reading of the code flash memory • Linear mode: blocks 8 and 9 • Dual mode: blocks 8, 9, 46, and 47	Not available
Background operations (BGOs)	• The code flash memory can be read while the code flash memory is being programmed or erased.*3 • The data flash memory can be read while the code flash memory is being programmed or erased. • The code flash memory can be read while the data flash memory is being programmed or erased.	

Table 55.1 Specifications of Flash Memory (2/2)

Item	Code Flash Memory	Data Flash Memory
Units of programming and erasure	• Units of programming for the user area: 128 bytes • Units of erasure for the user area: Block units	• Unit of programming for the data area: 4 bytes • Unit of erasure for the data area: 64/128/256 bytes
Other functions	Interrupts can be accepted during self-programming.	
	In the initial settings of this MCU, an expansion area of the option-setting memory can be set.	
On-board programming (Serial programming/Self-programming)	Programming/erasure in boot mode (SCI interface) • The asynchronous serial interface (SCI1) is used. • The transfer rate is adjusted automatically. Programming/erasure in boot mode (USB interface) • USB0 is used. • Dedicated hardware is not required, so direct connection to a PC is possible. Programming/erasure in boot mode (FINE interface) • FINE is used Programming/erasure in single-chip mode or on-chip ROM enabled extended mode • Programming or erasure by a routine for writing to code flash memory or data flash memory within the user program is possible.	
Off-board programming	Programming and erasure of the code flash memory and option-setting memory by using a parallel programmer is possible.	Programming or erasure of the data flash memory by a parallel programmer is not possible.
Unique ID	A 16-byte ID code provided for each MCU	

Note 1. If the frequency of ICLK is above 60 MHz, the setting of the ROMWT register must be changed. For details, see section 9.2.2, ROM Wait Cycle Setting Register (ROMWT).
Note 2. The required number of cycles for access must be set according to the FCLK frequency. For details, refer to section 55.4.22, Data Flash Memory Access Frequency Setting Register (EEPFCLK).
Note 3. Limitations apply to the combinations of the address ranges for programming/erasure process and reading process: see Table 55.25.

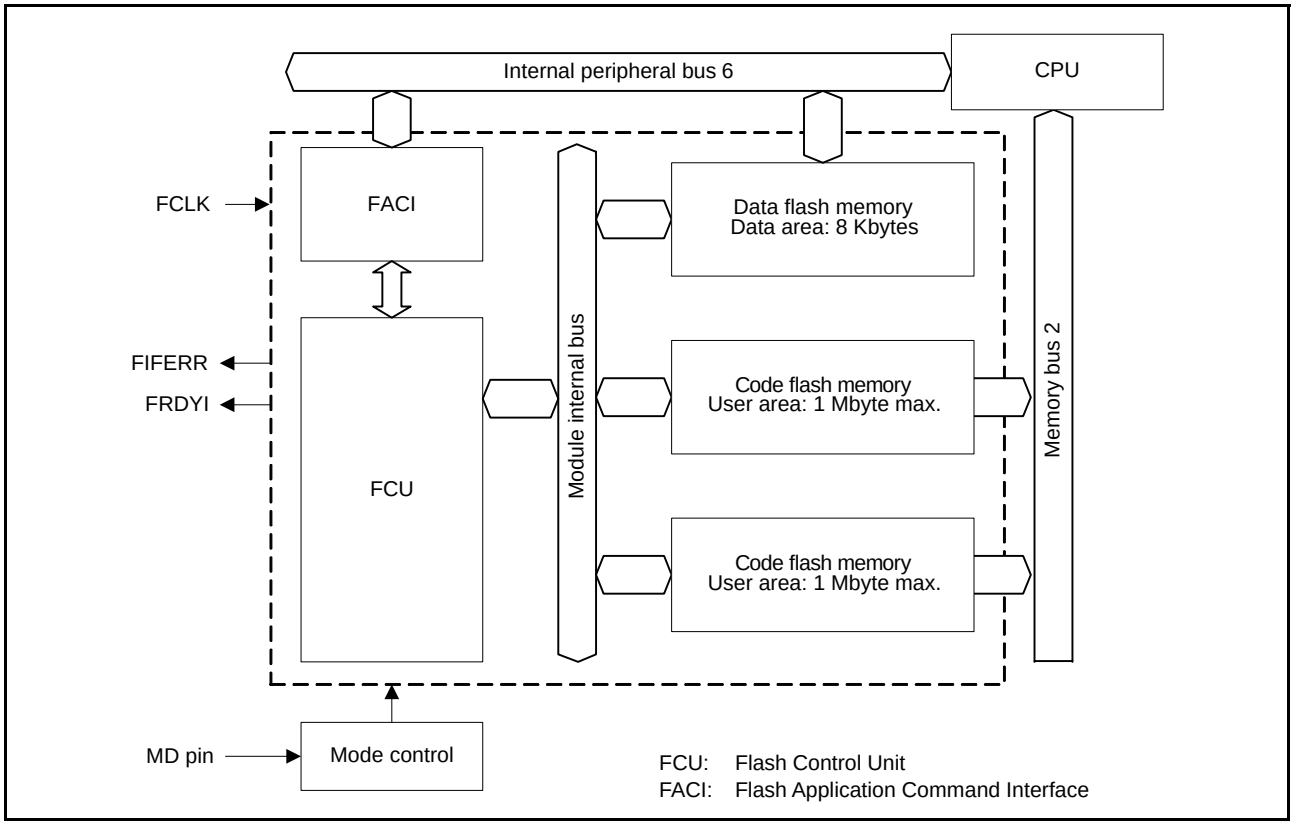


Figure 55.1 Block Diagram of Flash Memory Related Modules

55.2 Hardware Interface Area

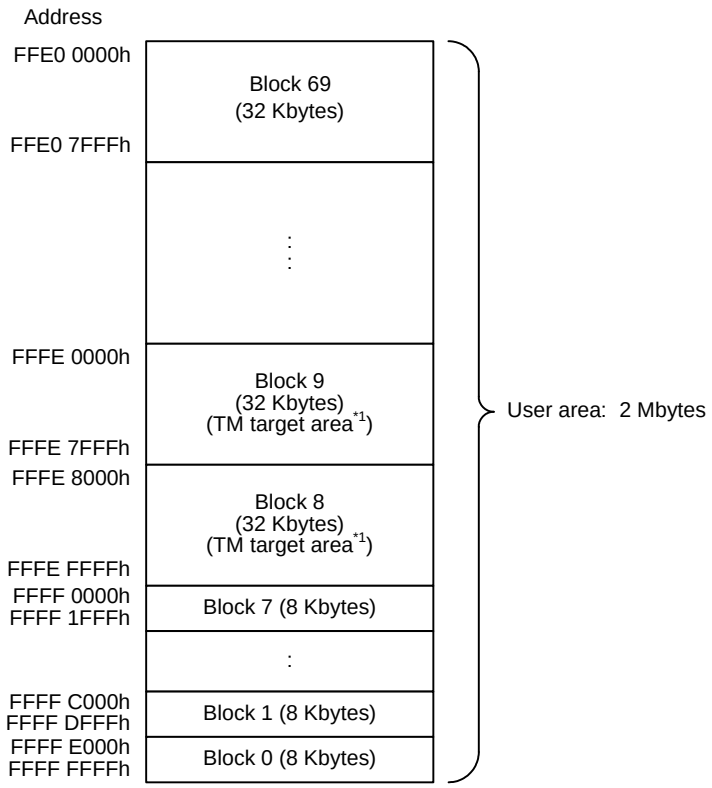
Using the hardware interface with the flash memory requires accessing to the area containing registers of the hardware, that for the issuing of FACI commands. Table 55.2 summarizes information on all of these areas.

Table 55.2 Information on the Hardware Interface Area

Area	Address	Capacity
Area containing the various registers of the hardware	See section 55.4, Register Descriptions.	See section 55.4, Register Descriptions.
FACI command-issuing area	007E 0000h	4 bytes
Option-setting memory (configuration setting area)	FE7F 5D00h to FE7F 5D7Fh	128 bytes

55.3 Structure of Memory

Figure 55.2 shows the memory map of code flash memory in linear mode. Figure 55.3 to Figure 55.5 show the memory map of code flash memory in dual mode. The code flash memory can be used as 2 bank areas by using the dual bank function. This dual-bank structure allows a safe update of a program while a user program is running.
The user area of the code flash memory in this MCU is divided into 8- and 32-Kbyte blocks, which serve as the units of erasure.
When the TM function is enabled in linear mode, blocks 8 and 9 are the TM target areas. The user area is available as areas for storing the user program.

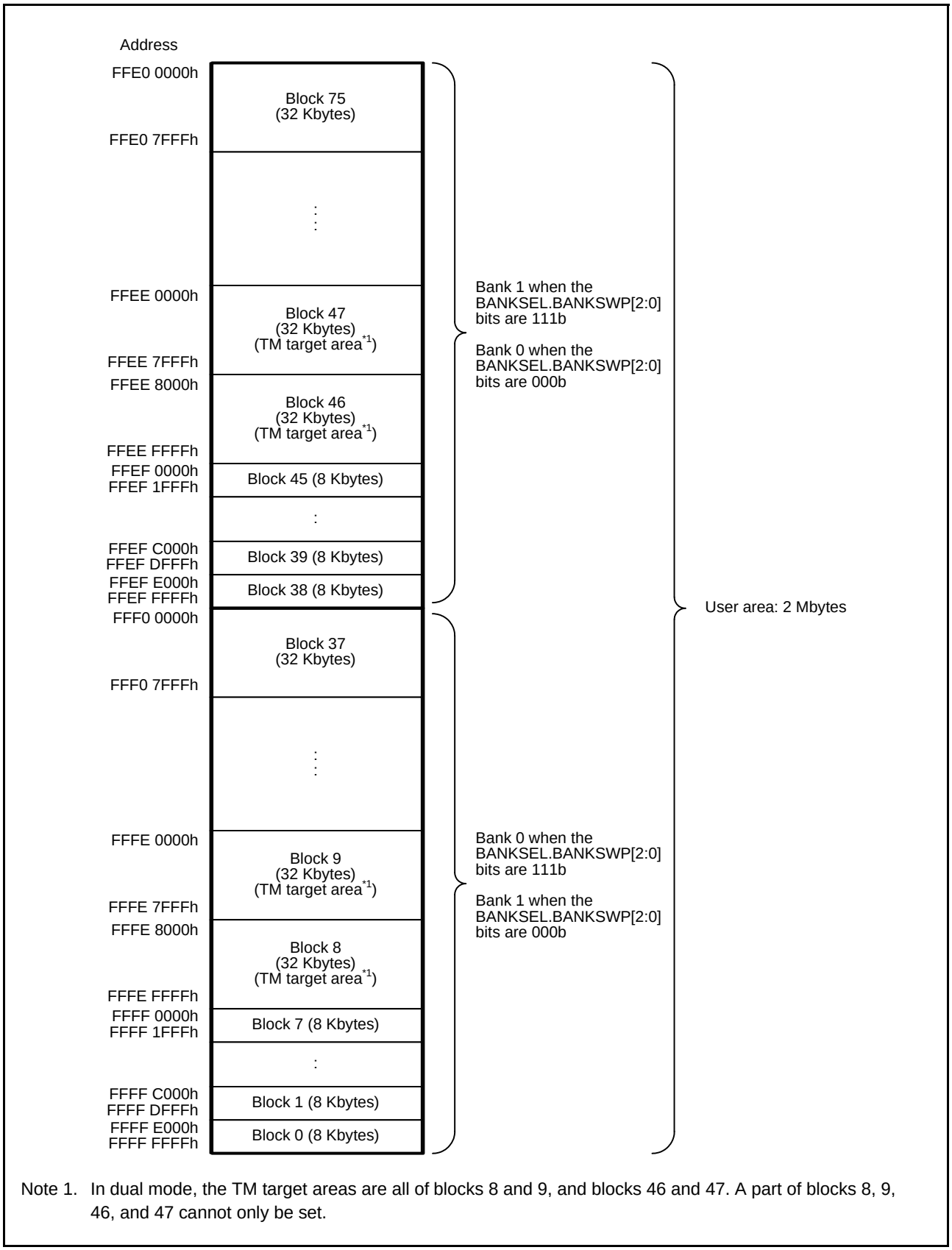


Code Flash Memory Capacity	Address	Number of Blocks
2 Mbytes	FFE0 0000h to FFFF FFFFh	0 to 69
1.5 Mbytes	FFE8 0000h to FFFF FFFFh	0 to 53
1 Mbyte	FFF0 0000h to FFFF FFFFh	0 to 37

Note 1. Both blocks 8 and 9 are the target areas for handling as TM. Setting only block 8 or only block 9 is not allowed.

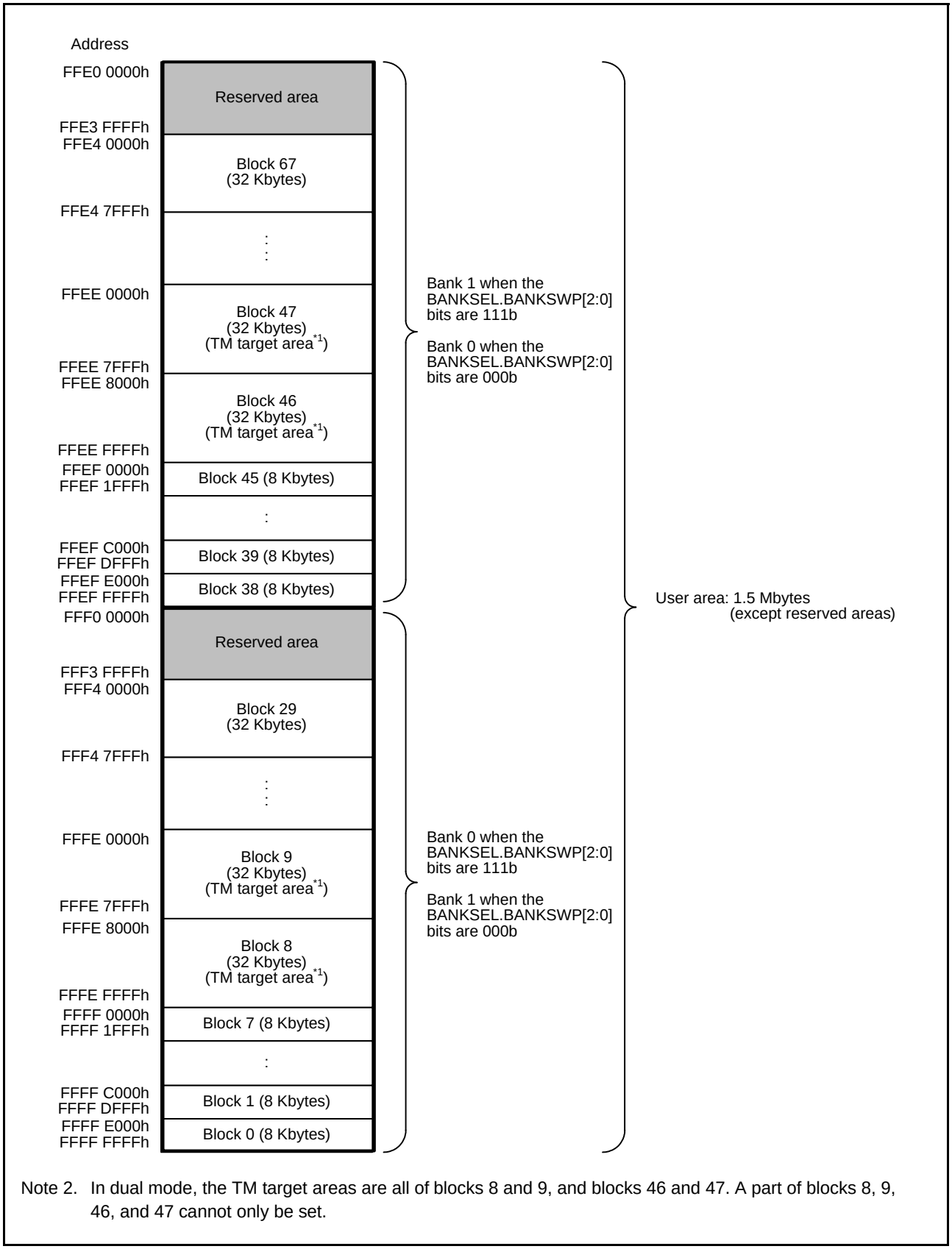
Figure 55.2 Map of the Code Flash Memory in Linear Mode

When the TM function is enabled in dual mode, blocks 8 and 9, and blocks 46 and 47 are the TM target areas. The user area can be used as storage area of the user program.



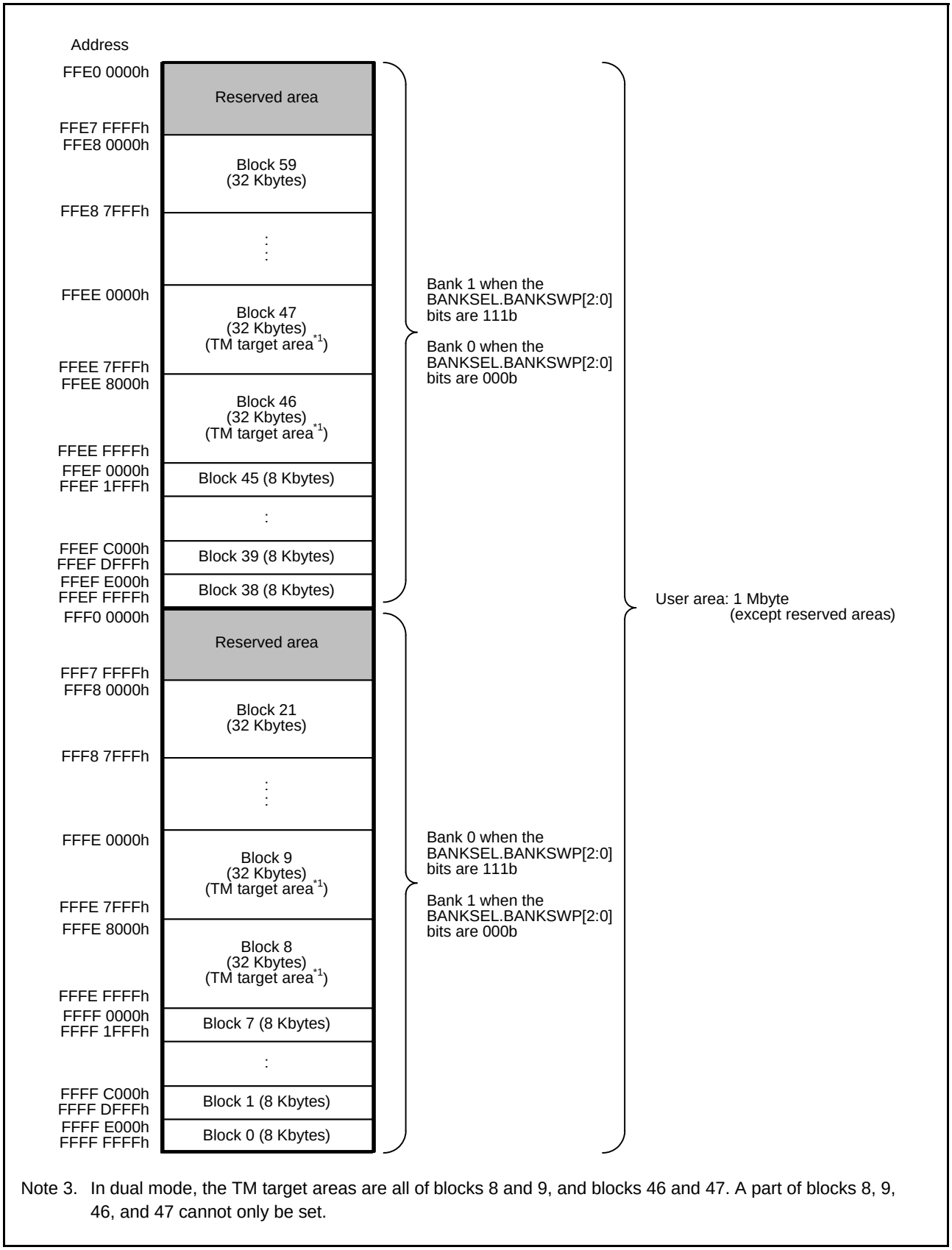
Note 1. In dual mode, the TM target areas are all of blocks 8 and 9, and blocks 46 and 47. A part of blocks 8, 9, 46, and 47 cannot only be set.

Figure 55.3 Map of the Code Flash Memory in Dual Mode (Products with 2 Mbytes of code flash memory)



Note 2. In dual mode, the TM target areas are all of blocks 8 and 9, and blocks 46 and 47. A part of blocks 8, 9, 46, and 47 cannot only be set.

Figure 55.4 Map of the Code Flash Memory in Dual Mode (Products with 1.5 Mbytes of code flash memory)



Note 3. In dual mode, the TM target areas are all of blocks 8 and 9, and blocks 46 and 47. A part of blocks 8, 9, 46, and 47 cannot only be set.

Figure 55.5 Map of the Code Flash Memory in Dual Mode (Products with 1 Mbyte of code flash memory)

The data area of the data flash memory in this MCU is divided into 64-byte blocks, with each being a unit for erasure. Figure 55.6 shows the mapping of the data flash memory.

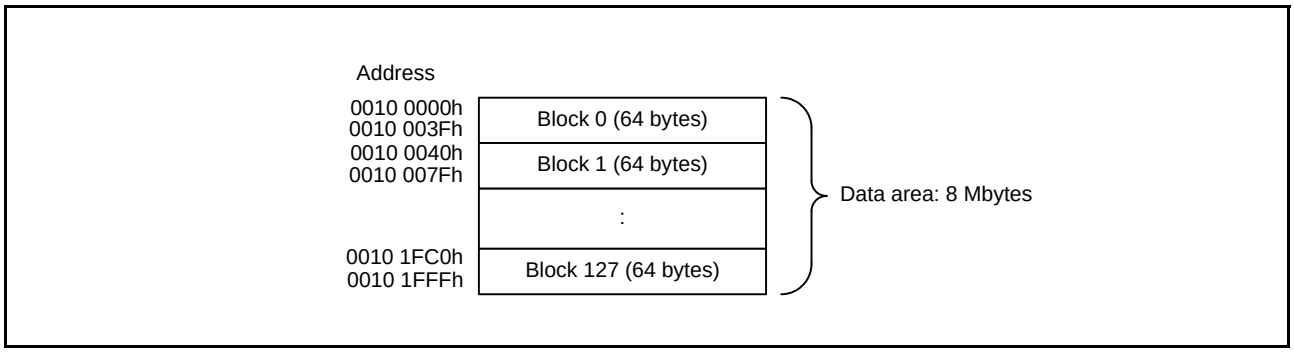


Figure 55.6 Map of the Data Flash Memory

55.4 Register Descriptions

55.4.1 ROM Cache Enable Register (ROMCE)

Address(es): FLASH.ROMCE 0008 1000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ROMCEN
Value after 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	ROMCEN	ROM Cache Enable	0: ROM cache is disabled. 1: ROM cache is enabled.	R/W
b15 to b1	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R

ROMCEN Bit (ROM Cache Enable)

When setting the ROMCEN bit to 1, the ROM cache starts operation. While ROM cache is enabled, the data is served from the cache when a cache hit occurs. Using ROM caching helps to reduce power consumption by the MCU.
Set a non-cacheable area while the ROMCEN bit is 0.

55.4.2 ROM Cache Invalidate Register (ROMCIV)

Address(es): FLASH.ROMCIV 0008 1004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ROMCIV
Value after 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	ROMCIV	ROM Cache Invalidate	When reading 0: Invalidation is not started/Invalidation is completed. 1: Invalidation is in progress. When writing Writing 1 invalidates the cache line. Writing 0 has no effect.	R/W
b15 to b1	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R

ROMCIV Bit (ROM Cache Invalidate)

When writing 1 to the ROMCIV bit, the ROM cache invalidation is performed.

55.4.3 Non-Cacheable Area n Address Register (NCRGn) (n = 0, 1)

Address(es): FLASH.NCRG0 0008 1040h, FLASH.NCRG1 0008 1048h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0

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

The NCRGn register specifies the address where an area in which caching is disabled starts (non-cacheable area).
The higher-order 10 bits (b31 to b22), which are fixed to 1 and the lower-order 4 bits (b3 to b0), which are fixed to 0, are reserved. When writing, write 1 and 0 to the respective sets of bits.
Set the NCRGn register while the ROMCE.ROMCEN bit is 0.

55.4.4 Non-Cacheable Area n Setting Register (NCRCn) (n = 0, 1)

Address(es): FLASH.NCRC0 0008 1044h, FLASH.NCRC1 0008 104Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	NCSZ[16:12]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	NCSZ[11:0]												NC3E	NC2E	NC1E	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b1	NC1E	IF Non-Cacheable Area Specification Enable	This bit selects whether an area of memory is non-cacheable by the IF cache. 0: Disabled 1: Enabled	R/W
b2	NC2E	OA Non-Cacheable Area Specification Enable	This bit selects whether an area of memory is non-cacheable by the OA cache. 0: Disabled 1: Enabled	R/W
b3	NC3E	DM Non-Cacheable Area Specification Enable	This bit selects whether an area of memory is non-cacheable by the DM cache. 0: Disabled 1: Enabled	R/W
b20 to b4	NCSZ[16:0]	Non-Cacheable Area Size Specification	These bits specify the size of the given non-cacheable area. b20 b4 0 0000 0000 0000 0000: 16 bytes 0 0000 0000 0000 0001: 32 bytes 0 0000 0000 0000 0011: 64 bytes 0 0000 0000 0000 0111: 128 bytes 0 0000 0000 0000 1111: 256 bytes 0 0000 0000 0001 1111: 512 bytes 0 0000 0000 0011 1111: 1 Kbyte 0 0000 0000 0111 1111: 2 Kbytes 0 0000 0000 1111 1111: 4 Kbytes 0 0000 0001 1111 1111: 8 Kbytes 0 0000 0011 1111 1111: 16 Kbytes 0 0000 0111 1111 1111: 32 Kbytes 0 0000 1111 1111 1111: 64 Kbytes 0 0001 1111 1111 1111: 128 Kbytes 0 0011 1111 1111 1111: 256 Kbytes 0 0111 1111 1111 1111: 512 Kbytes 0 1111 1111 1111 1111: 1 Mbyte Settings other than above are prohibited.	R/W
b31 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R

The NCRCn register specifies enabling or disabling of non-cacheable areas n (n = 0, 1) and the size of the area. Write to the register while the ROMCE.ROMCEN bit is 0.

NC1E Bit (IF Non-Cacheable Area Specification Enable)

The NC1E bit enables or disables the settings of the non-cacheable area in the cache (IF cache) used for fast instruction fetching (IF) by the CPU.

NC2E Bit (OA Non-Cacheable Area Specification Enable)

The NC2E bit enables or disables the settings of the non-cacheable area in the cache (OA cache) used for fast operand access (OA) by the CPU.

NC3E Bit (DM Non-Cacheable Area Specification Enable)

The NC3E bit enables or disables the settings of the non-cacheable area in the cache (DM cache) used for fast access to data by bus masters other than the CPU.

NCSZ[16:0] Bit (Non-Cacheable Area Size Specification)

The NCSZ[16:0] bits specify the size of the non-cacheable area. They are used to mask address bits 4 to 20.
When an address generated by the bus master is compared to the value of the NCRGn register, bits corresponding to NCSZ[16:0] bits with the value 1 are ignored.

55.4.5 Flash P/E Protect Register (FWEPROR)

Address(es): FLASH.FWEPROR 0008 C296h

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

Bit	Symbol	Bit Name	Description	R/W
b1, b0	FLWE[1:0]	Flash Programming and Erasure Enabling	b1 b0 0 0: Prohibits programming/erasure, and blank checking. 0 1: Permits programming/erasure, and blank checking. 1 0: Prohibits programming/erasure, and blank checking. 1 1: Prohibits programming/erasure, and blank checking.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Programming/erasure of the flash memory, and blank checking are prohibited or permitted by hardware.
FWEPROR is initialized by a reset due to the signal on the RES# pin, a power-on reset, a voltage-monitoring 0 reset, an independent watchdog timer reset, a watchdog timer reset, a voltage-monitoring 1 reset, a voltage-monitoring 2 reset, and a software reset, and by transitions to software standby and deep software standby.

55.4.6 Flash Access Status Register (FASTAT)

Address(es): FLASH.FASTAT 007F E010h

	b7	b6	b5	b4	b3	b2	b1	b0
	CFAE	—	—	CMDLK	DFAE	—	—	—
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	DFAE	Data Flash Memory Access Violation Flag	0: No data flash memory access violation has occurred. 1: A data flash memory access violation has occurred.	R/W*1
b4	CMDLK	Command Lock Flag	0: The flash sequencer is not in the command-locked state. 1: The flash sequencer 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	CFAE	Code Flash Memory Access Violation Flag	0: No code flash memory access violation has occurred. 1: A code flash memory access violation has occurred.	R/W*1

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

This register indicates whether a code flash memory or data flash memory access violation has occurred. If either of the CFAE, and DFAE flags is 1, the CMDLK flag is set to 1 and the flash sequencer enters the command-locked state (see section 55.10.2, Error Protection). To release it from the command-locked state, a status clear command or forced stop command must be issued by the FACI.

DFAE Flag (Data Flash Memory Access Violation Flag)

This flag indicates whether a data flash memory access violation has occurred. If this flag is set to 1, the FSTATR.ILGLERR flag is set to 1, placing the flash sequencer in the command-locked state.

[Setting Condition]

- See Table 55.16, Error Protection Type

[Clearing Condition]

- When 0 is written after reading of 1
- When a status clear command or forced stop command is issued

CMDLK Flag (Command Lock Flag)

This flag indicates that the flash sequencer is in the command-locked state.

[Setting Condition]

- When the flash sequencer detects any of errors listed in Table 55.16, Error Protection Type and transitions to the command-locked state

[Clearing Condition]

- When the flash sequencer starting to process a status clear or forced stop command

CFAE Flag (Code Flash Memory Access Violation Flag)

This flag indicates whether a code flash memory access violation has occurred. If this flag is set to 1, the FSTATR.ILGLERR flag is set to 1, placing the flash sequencer in the command-locked state.

[Setting Conditions]

- See Table 55.16, Error Protection Type

[Clearing Condition]

- When 0 is written after reading of 1

- When a status clear command or forced stop command is issued

55.4.7 Flash Access Error Interrupt Enable Register (FAEINT)

Address(es): FLASH.FAEINT 007F E014h

	b7	b6	b5	b4	b3	b2	b1	b0
	CFAEIE	—	—	CMDLKIE	DFAEIE	—	—	—
Value after reset:	1	0	0	1	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	DFAEIE	Data Flash Memory Access Violation Interrupt Enable	0: Generation of an FIFERR interrupt request is disabled when FASTAT.DFAE is set to 1. 1: Generation of an FIFERR interrupt request is enabled when FASTAT.DFAE is set to 1.	R/W
b4	CMDLKIE	Command Lock Interrupt Enable	0: Generation of an FIFERR interrupt request is disabled when FASTAT.CMDLK is set to 1. 1: Generation of an FIFERR interrupt request is enabled when FASTAT.CMDLK is set to 1.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	CFAEIE	Code Flash Memory Access Violation Interrupt Enable	0: Generation of an FIFERR interrupt request is disabled when FASTAT.CFAE is set to 1. 1: Generation of an FIFERR interrupt request is enabled when FASTAT.CFAE is set to 1.	R/W

This register enables or disables generation of a flash access error (FIFERR) interrupt request.

DFAEIE Bit (Data Flash Memory Access Violation Interrupt Enable)

This bit enables or disables generation of an FIFERR interrupt request when a data flash memory access violation occurs and the FASTAT.DFAE flag is set to 1.

CMDLKIE Bit (Command Lock Interrupt Enable)

This bit enables or disables generation of an FIFERR interrupt request when the flash sequencer enters the command-locked state and the FASTAT.CMDLK flag is set to 1.

CFAEIE Bit (Code Flash Memory Access Violation Interrupt Enable)

This bit enables or disables generation of an FIFERR interrupt request when a code flash memory access violation occurs and the FASTAT.CFAE flag is set to 1.

55.4.8 Flash Ready Interrupt Enable Register (FRDYIE)

Address(es): FLASH.FRDYIE 007F E018h

	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: Generation of an FRDY interrupt request is disabled. 1: Generation of an FRDY interrupt request is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register enables or disables generation of a flash ready (FRDY) interrupt request.

FRDYIE Bit (Flash Ready Interrupt Enable)

This bit is used to enable or disable generation of an FRDY interrupt request when the FASTAT.FRDY flag is changed from 0 to 1 upon completion of processing by the flash sequencer of programming, erasure, and blank checking command.

55.4.9 FACI Command Start Address Register (FSADDR)

Address(es): FLASH.FSADDR 007F E030h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	FSADDR[31:16]															
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
	FSADDR[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
b31 to b0	FSADDR[31:0]	Start Address for FACI Command Processing	Start Address for FACI Command Processing	R/W*1

Note 1. Writing to these bits is possible only when the FSTATR.FRDY flag is 1. Writing to these bits while the FSTATR.FRDY flag = 0 is ignored. Note that b0 and b1 are read-only.

This register specifies the address where the target area for command processing starts when the FACI command for programming, block erasure, multi-block erasure, blank checking, or configuration setting is issued.
The FSADDR register is initialized when the FSUINITR.SUINIT bit is set to 1. It is also initialized by a reset.

FSADDR[31:0] Bits (Start Address for FACI Command Processing)

These bits specify the start address for FACI command processing. Bits 31 to 24 are ignored in FACI command processing for the code flash memory. Bits 31 to 17 are ignored in FACI command processing for the data flash memory. Bits 31 to 10 are ignored in processing of the FACI command for the option-setting memory. Bits that do not reach the address boundaries are also ignored. Table 55.3 shows the address boundary for each command.

Table 55.3 Address Boundary for Each of the Commands

Command	Address Boundary
Programming (code flash memory)	128-byte
Programming (data flash memory)	4-byte
Block erase (code flash memory)	8-Kbyte or 32-Kbyte
Block erase (data flash memory)	64-byte
Multi-block erase (data flash memory)	64-, 128-, or 256-byte
Blank check (data flash memory)	4-byte
Configuration setting	16-byte

See Table 55.15, Address Used by Configuration Set Command for the start address of the option-setting memory (configuration setting area).

55.4.10 FACI Command End Address Register (FEADDR)

Address(es): FLASH.FEADDR 007F E034h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	FEADDR[31:16]															
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
	FEADDR[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
b31 to b0	FEADDR[31:0]	End Address for FACI Command Processing	The end address for FACI command processing	R/W*1

Note 1. Writing to these bits is possible only when the FSTATR.FRDY flag is 1. Writing to these bits while the FSTATR.FRDY flag = 0 is ignored. Note that b0 and b1 are read-only.

This register is used to specify the end address of the area targeted for the multi-block erase command and blank checking command handling. When executing the multi-block erase command, the setting value of the register must be the setting of the FEADDR register or lower. If the setting value of the FSADDR register is larger than the FEADDR register setting value, the flash sequencer enters the command locked state.
When the FBCCNT.BCDIR bit is 0, the value of the FSADDR register must be that of this register or less. When the FBCCNT.BCDIR bit is 1, the value of the FSADDR register must be at least that of this register. If the settings of the FBCCNT.BCDIR bit and the FSADDR and FEADDR registers are inconsistent with the above rules, the flash sequencer enters the command-locked state (see section 55.10.2, Error Protection).
The FEADDR register is initialized when the FSUINITR.SUINIT bit is set to 1. It is also initialized by a reset.

FEADDR[31:0] Bits (End Address for FACI Command Processing)

These bits are used to specify the end address for handling of the multi-block erase command and blank checking command. In command processing, bits 31 to 17 and any bits that do not reach the address boundaries listed in Table 55.3, Address Boundary for Each of the Commands are ignored.

55.4.11 Flash Status Register (FSTATR)

Address(es): FLASH.FSTATR 007F E080h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	ILGCOMERR	FESETERR	SECERR	OTERR	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FRDY	ILGLERR	ERSERR	PRGERR	SUSRDY	DBFULL	ERSSPD	PRGSPD	—	FLWEERR	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	FLWEERR	Flash Write/Erase Protect Error Flag	0: An error has not occurred. 1: An error has occurred.	R
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b8	PRGSPD	Programming Suspend Status Flag	0: The flash sequencer is in a state other than those corresponding to the value 1. 1: The flash sequencer is in the programming suspension processing state or the programming suspended state.	R
b9	ERSSPD	Erasure Suspend Status Flag	0: The flash sequencer is in a state other than those corresponding to the value 1. 1: The flash sequencer is in the erasure suspension processing state or the erasure-suspended state.	R
b10	DBFULL	Data Buffer Full Flag	0: The data buffer is empty. 1: The data buffer is full.	R
b11	SUSRDY	Suspend Ready Flag	0: The flash sequencer cannot receive P/E suspend commands. 1: The flash sequencer can receive P/E suspend commands.	R
b12	PRGERR	Programming Error Flag	0: Programming has been completed successfully. 1: An error has occurred during programming.	R
b13	ERSERR	Erasure Error Flag	0: Erasure has been completed successfully. 1: An error has occurred during erasure.	R
b14	ILGLERR	Illegal Error Flag	0: The flash sequencer has not detected an illegal FACI command or illegal flash memory access. 1: The flash sequencer has detected an illegal FACI command or illegal flash memory access.	R
b15	FRDY	Flash Ready Flag	0: Programming, block erase, multi-block erase, P/E suspend, P/E resume, forced stop, blank check, or configuration setting command processing is in progress. 1: None of the above is in progress.	R
b19 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20	OTERR	Other Error Flag	0: An error has not occurred. 1: An error has occurred.	R
b21	SECERR	Security Error Flag	0: Write protection by the FAW.FSPR bit is not violated 1: Write protection by the FAW.FSPR bit is violated	R
b22	FESETERR	FENTRY Setting Error Flag	0: A setting error in the FENTRYR register has not been detected. 1: A setting error in the FENTRYR register has been detected.	R
b23	ILGCOMERR	Illegal Command Error Flag	0: The flash sequencer has not detected an illegal FACI command error. 1: The flash sequencer has detected an illegal FACI command error.	R
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register indicates the state of the flash sequencer.

FLWEERR Flag (Flash Write/Erase Protect Error Flag)

This flag indicates a violation of the flash memory overwrite protection setting in the FWEPROR register. When this flag is 1, the flash sequencer is in the command-locked state.

[Setting Condition]

- See Table 55.16, Error Protection Type

[Clearing Condition]

- When the flash sequencer starts processing of a forced stop command

PRGSPD Flag (Programming Suspend Status Flag)

This flag indicates that the flash sequencer is in the processing of suspension of programming or has transitioned to the programming suspended state.

[Setting Condition]

- When the flash sequencer starts processing in response to a programming suspend command

[Clearing Conditions]

- When the flash sequencer has received a P/E resume command
  (after write access to the FACI command-issuing area is completed)
- When the flash sequencer starts processing of a forced stop command

ERSSPD Flag (Erasure Suspend Status Flag)

This flag indicates that the flash sequencer is the processing of erasure suspension or has transitioned to the erasure suspended state.

[Setting Condition]

- When the flash sequencer starts processing in response to an erasure suspend command

[Clearing Conditions]

- Reception of the P/E resume command by the flash sequencer
  (after write access to the FACI command-issuing area is completed)
- When the flash sequencer starts processing of a forced stop command

DBFULL Flag (Data Buffer Full Flag)

This flag indicates the state of the data buffer when a programming command or configuration setting command is issued. The FACI incorporates a buffer for write data (data buffer). When data for writing to the flash memory are issued to the FACI command-issuing area while the data buffer is full, the FACI inserts a wait cycle in the peripheral bus 6.

[Setting Condition]

- When the data buffer becomes full while a programming command or configuration setting command is being issued

[Clearing Condition]

- When the data buffer becomes empty

SUSRDY Flag (Suspend Ready Flag)

This flag indicates whether the flash sequencer can receive a P/E suspend command.

[Setting Condition]

- After starting programming/erasure processing and when the flash sequencer enters a state in which P/E suspend commands can be received

[Clearing Conditions]

- When the flash sequencer has accepted the P/E suspend command or forced stop command (after write access to the

FACI command-issuing area is completed)

- When the flash sequencer enters the command-locked state during programming or erasure
- When programming or erasure has been completed

PRGERR Flag (Programming Error Flag)

This flag indicates the result of programming of the flash memory. If this flag is 1, the flash sequencer is in the command-locked state.

[Setting Condition]

- See Table 55.16, Error Protection Type

[Clearing Condition]

- When the flash sequencer starts processing of a status clear or forced stop command

ERSERR Flag (Erasure Error Flag)

This flag indicates the result of erasure of the flash memory. If this flag is 1, the flash sequencer is in the command-locked state.

[Setting Condition]

- See Table 55.16, Error Protection Type

[Clearing Condition]

- When the flash sequencer starts processing of a status clear or forced stop command

ILGLERR Flag (Illegal Error Flag)

This flag indicates that the flash sequencer has detected an illegal FACI command or flash memory access. When this flag is 1, the flash sequencer is in the command-locked state.

[Setting Condition]

- See Table 55.16, Error Protection Type

[Clearing Condition]

- When the flash sequencer starts processing of a status clear or forced stop command.

FRDY Flag (Flash Ready Flag)

This flag indicates the command processing state of the flash sequencer.

[Setting Conditions]

- When the flash sequencer completes command processing
- When the flash sequencer receives a P/E suspend command and suspends programming or erasure of the flash memory
- When the flash sequencer has received a forced stop command and ended command processing

[Clearing Conditions]

- When the flash sequencer receives the FACI command of the setting of the program and configuration and after the first write access is made to the FACI command-issuing area
- When the flash sequencer receives any FACI command other than of the setting of the program and configuration and after the last write access is made to the FACI command issuing area

OTERR Flag (Other Error Flag)

This flag indicates that an FACI command has been issued when the condition of accepting commands is not satisfied. The OTERR flag is 1, the flash sequencer is in the command-locked state.

[Setting Condition]

- See Table 55.16, Error Protection Type

[Clearing Condition]

- When the flash sequencer starts processing of a status clear or forced stop command

SECERR Flag (Security Error Flag)

This flag indicates that write protection by the FAW.FSPR bit is violated. When the SECERR flag is 1, the flash sequencer is in the command-locked state.

[Setting Condition]

- See Table 55.16, Error Protection Type

[Clearing Condition]

- When the flash sequencer starts processing of a status clear or forced stop command

FESETERR Flag (FENTRY Setting Error Flag)

This flag indicates that a value of AA81h is written in the FENTRYR register or the value in the FENTRYR register differs when P/E is suspended and resumed. When the FESETERR flag is 1, the flash sequencer is in the command-locked state.

[Setting Condition]

- See Table 55.16, Error Protection Type

[Clearing Condition]

- When the flash sequencer starts processing of a status clear or forced stop command

ILGCOMERR Flag (Illegal Command Error Flag)

This flag indicates that the flash sequencer has detected an illegal FACI command. When the ILGCOMERR flag is 1, the flash sequencer is in the command-locked state.

[Setting Condition]

- See Table 55.16, Error Protection Type

[Clearing Condition]

- When the flash sequencer starts processing of a status clear or forced stop command

55.4.12 Flash P/E Mode Entry Register (FENTRYR)

Address(es): FLASH.FENTRYR 007F E084h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								FENTRYD	—	—	—	—	—	—	FENTRYC
Value after 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	FENTRYC	Code Flash Memory P/E Mode Entry	0: Code flash memory is in read mode. 1: Code flash memory is in P/E mode.	R/W*1, *2
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	FENTRYD	Data Flash Memory P/E Mode Entry	0: Data flash memory is in read mode. 1: Data flash memory is in P/E mode.	R/W*1, *2
b15 to b8	KEY[7:0]	Key Code	Key code	R/W*3

Note 1. Writing to these bits is possible only when the FSTATR.FRDY flag is 1. Writing to these bits while the FSTATR.FRDY flag = 0 is ignored.
Note 2. Writing to this bit is possible only when AAh is written to the KEY bits in 16-bit units.
Note 3. Written values are not retained by these bits. These bits are read as 0.

This register is used to specify code flash memory P/E mode and data flash memory P/E mode. To specify code flash memory P/E mode or data flash memory P/E mode so that the flash sequencer can receive FACI commands, set either the FENTRYD or FENTRYC bit to 1 to place the flash sequencer in P/E mode.
Note that writing AA81h in this register causes the FSTATR.ILGLERR and FSTATR.FESETERR flags to be set to 1, and the flash sequencer to enter the command-locked state.
The FENTRYR register is initialized when the FSUINITR.SUINIT bit is set to 1. It is also initialized by a reset.

FENTRYC Bit (Code Flash Memory P/E Mode Entry)

This bit specifies the P/E mode for code flash memory.
[Setting Condition]

- When 1 is written to the FENTRYC bit while writing to the FENTRYR register is enabled and the FENTRYR register is 0000h

[Clearing Conditions]

- When the FENTRYR register is accessed in 8-bit units while the FSTATR.FRDY flag is 1
- When a value other than AAh is specified in the KEY bits and the FENTRYR register is accessed in 16-bit units while the FSTATR.FRDY flag is 1
- When 0 is written to the FENTRYC bit while writing to the FENTRYR register is enabled
- When the FENTRYR register is written while writing to the FENTRYR register is enabled and the value of the FENTRYR register is other than 0000h

FENTRYD Bit (Data Flash Memory P/E Mode Entry)

This bit specifies the P/E mode for data flash memory.
[Setting Condition]

- When 1 is written to the FENTRYR.FENTRYD bit while writing to the FENTRYR register is enabled and the FENTRYR register is 0000h

[Clearing Conditions]

- When the FENTRYR register is written in 8-bit units while the FSTATR.FRDY flag is 1
- When a value other than AAh is specified for the KEY bits while the FSTATR.FRDY flag is 1, and the FENTRYR register is written in 16-bit units

- When 0 is written to the FENTRYD bit while writing to the FENTRYR register is enabled
- When the FENTRYR register is written while writing to FENTRYR register is enabled and the value of the FENTRYR register is other than 0000h

KEY[7:0] Bits (Key Code)

These bits control permission and prohibition of writing to the FENTRYD and FENTRYC bits.

55.4.13 Flash Sequencer Set-Up Initialization Register (FSUINITR)

Address(es): FLASH.FSUINITR 007F E08Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	—	SUINIT
Value after 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	SUINIT	Set-Up Initialization	0: The FEADDR, FCPSR, FSADDR, FENTRYR, and FBCCNT flash sequencer set-up registers keep their current values. 1: The FEADDR, FCPSR, FSADDR, FENTRYR, and FBCCNT flash sequencer set-up registers are initialized.	R/W*1, *2
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	Key code	R/W*3

Note 1. Writing to these bits is possible only when the FSTATR.FRDY flag is 1. Writing to these bits while the FSTATR.FRDY flag = 0 is ignored.
Note 2. Writing to these bits is possible only when 2Dh is written to the KEY bits in 16-bit units.
Note 3. Written values are not retained by these bits. This bit is read as 0.

This register is used for initialization of the flash sequencer set-up.

SUINIT Bit (Set-Up Initialization)

This bit initializes the following flash sequencer set-up registers.

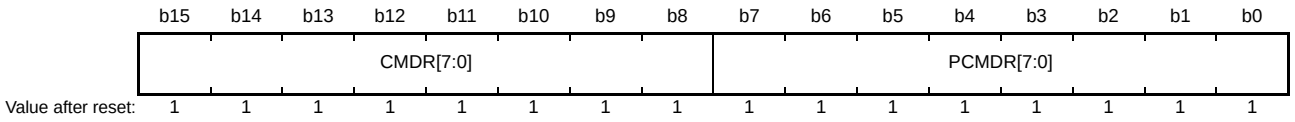
- FEADDR
- FCPSR
- FSADDR
- FENTRYR
- FBCCNT

KEY[7:0] Bits (Key Code)

These bits control permission and prohibition of writing to the SUINIT bit.

55.4.14 FACI Command Register (FCMDR)

Address(es): FLASH.FCMDR 007F E0A0h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	PCMDR[7:0]	Precommand Flag	The command immediately before the latest command is stored.	R
b15 to b8	CMDR[7:0]	Command Flag	The latest command is stored.	R

This register records the two most recent commands accepted by the FACI.

PCMDR[7:0] Flags (Precommand Flag)

These flags indicate the command received immediately before the last command received by the FACI.

CMDR[7:0] Flags (Command Flag)

These flags indicate the latest command received by the FACI.

Table 55.4 States of FCMDR after Receiving Commands

Command	CMDR	PCMDR
Programming	E8h	Previous command
Block erase	D0h	20h
Multi-block erase	D0h	21h
P/E suspend	B0h	Previous command
P/E resume	D0h	Previous command
Status clear	50h	Previous command
Forced stop	B3h	Previous command
Blank check	D0h	71h
Configuration setting	40h	Previous command

55.4.15 Data Flash Blank Check Control Register (FBCCNT)

Address(es): FLASH.FBCCNT 007F E0D0h

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

Bit	Symbol	Bit Name	Description	R/W
b0	BCDIR	Blank Check Direction	0: Blank checking is executed from lower addresses to higher addresses (incremental mode). 1: Blank checking is executed from higher addresses to lower addresses (decremental mode).	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register specifies the addressing mode in processing of a blank check command. The register is initialized when the FSUINITR.SUINIT bit is set to 1. It is also initialized by a reset.

BCDIR Bit (Blank Check Direction)

This bit specifies the addressing mode for blank checking.

55.4.16 Data Flash Blank Check Status Register (FBCSTAT)

Address(es): FLASH.FBCSTAT 007F E0D4h

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

Bit	Symbol	Bit Name	Description	R/W
b0	BCST	Blank Check Status Flag	0: The target area is in the non-programmed state (i.e. is blank; the area has been erased but has not yet been re-programmed). 1: The target area has been programmed with 0s or 1s.	R
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register stores the results of checking in response to a blank check command.

BCST Flag (Blank Check Status Flag)

This flag indicates the results of checking in response to a blank check command.

[Setting condition]

- When the blank checking command is issued and data of 0 or 1 is written to the target area

[Clearing condition]

- When the blank checking command is executed and the target area is blank

55.4.17 Data Flash Programming Start Address Register (FPSADDR)

Address(es): FLASH.FPSADDR 007F E0D8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PSADR[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
	PSADR[15:0]															
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b16 to b0	PSADR[16:0]	Programmed Area Start Address	The starting address of the programmed area to be found firstly	R
b31 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register indicates the starting address of the programmed area to be found firstly in processing of a blank check command.

PSADR[16:0] Bits (Programmed Area Start Address)

These bits indicate the starting address of the programmed area to be found firstly in processing of a blank check command. The offset value is stored from the first address of the data flash memory. The value of the bits is valid only while the FBCSTAT.BCST bit is 1.

55.4.18 Flash Sequencer Processing Switching Register (FCPSR)

Address(es): FLASH.FCPSR 007F E0E0h

	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

This register is for selecting the erasure suspension mode.
The register is initialized when the FSUINITR.SUINIT bit is set to 1. It is also initialized by a reset.

ESUSPMD Bit (Erasure Suspend Mode)

This bit is for selecting the erasure suspension mode when a P/E suspend command is issued while the flash sequencer is executing erasure processing (see section 55.8.3.9, P/E Suspend Command). This bit should be set before issuing a block erase command.

55.4.19 Flash Sequencer Processing Clock Frequency Notification Register (FPCKAR)

Address(es): FLASH.FPCKAR 007F E0E4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								PCKA[7:0]							
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

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	PCKA[7:0]	Flash Sequencer Processing Clock Frequency Notification	These bits are used to set the frequency of the FlashIF clock (FCLK) and notify the flash sequencer of the frequency used	R/W*1, *2
b15 to b8	KEY[7:0]	Key Code	Key code	R/W*3

Note 1. Writing to these bits is possible only when the FSTATR.FRDY flag is 1. Writing to these bits while the FSTATR.FRDY flag = 0 is ignored.
Note 2. Writing to these bits is possible only when 1Eh is written to the KEY[7:0] bits in 16-bit units.
Note 3. Written values are not retained by these bits. These bits are read as 0.

This register specifies the frequency of the FlashIF clock (FCLK) generated in the clock generator and notifies the flash sequencer of the frequency used. The flash sequencer determines the FACI command processing time based on the frequency notified by the FPCKAR register. The initial value is set to the maximum operating frequency of the FCLK.

PCKA[7:0] Bits (Flash Sequencer Processing Clock Frequency Notification)

These bits are used to specify the frequency of the FCLK generated in the clock generator and to notify the flash sequencer of the frequency used. Set the desired frequency in these bits before issuing an FACI command. Specifically, convert the frequency represented in MHz into a binary number and set it in these bits.

Example: When frequency is 35.9 MHz (PCKA[7:0] = 24h)
Round up the first decimal place of 35.9 MHz to a whole number (= 36) and convert it into a binary number.

If the value set in these bits is smaller than the frequency of the FCLK, the rewriting characteristics of the flash memory cannot be guaranteed. Conversely, if the value set in these bits is greater than the frequency of the FCLK, the rewriting characteristics of the flash memory can be guaranteed although the FACI command processing time such as time for rewriting will increase (the FACI command processing time becomes the shortest when the frequency of the FCLK is the same as the value set in the bits).

KEY[7:0] Bits (Key Code)

These bits control permission and prohibition of writing to the PCKA[7:0] bits.

55.4.20 Flash Access Window Monitor Register (FAWMON)

Address(es): FLASH.FAWMON 007F E0DCh

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	BTFLG	—	—	—	FAWE[11:0]											
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FSPR	—	—	—	FAWS[11:0]											
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b11 to b0	FAWS[11:0]	Flash Access Window Start Address	Flash access window start address	R
b14 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	FSPR	Access Window Protection Flag	0: With protection (P/E disabled) 1: Without protection (P/E enabled)	R
b27 to b16	FAWE[11:0]	Flash Access Window End Address	Flash access window end address	R
b30 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b31	BTFLG	Start-Up Area Select Flag*2	0: Startup area 0 is in the range from FFFF C000h to FFFF DFFFh, startup area 1 is in the range from FFFF E000h to FFFF FFFFh. 1: Startup area 1 is in the range from FFFF C000h to FFFF DFFFh, startup area 0 is in the range from FFFF E000h to FFFF FFFFh.	R

Note 1. For products in which the option-setting memory has not been specifically set, the value is FFFF FFFFh. The value will be set by the user.

Note 2. When the FSUACR.SAS[1:0] bits are changed to 1xb, the startup area is dependent on the setting of the FSUACR.SAS[1:0] bits regardless of the setting of the FAW.BTFLG bit.

This register indicates the values of the write protection flag and start-up area select flag for setting the flash access window start/end address, and the access window. When a reset or configuration setting command is executed, the FACI transfers data from the option-setting memory to this register and the setting of the option-setting memory is enabled.

FAWS[11:0] Bits (Flash Access Window Start Address)

These bits are used to verify the access window start address setting value.

FSPR Flag (Access Window Protection Flag)

This flag indicates whether or not protection for a configuration setting command for the access window setting, for a configuration clear command, or for writing to the FSUACR register is available.

FAWE[11:0] Bits (Flash Access Window End Address)

These bits are used to verify the access window end address setting value. The value of these bits indicates the first address of the next P/E-enabled block as configured in the access window.

BTFLG Flag (Start-Up Area Select Flag)

This flag indicates whether the start-up area is switched by using start-up program protection.

55.4.21 Start-Up Area Control Register (FSUACR)

Address(es): FLASH.FSUACR 007F E0E8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	KEY[7:0]								—	—	—	—	—	—	SAS[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	SAS[1:0]	Start-Up Area Select	b1 b0 0 x: A start-up area is selected based on the setting of the FAW.BTFLG bit. 1 0: The area from FFFF E000h to FFFF FFFFh is selected as the address range for startup area 0, and the area from FFFF C000h to FFFF DFFFh is selected as the address range for startup area 1, regardless of the setting of the FAW.BTFLG bit. 1 1: The area from FFFF E000h to FFFF FFFFh is selected as the address range for startup area 1, and the area from FFFF C000h to FFFF DFFFh is selected as the address range for startup area 0, regardless of the setting of the FAW.BTFLG bit.	R/W*1, *2
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	Key codes	R/W*3

x: Don’t care

Note 1. Writable only when the FAW.FSPR bit is 1. When the FAW.FSPR bit is 0, writing to these bits are ignored.

Note 2. Writing to these bits is possible only when 66h is written to the KEY bit in 16-bit units.

Note 3. Written values are not retained by these bits. These bits are read as 0.

This register is used to switch startup areas 0 and 1 by startup program protection.

Do not use this register in dual mode (the MDE.BANKMD[2:0] bits are 000b). In dual mode, starting up proceeds from startup area 0.

SAS[1:0]Bits (Start-Up Area Select)

These bits are used to switch startup areas 0 and 1.

KEY[7:0] Bits (Key Code)

These bits enables or disable overwriting to the SAS[1:0] bits.

55.4.22 Data Flash Memory Access Frequency Setting Register (EEPFCLK)

Address(es): FLASH.EEPFCLK 007F C040h

	b7	b6	b5	b4	b3	b2	b1	b0
	EEPFCLK[7:0]							
Value after reset:	0	0	1	1	1	1	0	0

This register optimizes the speed of reading the data flash memory.
Set the frequency of the peripheral module clock (FCLK) of internal peripheral bus 6 which is the clock for access to the data flash memory, in MHz units. For example, 35.9 MHz should be rounded up and set the frequency to 36. Number of cycles required for access to the data flash memory are inserted according to the frequency.
When changing the frequency of the FCLK, follow the procedure below to modify the value of the data flash memory access frequency setting register (EEPFCLK) in either of the following ways according to whether operation is at a lower frequency before or after the change.

- When changing the speed from low to high: Modify EEPFCLK. After confirming the change by reading EEPFCLK, change the frequency.
- When changing the speed from high to low: Change the frequency. After the frequency is changed, modify EEPCFLK.

55.4.23 Unique ID Register n (UIDRn) (n = 0 to 3)

Address(es): FLASHCONST.UIDR0 FE7F 7D90h, FLASHCONST.UIDR1 FE7F 7D94h, FLASHCONST.UIDR2 FE7F 7D98h, FLASHCONST.UIDR3 FE7F 7D9Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	Unique value for each chip															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	Unique value for each chip															

Note: These registers are only readable when the SYSCR0.ROME bit is 1 (the on-chip ROM is enabled).

The UIDRn is a read-only register that stores a 16-byte ID code (unique ID) for identifying the individual MCU. The UIDRn register should be read in 32-bit units.

55.5 Operating Modes Associated with Flash Memory

Figure 55.7 is a diagram of the mode transitions associated with the flash memory. For the procedures for setting the modes, refer to section 3, Operating Modes.

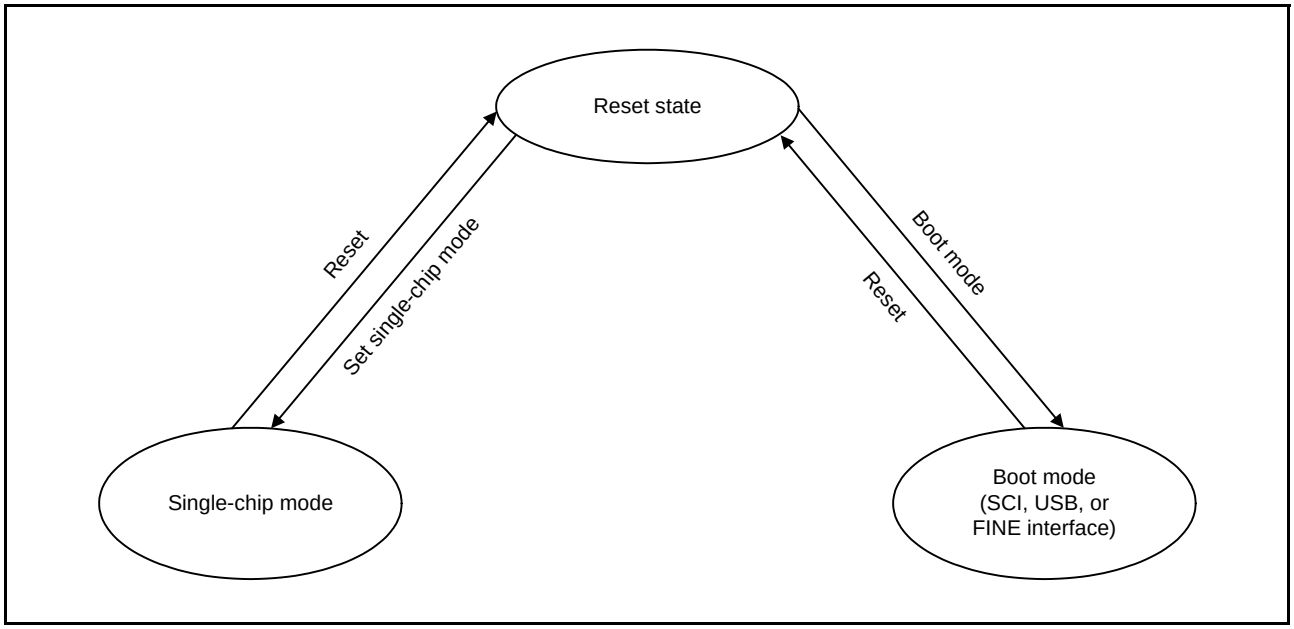


Figure 55.7 Mode Transitions Associated with Flash Memory

The flash memory area where programming and erasure are permitted and the boot program after a reset are different according to each mode. The differences between modes are listed in Table 55.5.

Table 55.5 Differences between Modes

Item	Single-chip Mode or On-chip ROM Enabled Extended Mode	Boot Mode (SCI, USB, or FINE Interface)
Programmable and erasable area	Code flash memory Data flash memory Option-setting memory (programming only)	Code flash memory Data flash memory Option-setting memory
Boot program at a reset	Program of the code flash memory	Boot program

55.6 Operating Modes of the Flash Sequencer

The flash sequencer has three operating modes as shown in Figure 55.8. Transitions between modes are initiated by changing the value of the FENTRYR register.

When the value of the FENTRYR register is 0000h, the flash sequencer is in read mode. In this mode, it does not receive FACI commands. The code flash memory and data flash memory are readable.

When the value of the FENTRYR register is 0001h, the flash sequencer is in code flash memory P/E mode where the code flash memory can be programmed or erased by FACI commands. In this mode, reading from the code flash memory is disabled under the conditions where BGO cannot be used. Under the conditions where BGO can be used, reading from the code flash memory is enabled.

When the value of the FENTRYR register is 0080h, the flash sequencer is in data flash memory P/E mode where the data flash memory can be programmed or erased by FACI commands. In this mode, the data flash memory is not readable. However, the code flash memory is readable.

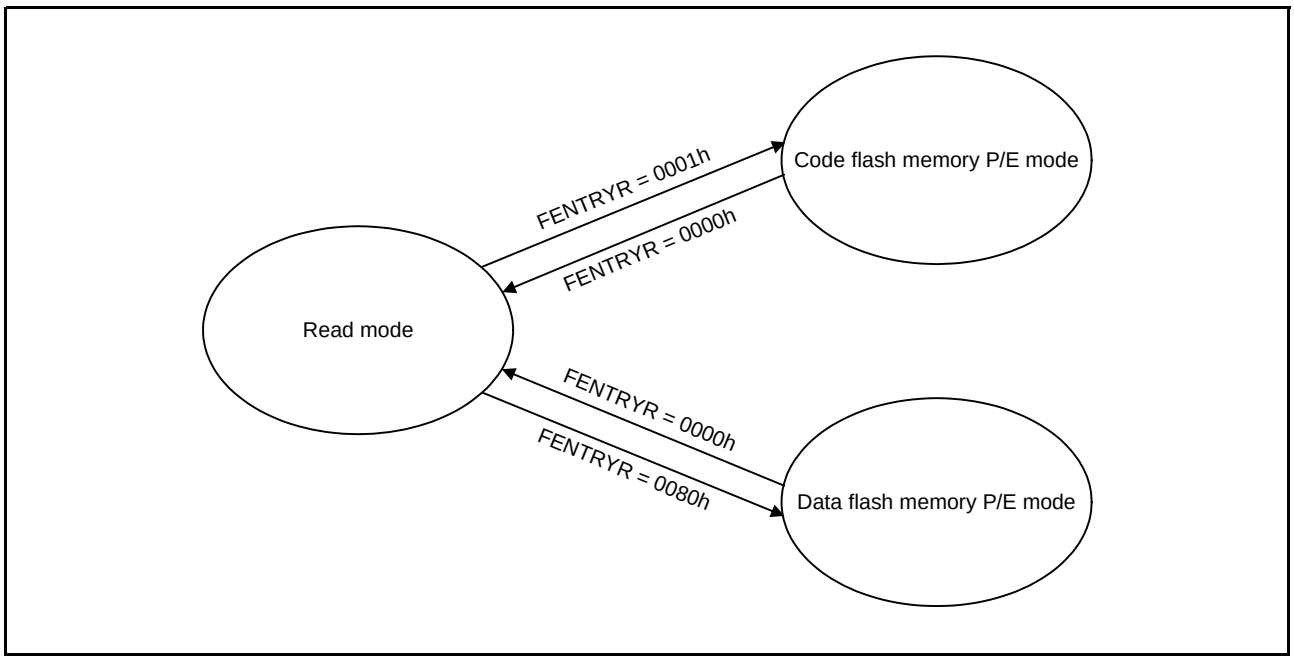


Figure 55.8 Modes of the Flash Sequencer

55.7 Overview of Functions

55.7.1 ROM Cache

This MCU has three types of cache as listed below.

- Instruction fetching cache (IF cache): 8 Kbytes
- Data cache (OA cache): 16 bytes
- Data cache (DM cache): 16 bytes

The IF cache and OA cache are for the CPU, and the DM cache is for bus masters other than the CPU, such as the DMAC and DTC.

After release from a reset, the caches are automatically invalidated following programming and erasure of the flash memory.

The area subject to caching is the 2-Mbyte address range from FFE0 0000h through FFFF FFFFh, in which up to two areas can be set as not within the scope of caching (non-cacheable areas).

55.7.1.1 Setting Non-Cacheable Area

Parts of the flash memory of this MCU that are within the cacheable areas can be set as not in the scope of caching by setting the NCRGn and NCRCn registers (n = 0, 1). Figure 55.9 shows the settings of each register and the non-cacheable areas.

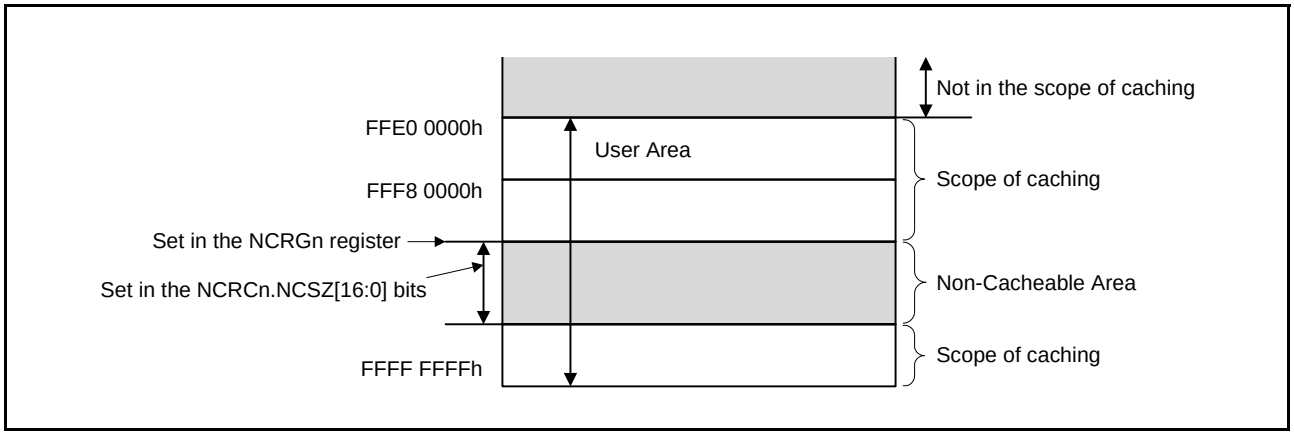


Figure 55.9 Specifying a Non-Cacheable Area (n = 0, 1)

The restrictions shown below in Table 55.6 apply to the settings of the NCRCn.NCSZ[16:0] bits and of the NCRGn register.

Table 55.6 Restrictions Applying to the Settings of the NCRCn.NCSZ[16:0] Bits and of the NCRGn Register

Settings of the NCRCn.NCSZ[16:0] Bits	Restrictions Applying to the Settings of the NCRGn Register
0 0000 0000 0000 0000: 16 bytes	No restriction
0 0000 0000 0000 0001: 32 bytes	Bit 4 in the NCRGn register is ignored.
0 0000 0000 0000 0011: 64 bytes	Bits 5 and 4 in the NCRGn register are ignored.
0 0000 0000 0000 0111: 128 bytes	Bits 6 to 4 in the NCRGn register are ignored.
0 0000 0000 0000 1111: 256 bytes	Bits 7 to 4 in the NCRGn register are ignored.
0 0000 0000 0001 1111: 512 bytes	Bits 8 to 4 in the NCRGn register are ignored.
0 0000 0000 0011 1111: 1 Kbyte	Bits 9 to 4 in the NCRGn register are ignored.
0 0000 0000 0111 1111: 2 Kbytes	Bits 10 to 4 in the NCRGn register are ignored.
0 0000 0000 1111 1111: 4 Kbytes	Bits 11 to 4 in the NCRGn register are ignored.
0 0000 0001 1111 1111: 8 Kbytes	Bits 12 to 4 in the NCRGn register are ignored.
0 0000 0011 1111 1111: 16 Kbytes	Bits 13 to 4 in the NCRGn register are ignored.
0 0000 0111 1111 1111: 32 Kbytes	Bits 14 to 4 in the NCRGn register are ignored.
0 0000 1111 1111 1111: 64 Kbytes	Bits 15 to 4 in the NCRGn register are ignored.
0 0001 1111 1111 1111: 128 Kbytes	Bits 16 to 4 in the NCRGn register are ignored.
0 0011 1111 1111 1111: 256 Kbytes	Bits 17 to 4 in the NCRGn register are ignored.
0 0111 1111 1111 1111: 512 Kbytes	Bits 18 to 4 in the NCRGn register are ignored.
0 1111 1111 1111 1111: 1 Mbyte	Bits 19 to 4 in the NCRGn register are ignored.

55.7.2 Methods of Programming/Erasure

By using a dedicated flash-memory programmer to program the flash memory of this MCU, the device can be rewritten regardless of whether this is before or after it is mounted on the target system.
Furthermore, security functions to prohibit rewriting or reading of the user program written to the flash memory are incorporated, and this can prevent falsification and illicit reading of the programs by third parties. TM target areas of the code flash memory can also be protected against reading by using the TM function.
Programming by the user program (self-programming) is available to suit applications where the application on the target system may require updating after manufacturing or shipment. Protection features for the safe rewriting of the flash memory are also incorporated. Furthermore, interrupt processing during self-programming is supported, so programming can proceed at the same time as processing for external communications, etc., and this is the case in various situations.
Table 55.7 lists the overview of the methods of programming and the corresponding operating modes.

Table 55.7 Methods of Programming

Method of Programming	Functional Overview	Operating Mode
Programming by a flash-memory programmer	A serial programmer is capable of on-board programming of the flash memory after the device is mounted on the target system. The TM function can be in enabled or disabled at this time.	Boot mode
	A parallel programmer and dedicated programming adapter board allow off-board programming of the flash memory, i.e. programming of the device before it is mounted on the target system.	
Self-programming	The execution of the user program that is written to code flash memory in advance by serial programming executing is also capable of programming the flash memory. The TM function can be in the enabled state at this time. The background operation capability makes it possible to fetch instructions or otherwise read data from code flash memory while the data flash memory is being programmed. Therefore, the data flash memory can be rewritten by executing a program for writing contained by the code flash memory. Furthermore, background operation can also be used for reading and writing to code flash memory alone, but only when the address range of code flash memory that is the target for programming and the address range of code flash memory that is the target for reading satisfy particular conditions (see Table 55.25). When this is the case, at the time of self-programming, a program for programming that is in the code flash memory can be executed to rewrite the code flash memory. In cases where background operation is not possible, instructions in the code flash memory cannot be fetched and data cannot be accessed while code flash memory is being re-written by self-programming. In such cases, a program for programming must be transferred to the internal RAM or external memory in advance and executed.	Single-chip mode On-chip ROM enabled extended mode

Table 55.8 lists the functions of the flash memory. Serial programmer commands realize each function of serial programming, while reading of the flash memory by an FACI command or the user program realizes each function of self-programming.
For security function settings, see section 7.2.1, Serial Programmer Command Control Register (SPCC), in section 7, Option-Setting Memory (OFSM).

Table 55.8 List of Basic Functions

Function	Functional Overview	Support Status	
		Serial Programming	Self-programming
Blank check	This is used to check a specified block to ensure that writing to it has not already proceeded. Results of reading from data flash memory to which nothing has been written after erasure are not guaranteed, so use blank checking to confirm that programming to memory has not proceeded after erasure.	Supported	Supported
Block erasure	Erases the memory contents in the specified block.	Supported	Supported
Multi-block erasure	Erases the contents of the specified block of the data flash memory.	Not supported	Supported
Programming	Write to the specified address.	Supported	Supported
Verify/Checksum	Compares the data read from flash memory and the data transferred from a serial programmer.	Supported	Not supported (read by user program is possible)
Read	Read data programmed to flash memory.	Supported	Supported
Setting of control or ID codes	Sets the OSIS register.	Supported	Supported
Prohibition of serial programmer connection	Sets the permission/prohibition of serial programmer connection.	Supported	Supported with conditions (Only switching the configuration from enabled to disabled is possible)
Prohibition of on-chip debugger connection	Sets the permission/prohibition of on-chip debugger connection.	Supported	Supported with conditions (Only switching the configuration from enabled to disabled is possible)
Area protection and start-up program protection functions	Configures the area protection and start-up program protection functions.	Supported	Supported
Option function selection	Selects the option function, and modifies the initial settings of this MCU.	Supported	Supported
Configuration clearing	Erases the option-setting memory and the TM target areas.	Supported	Not supported
Dual bank function	Switches different modes (linear or dual).	Supported	Supported
Setting the TM function	Sets the TM function.	Supported	Supported with conditions (Only switching the configuration from disabled to enabled is possible)

55.7.3 Security Functions

The flash memory supports various security functions.
The security function includes serial programmer ID code protection, prohibition of serial programmer connection, on-chip debugger ID code protection, prohibition of on-chip debugger connection, and the ROM code protection.
In serial programming, serial programmer ID code protection and prohibition of serial programmer connection can be used. On-chip debugger ID code protection and prohibition of on-chip debugger connection can be used during connection of the on-chip debugger. In off-board programming, ROM code protection can be used.
Table 55.9 lists the security functions supported by the flash memory.

Table 55.9 Lists of Security Functions

Function	Description
Serial programmer ID code protection	Connection of a serial programmer can be controlled by judging the control code or ID code.
Prohibition of serial programmer connection	The connection of a serial programmer for serial programming is prohibited. Since execution of the configuration clearing command is also prohibited when the connection of a serial programmer is prohibited, changing a security setting from “disabled” to “enabled” is not possible.
On-chip debugger ID code protection	Connection of an on-chip debugger can be controlled by judging the ID code.
Prohibition of on-chip debugger connection	The connection of an on-chip debugger is prohibited regardless of the ID code setting.
ROM code protection	This function is to prohibit reading, programming, and erasure of the flash memory when a parallel programmer is in use.

55.8 FACI Commands

55.8.1 List of FACI Commands

Table 55.10 List of FACI Commands

FACI Command	Description
Programming	This is used to program the code or data flash memory. Units of programming are 128 bytes for the code flash memory and 4 bytes for the data flash memory.
Block erase	This is used to erase the code or data flash memory. The unit of erasure is one block. (code flash memory: 8 K or 32 Kbytes, data flash memory: 64 bytes)
Multi-block erase	This is used to erase the data flash memory. The unit of erasure is 64, 128, 256 bytes
P/E suspend	This suspends programming or erasure processing.
P/E resume	This resumes suspended programming or erasure processing.
Status clear	This initializes the ILGLERR, ERSERR, PRGERR, ILGCOMERR, FESETERR, SECERR, and OTERR flags in the FSTATR register and the CMDLK, CFAE, and DFAE flags in the FASTAT releases the flash sequencer from the command-locked state.
Forced stop	This forcibly stops processing of FACI commands and initializes the FASTAT and FSTATR registers.
Blank check	This is used to blank-check the data flash memory. Range of checking: 4 bytes to 8 Kbytes (specified in 4-byte units).
Configuration setting	This is used to set the option-setting memory (configuration setting area). Units of setting: 16 bytes.

The FACI commands are issued by writing to the FACI command-issuing area (see Table 55.2). When write access as shown in Table 55.11 proceeds in the specified state, the flash sequencer executes the processing corresponding to the given command (see section 55.8.2, Relationship between the Flash Sequencer State and FACI Commands).

Table 55.11 FACI Command Formats

FACI Commands	Number of Write Access	Data to be Written to the FACI Command-Issuing Area			
		1st Access	2nd Access	3rd to (N+2)th Access	(N+3)th Access
Programming (code flash memory) 128-byte programming, N = 64	67	E8h	40h (= N)	WD_1 to WD_{64}	D0h
Programming (data flash memory) 4-byte programming, N = 2	5	E8h	02h (= N)	WD_1 to WD_2	D0h
Block erase (code flash memory)	2	20h	D0h	—	—
Block erase (data flash memory 64-byte)	2	20h	D0h	—	—
Multi-block erase (data flash memory 64-byte/128-byte/256-byte)	2	21h	D0h	—	—
P/E suspend	1	B0h	—	—	—
P/E resume	1	D0h	—	—	—
Status clear	1	50h	—	—	—
Forced stop	1	B3h	—	—	—
Blank check	2	71h	D0h	—	—
Configuration setting N = 8	11	40h	08h (= N)	WD_1 to WD_8	D0h

Note: WD_N (N = 1, 2,…): Nth 16-bit data to be programmed.

The flash sequencer clears the FSTATR.FRDY flag to 0 at the start of processing of a command other than the status clear command and sets this bit to 1 upon completion of command processing.
When the setting of the FRDYIE.FRDYIE bit is 1 and when the FSTATR.FRDY flag is set to 1, a flash ready (FRDY) interrupt is generated.

55.8.2 Relationship between the Flash Sequencer State and FACI Commands

Each FACI command can be accepted in a specific mode or state of the flash sequencer. FACI commands should be issued after the transition of the flash sequencer to the code flash memory P/E mode or data flash memory P/E mode and checking of the state of the flash sequencer. Use the FSTATR and FASTAT registers to check the state of the flash sequencer. The value of the FASTAT.CMDLK flag is the logical OR of values of the ILGLERR, ILGCOMERR, FESETERR, SECERR, OTERR, ERSERR, PRGERR, and FLWEERR flags in the FSTATR register. Therefore, the occurrence of errors can be checked by reading the value of the FASTAT.CMDLK flag.

Table 55.12 lists the available commands in each operating mode.

Table 55.12 Operating Mode and Available Commands

Operating Mode	FENTRYR Register Value	Available Commands
Read mode	0000h	None
Code flash memory P/E mode	0001h	Programming Block erase P/E suspend P/E resume Status clear Forced stop Configuration setting
Data flash memory P/E mode	0080h	Programming Block erase Multi-block erase P/E suspend P/E resume Status clear Forced stop Blank check

Table 55.13 shows the state of the flash sequencer and acceptable FACI commands. An appropriate mode is assumed to be set before the commands are executed.

Table 55.13 Acceptable FACI Commands and the State of the Flash Sequencer

	Processing of Programming or Erasure	Processing of Configuration Setting	Processing to Suspend Programming or Erasure	Processing of Blank Checking	Programming Suspended	Erasure Suspended	Programming while Erasure is Suspended	Command-Locked State (FRDY = 1)	Command-Locked State (FRDY = 0)	Processing of Forced Stop Command	Other State
FRDY flag	0	0	0	0	1	1	0	1	0	0	1
SUSRDY flag	1	0	0	0	0	0	0	0	0	0	0
ERSSPD flag	0	0	0/1	0/1	0	1	1	0/1	0/1	0	0
PRGSPD flag	0	0	0/1	0/1	1	0	0	0/1	0/1	0	0
CMDLK flag	0	0	0	0	0	0	0	1	1	0	0
Programming	×	×*4	×	×	×	✓*3	×	×	×	×	✓
Block erase	×	×*4	×	×	×	×	×	×	×	×	✓
Multi-block erase	×	×*4	×	×	×	×	×	×	×	×	✓
P/E suspend	✓	×*4	×	×	×	×	×	—	×	×	—
P/E resume	×	×*4	×	×	✓	✓	×	×	×	×	×
Status clear	×	×*4	×	×	✓	✓	×	✓	×	×	✓
Forced stop	✓	✓*4	✓	✓	✓	✓	✓	✓	✓	✓	✓
Blank check	×	×	×	×	✓*1	✓*1	×	×	×	×	✓*1
Configuration setting	×	×*4	×	×	×	×	×	×	×	×	✓*2

✓: Acceptable
×: Not acceptable (the sequencer in the command-locked state)
—: Ignored
Note 1. Acceptable only in data flash memory P/E mode
Note 2. Acceptable only in code flash memory P/E mode
Note 3. Programming is acceptable only for blocks other than blocks where erasure has been suspended.
Note 4. Do not issue a FACI command when configuration setting is being processed and the FSTATR.DBFULL bit is 1.

55.8.3 Usage of FACI Commands

This section gives an overview of the usage of FACI commands.

55.8.3.1 Transition to Code Flash Memory P/E Mode

To use the FACI commands for the code flash memory, a transition to code flash memory P/E mode is required. To cause shift to code flash memory P/E mode, set the FENTRYR.FENTRYC bit to 1.

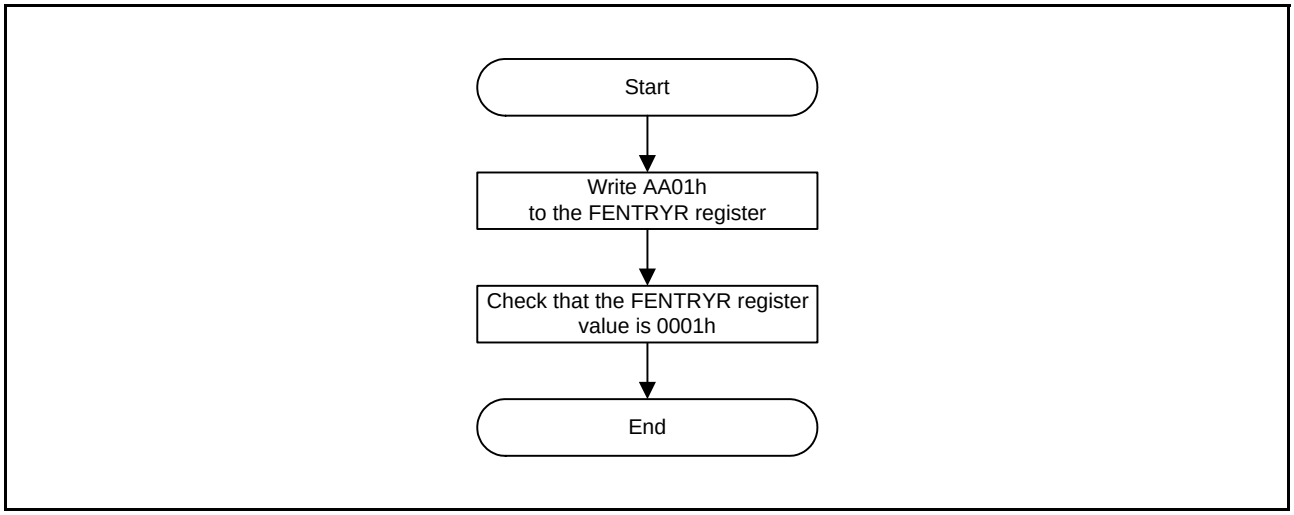


Figure 55.10 Procedure for Transition to Code Flash Memory P/E Mode

55.8.3.2 Transition to Data Flash Memory P/E Mode

To use the FACI commands for the data flash memory, a transition to data flash memory P/E mode is required. To shift to data flash memory P/E mode, set the FENTRYR.FENTRYRD bit to 1.

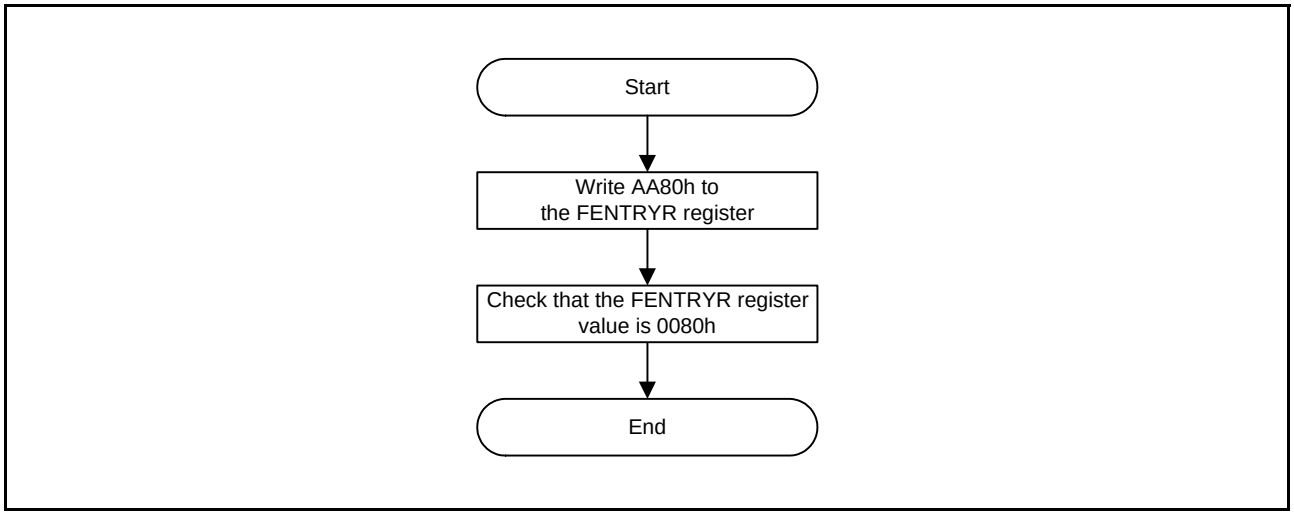
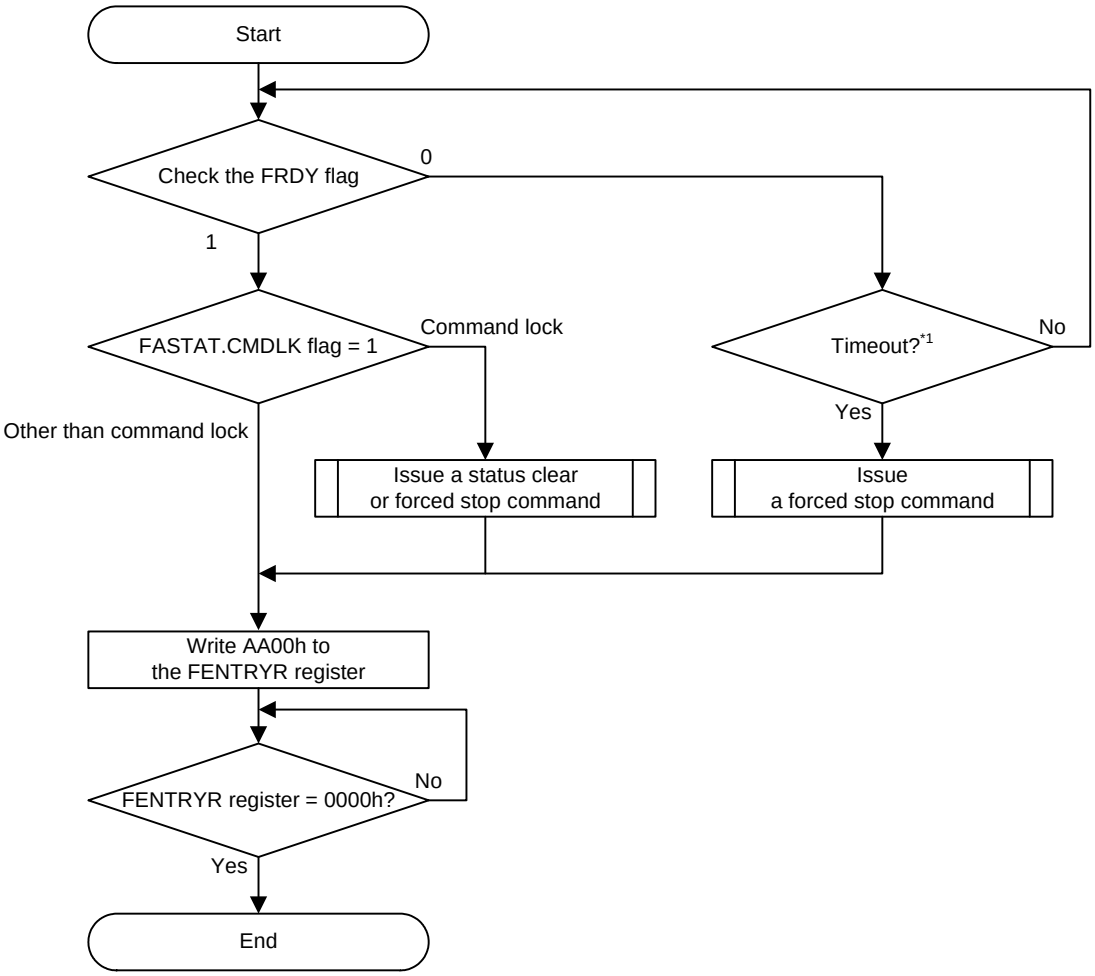


Figure 55.11 Procedure for Transition to Data Flash Memory P/E Mode

55.8.3.3 Transition to Read Mode

To read the flash memory without using the BGO function, a transition to read mode is required. To shift to read mode, set the FENTRYR register to 0000h. The transition to read mode should be made after processing by the flash sequencer is completed and while operation is in other than in the command-locked state. In addition, operation is started in read mode after release from the reset state.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time for processing of the FACI command that is in progress (refer to the Electrical Characteristics chapter).

Figure 55.12 Procedure for Transition to Read Mode

55.8.3.4 Overview Flow when FACI Command is Used

Figure 55.13 shows an overview flow when the FACI command is used.
If BGO is enabled, the jump to the internal RAM or external area (other than code flash memory) is not required because an FACI command can be issued for the code or data flash memory by using the rewriting program in the code flash memory.
When the FCLK is changed, changing the FPCKAR register shortens time for processing the FACI command. For details, refer to section 55.4.19, Flash Sequencer Processing Clock Frequency Notification Register (FPCKAR).

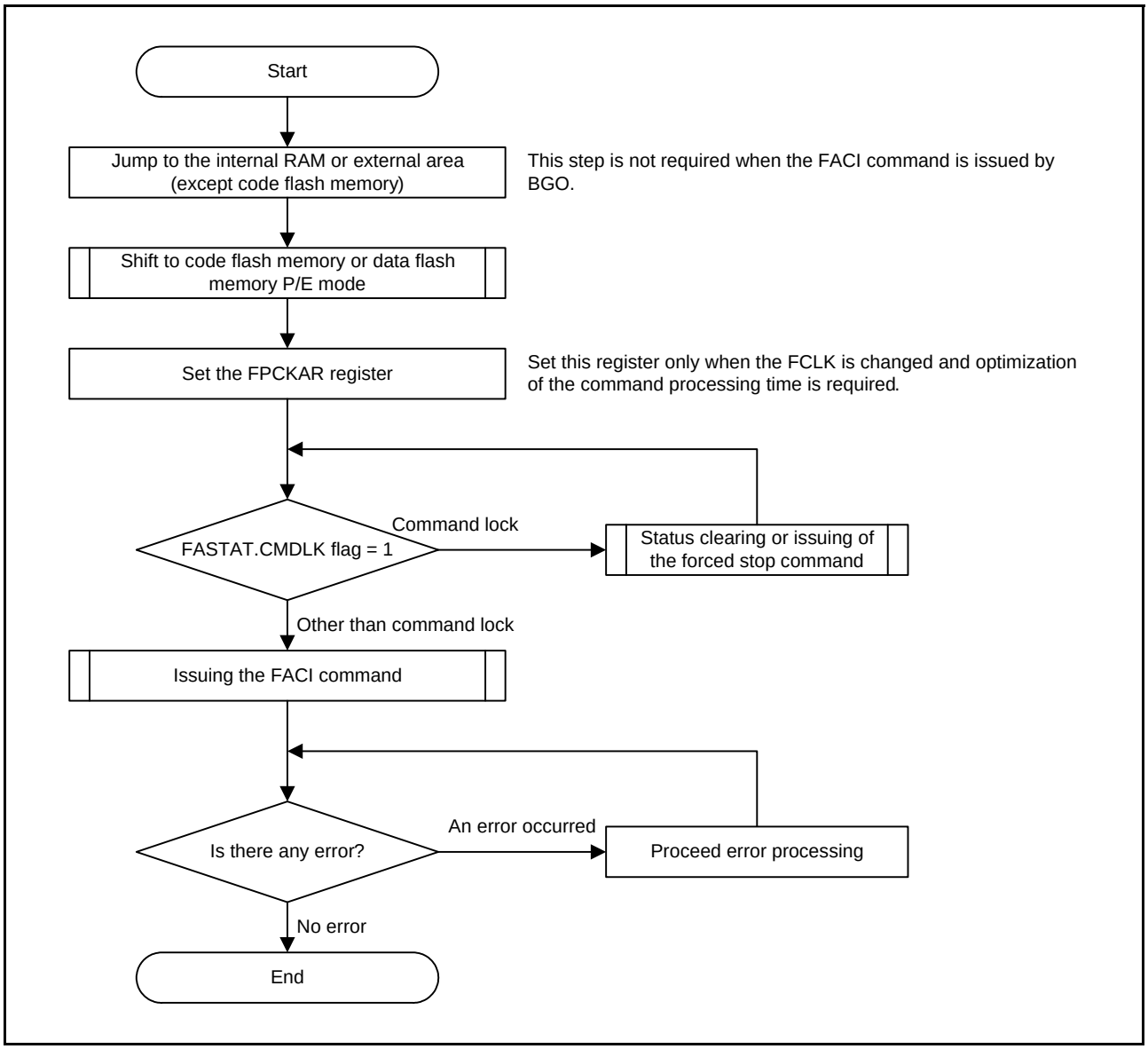
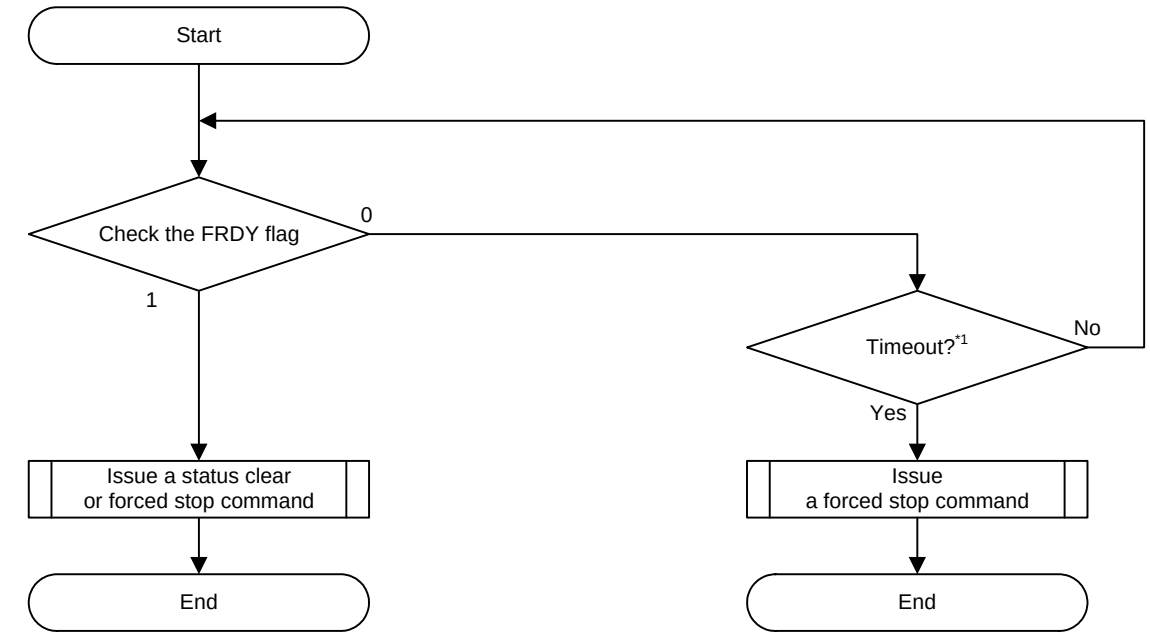


Figure 55.13 Overview Flow when FACI Command is Used

55.8.3.5 Recovery from the Command-Locked State

When the flash sequencer enters the command-locked state, FACI commands cannot be accepted. To release the sequencer from the command-locked state, use the status clear command, or forced stop command.
When the command-locked state is detected by checking for an error before issuing the P/E suspend command, the FSTATR.FRDY flag may hold 0 as the command processing has not been completed. When the processing is not completed even after time equal to the maximum programming or erasure time specified in the section 56, Electrical Characteristics times 1.1 has elapsed, this is considered a time-out and the flash sequencer should be stopped by the forced stop command.

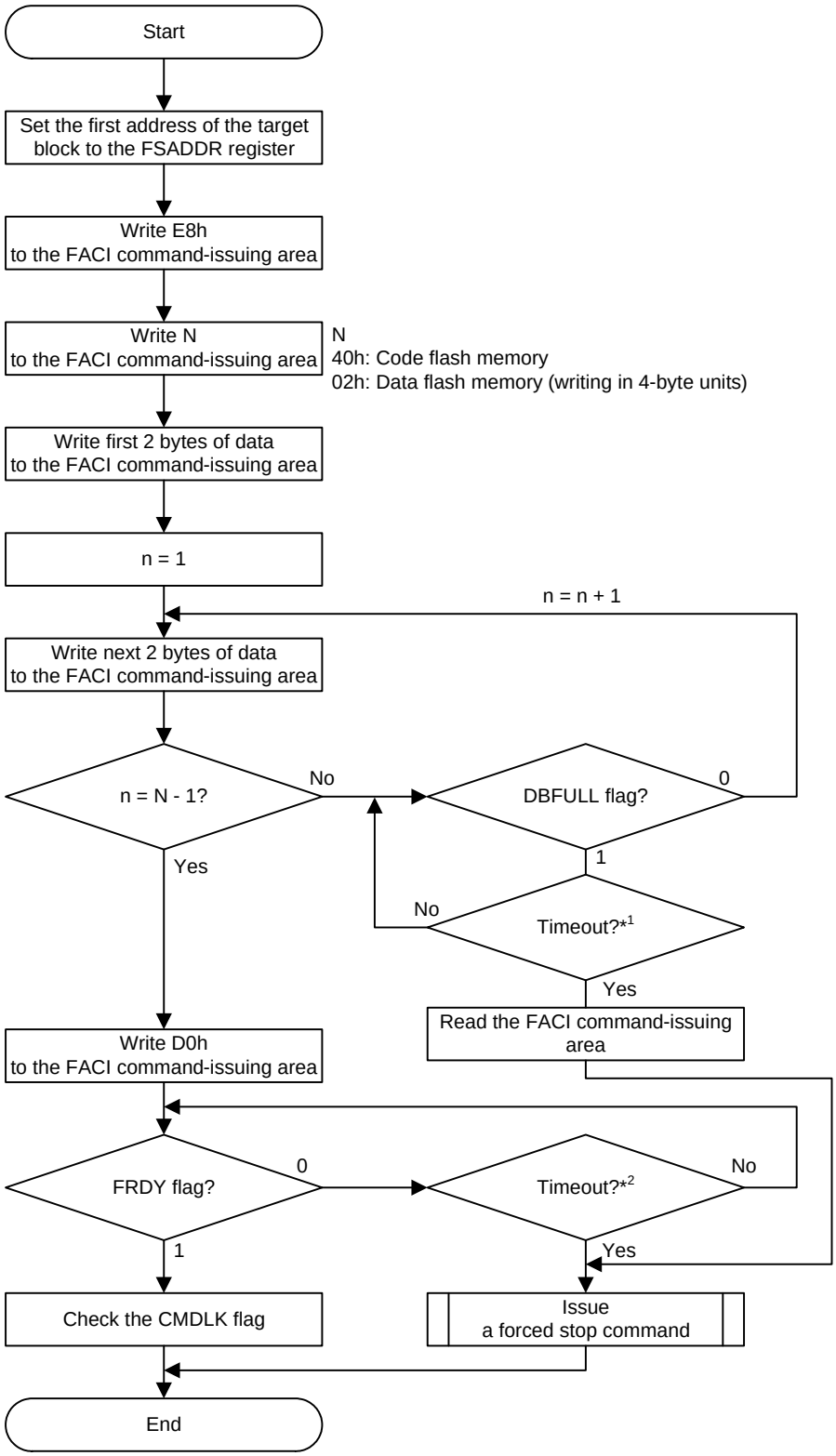


Note 1. Judgment of the timeout is based on 1.1 times the maximum time for processing of the FACI command that is in progress (refer to the Electrical Characteristics chapter).

Figure 55.14 Recovery from the Command-Locked State

55.8.3.6 Programming Command

A programming command is used for programming the code or data flash memory.
Before issuing a programming command, set the first address of the target block in the FSADDR register.
Writing D0h to the FACI command-issuing area at the final access of the FACI command-issuing starts the programming command processing. Completion of command processing can be checked by reading the FSTATR.FRDY flag. If the target area of programming command processing contains the area not for writing, write FFFFh to the corresponding area.
Issuing a programming command consecutively while the FACI internal data buffer is full leads to a wait on the peripheral bus 6 and this may affect on the bus accesses of the other peripheral IP modules. To avoid the generation of such a wait, issue an FACI command while the FSTATR.DBFULL flag is 0.
In addition, the data buffer never becomes full during programming of the data flash memory.

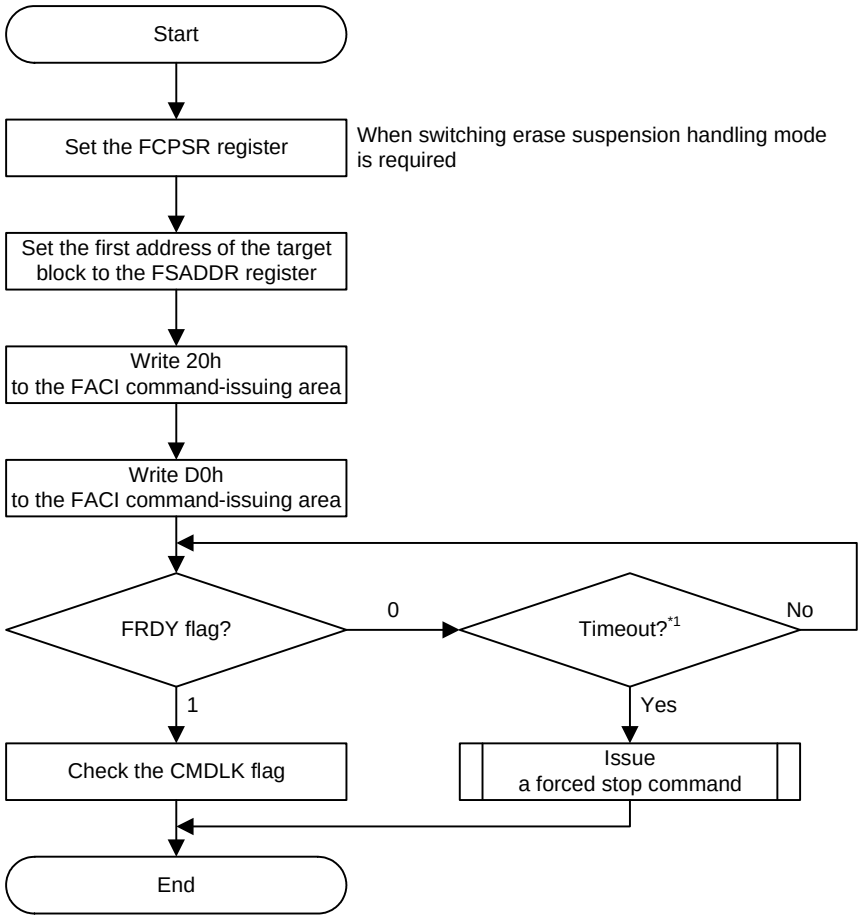


Note 1. Judgment of the timeout is based on 1.4 µs elapsing (when FCLK ≥ 20 MHz)
Note 2. Judgment of the timeout is based on 1.1 times 128-byte programming time (max) for code flash memory. For data flash memory, judgment is based on 1.1 times 4-byte programming time (max). For the 128-byte programming time and 4-byte programming time, refer to the Electrical Characteristics chapter.

Figure 55.15 Usage of the Programming Command

55.8.3.7 Block Erase Command

A block erase command is used to erase the code or data flash memory in single-block units.
Before issuing a block erase command, set the first address of the target block in the FSADDR register. Writing 20h and D0h to the FACI command-issuing area starts processing of a block erase command. Completion of command processing can be confirmed by reading the FSTATR.FRDY flag.
The FCPSR register must be set before issuing the block erase command. The setting of the FCPSR register must be changed to switch the suspending method (suspension priority mode/erasure priority mode) by the P/E suspend command.



Note 1. Judgment of the timeout is based on 1.1 times 32-Kbyte erasure time (max) for code flash memory.
For data flash memory, judgment is based on 1.1 times 64-byte erasure time (max).
For the 32-Kbyte erasure time and 64-byte erasure time, refer to the Electrical Characteristics chapter.

Figure 55.16 Usage of the Block Erase Command

55.8.3.8 Multi-Block Erase Command

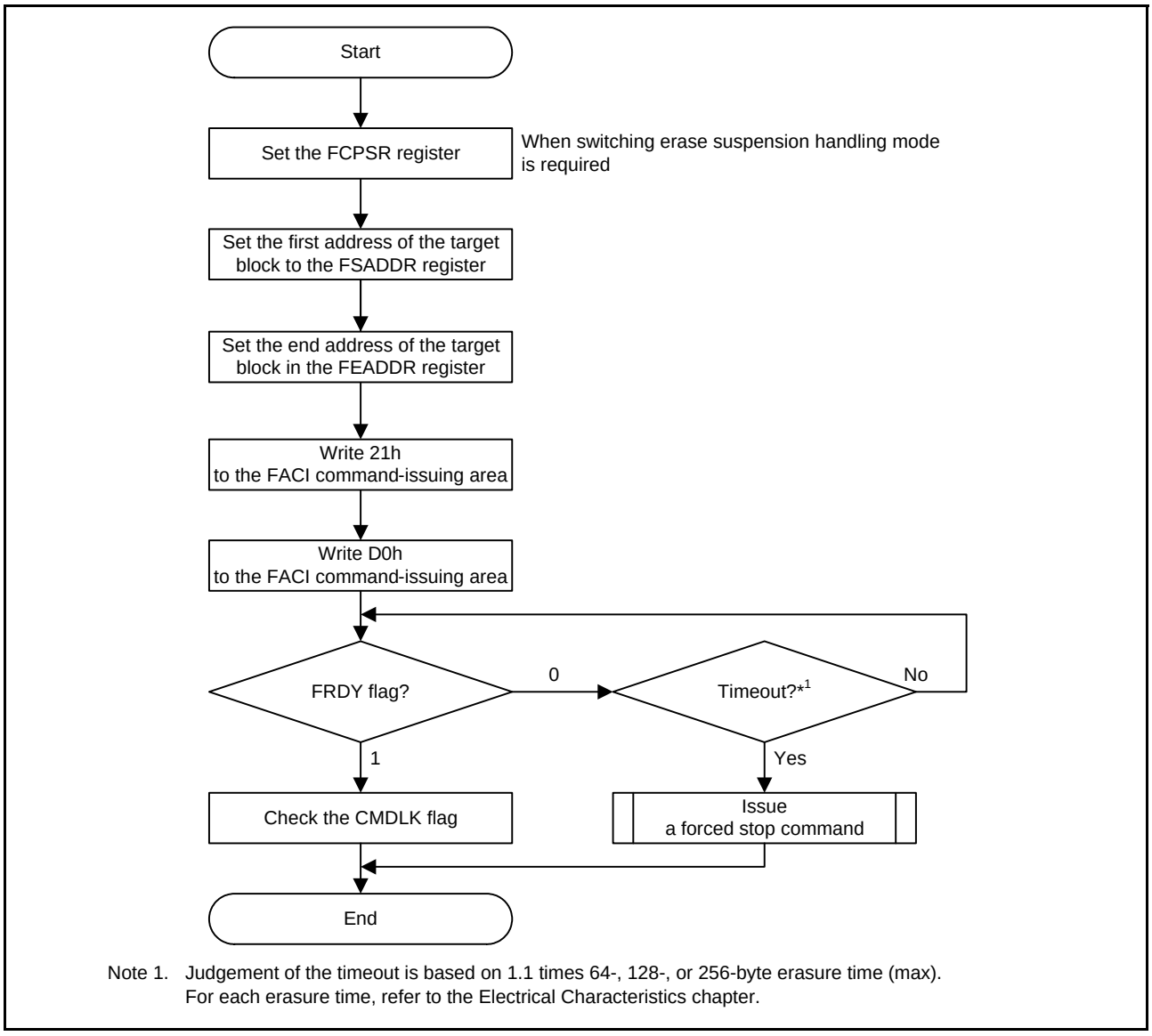
The multi-block erase command can also be issued to erase data in the data flash memory. The unit for erasure is 64, 128, or 256 bytes.

Set the first address of the target area for erasure to the FSADDR register and the end address to the FEADDR register before issuing a multi-block erase command. When 21h and D0h are written to the FACI command issuing area, the process of the multi-block erase command proceeds. The end of command processing can be confirmed by the FSTATR.FRDY flag.

The FCPSR register must be set before issuing the multi-block erase command. The setting of the FCPSR register must be changed when P/E suspension command is used to switch the handling of suspension of erasure (between suspension priority and erasure priority).

Table 55.14 Size Setting for Erasure

Erasure Size	FSADDR	FEADDR
64 bytes	FSA0 to FSA5 = 0 (64-byte boundary)	FSADDR + 3Ch
128 bytes	FSA0 to FSA6 = 0 (128-byte boundary)	FSADDR + 7Ch
256 bytes	FSA0 to FSA7 = 0 (256-byte boundary)	FSADDR + FCh



Note 1. Judgement of the timeout is based on 1.1 times 64-, 128-, or 256-byte erasure time (max). For each erasure time, refer to the Electrical Characteristics chapter.

Figure 55.17 Usage of the Multi-Block Erase Command

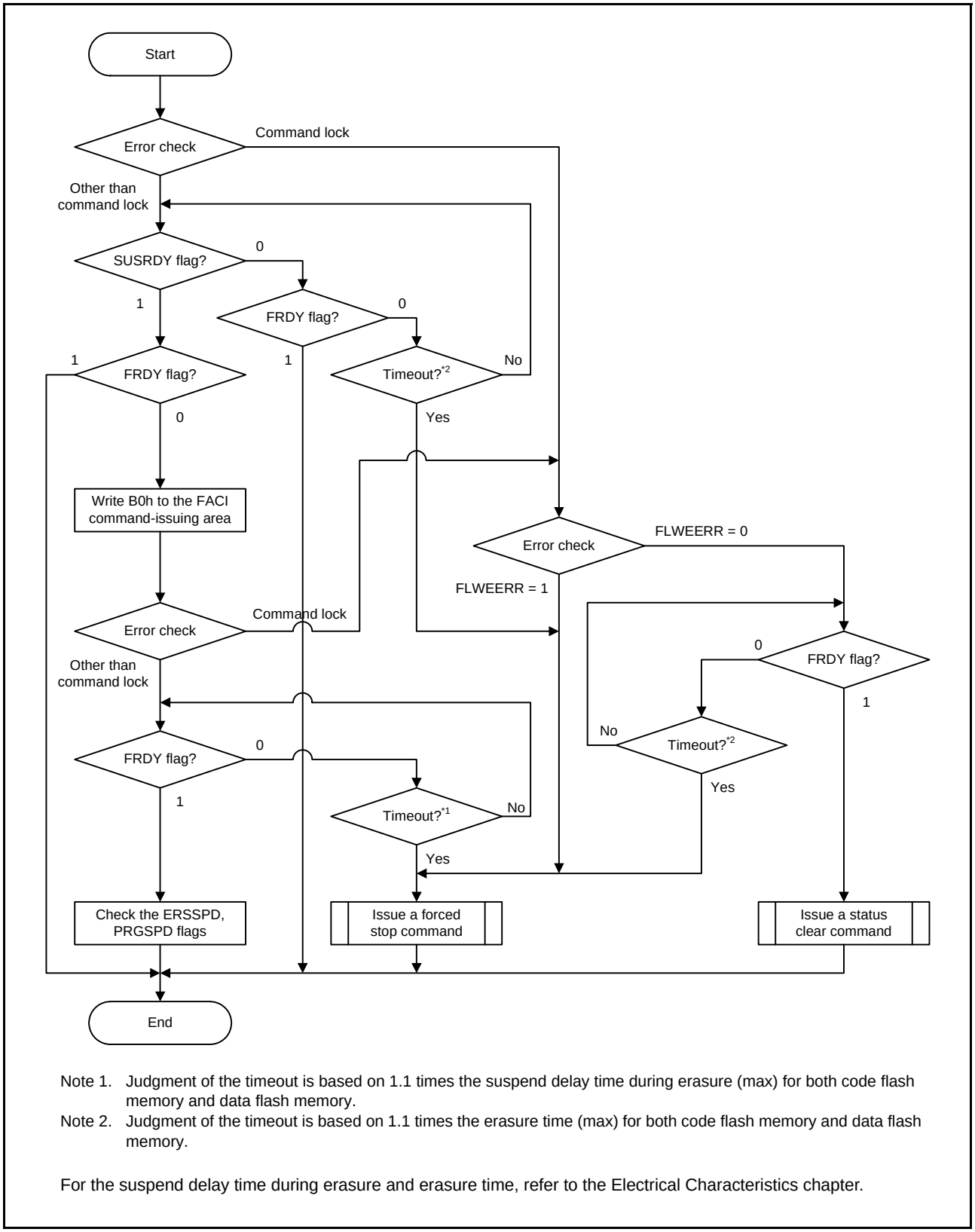
55.8.3.9 P/E Suspend Command

The P/E suspend command is used to suspend programming or erasure. Before issuing a P/E suspend command, check that the FASTAT.CMDLK flag is 0, and the execution of programming/erasure is normally performed. To confirm that the P/E suspend command can be received, also check that the FSTATR.SUSRDY flag is 1. After issuing a P/E suspend command, read the FASTAT.CMDLK flag to confirm that its value is not 1 (the flash sequencer is not in the command-locked state).

If an error occurs during programming or erasure processing, the FASTAT.CMDLK flag is set to 1. When P/E processing is completed between the FSTATR.SUSRDY flag having been confirmed to be 1 and acceptance of the P/E suspension command, the P/E suspension command is ignored and the flash sequencer does not enter the suspended state (the FSTATR.FRDY flag is 1 and the ERSSPD and PRGSPD flags in the FSTATR register are 0).

When a P/E suspend command is received and then the programming/erasure suspend processing finishes normally, the flash sequencer enters the suspended state, the FSTATR.FRDY flag is set to 1, and the ERSSPD or PRGSPD flag in the FSTATR register is 1. After issuing a P/E suspend command, check that the ERSSPD or PRGSPD flag in the FSTATR register is 1 and the suspended state is entered, and then decide the subsequent flow. If a P/E resume command is issued in the subsequent flow although the suspended state is not entered, an illegal command error occurs and the flash sequencer shifts to the command-locked state (see section 55.10.2, Error Protection).

If the erasure suspended state is entered, programming to blocks targeted for other than erasure can be performed. Additionally, the programming and erasure suspended states can shift to read mode by clearing the FENTRYR register.



Note 1. Judgment of the timeout is based on 1.1 times the suspend delay time during erasure (max) for both code flash memory and data flash memory.
Note 2. Judgment of the timeout is based on 1.1 times the erasure time (max) for both code flash memory and data flash memory.

For the suspend delay time during erasure and erasure time, refer to the Electrical Characteristics chapter.

Figure 55.18 Usage of the P/E Suspend Command

(1) Suspension during Programming

When issuing a P/E suspend command during the flash memory programming, the flash sequencer suspends programming processing. Figure 55.19 shows the suspend operation of programming. When receiving a programming-related command, the flash sequencer clears the FSTATR.FRDY flag to 0 to start programming. When the flash sequencer enters the state in which the P/E suspend command can be received after starting programming, it sets the FSTATR.SUSRDY flag to 1. When a P/E suspend command is issued, the flash sequencer receives the command and clears the FSTATR.SUSRDY flag to 0. When the flash sequencer receives a P/E suspend command while a programming pulse is being applied, the flash sequencer continues applying the pulse. After the specified pulse application time, the flash sequencer finishes pulse application, and starts the programming suspend processing and sets the FSTATR.PRGSPD flag to 1.

When the suspend processing finishes, the flash sequencer sets the FSTATR.FRDY flag to 1 to enter the programming suspended state. When receiving a P/E resume command in the programming suspended state, the flash sequencer clears the FSTATR.FRDY and FSTATR.PRGSPD flags to 0 and resumes programming.

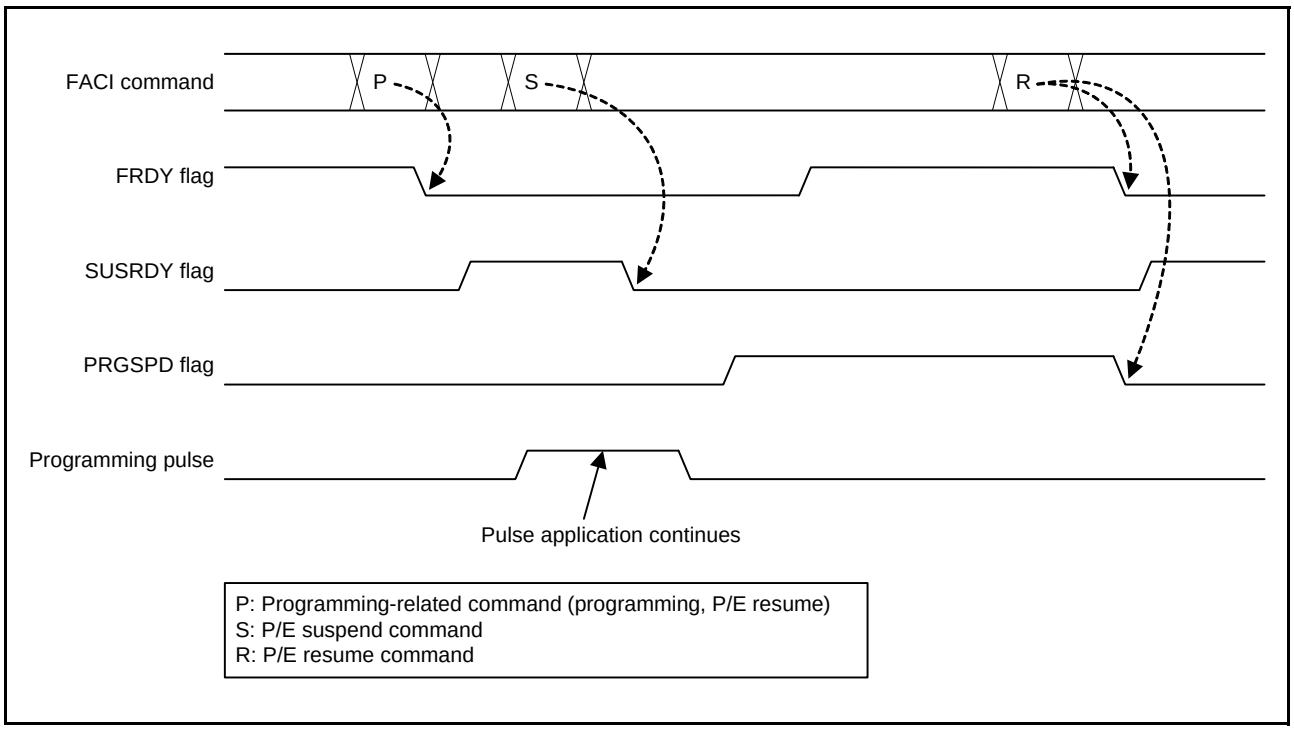


Figure 55.19 Suspension during Programming

(2) Suspension during Erasure (Suspension Priority Mode)

This MCU has a suspension priority mode for the suspension of erasure. Figure 55.20 shows the suspend operation of erasure in suspension priority mode (the FCPSR.ESUSPMD bit is 0).

When receiving an erasure-related command, the flash sequencer clears the FSTATR.FRDY flag to 0 to start erasure. When the flash sequencer enters the state in which the P/E suspend command can be received after starting erasure, it sets the FSTATR.SUSRDY flag to 1. When a P/E suspend command is issued, the flash sequencer receives the command and clears the FSTATR.SUSRDY flag to 0. When receiving a suspend command during erasure, the flash sequencer starts the suspend processing and sets the FSTATR.ERSSPD flag to 1 even if it is applying an erasure pulse. When the suspend processing is completed, the flash sequencer sets the FSTATR.FRDY flag to 1 to enter the erasure suspended state. When receiving a P/E resume command in the erasure suspended state, the flash sequencer clears the FRDY and ERSSPD flags in the FSTATR register to 0 and resumes erasure. Operations of the FRDY, SUSRDY, and ERSSPD flags in the FSTATR register 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 flash sequencer 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 flash sequencer continues applying erasure pulse A. After the specified pulse application time, the flash sequencer finishes erasure pulse application and enters the erasure suspended state. When the flash sequencer receives a P/E resume command next and erasure pulse B starts to be newly applied, and then the flash sequencer 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.

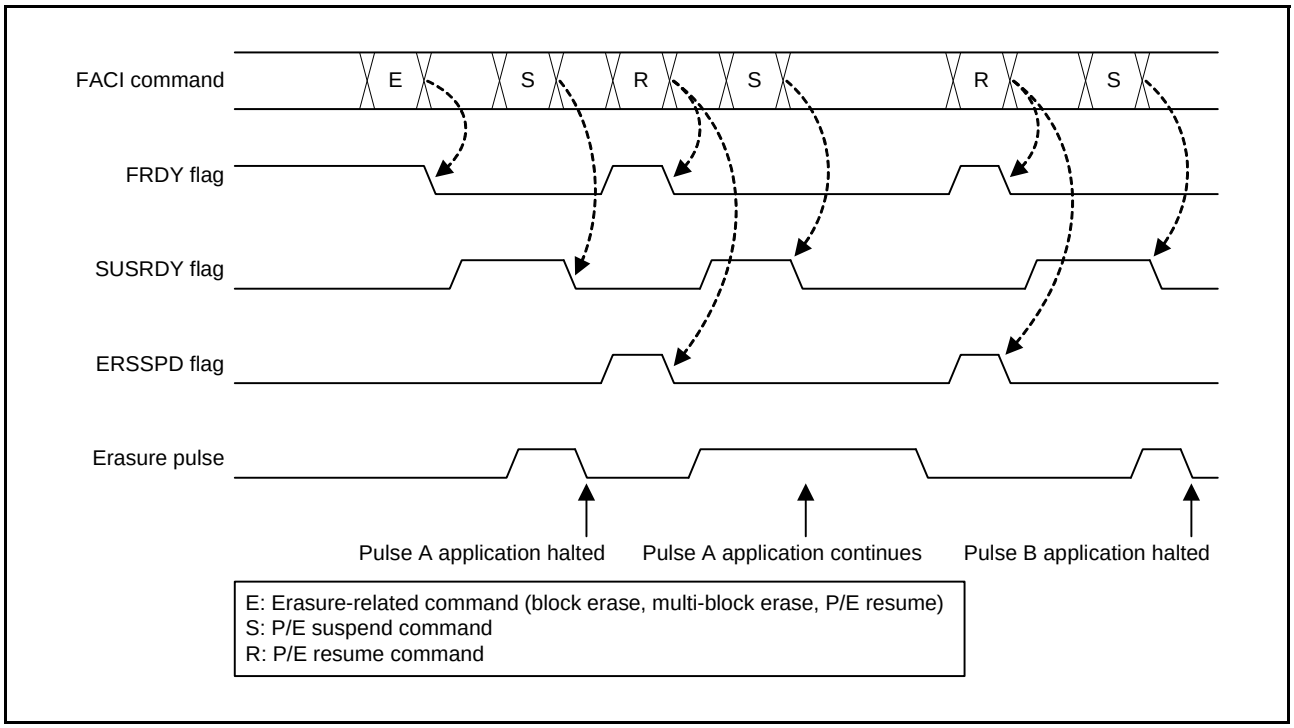


Figure 55.20 Suspension during Erasure (Suspension Priority Mode)

(3) Suspension during Erasure (Erasure Priority Mode)

This MCU has an erasure priority mode for the suspension of erasure.

Figure 55.21 shows the suspend operation of erasure when the erasure suspend mode is set to the erasure priority mode (the 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.

When the flash sequencer receives a P/E suspend command while an erasure pulse is being applied, the flash sequencer 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.

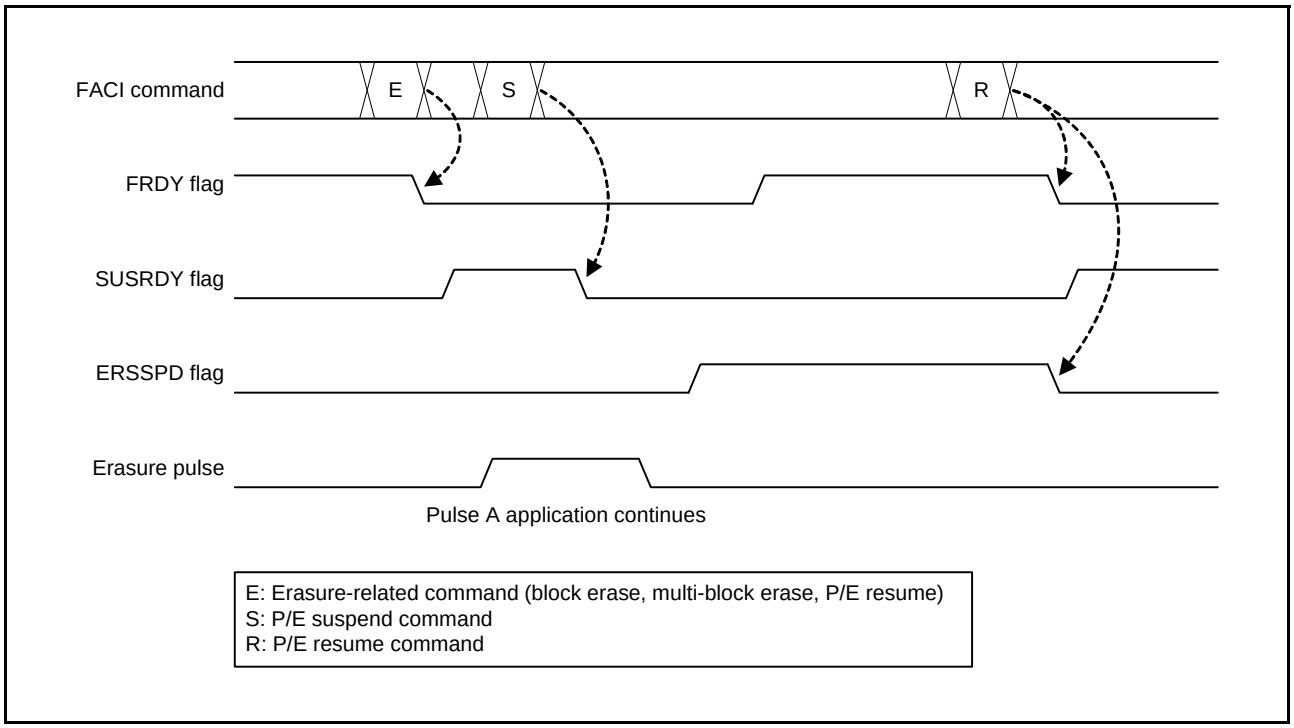
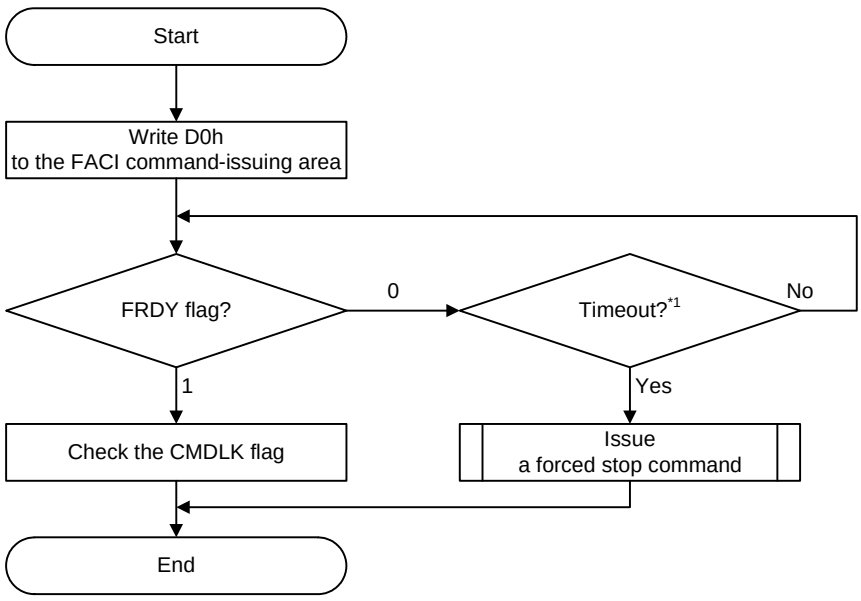


Figure 55.21 Suspension during Erasure (Erasure Priority Mode)

55.8.3.10 P/E Resume Command

To resume suspended programming or erasure, use the P/E resume command. When the settings of the FENTRYR register are changed during suspension, reset the setting of the FENTRYR register to the value immediately before the P/E suspend command was issued, and then issue a P/E resume command. Completion of processing of the resumed command can be confirmed by reading the FSTATR.FRDY flag.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time for halted processing of the FACI command (refer to the Electrical Characteristics chapter).

Figure 55.22 Usage of the P/E Resume Command

55.8.3.11 Status Clear Command

When one of the ILGLERR, ILGCOMERR, FESETERR, SECERR, OTERR, ERSERR, PRGERR, and FLWEERR flag bits in the FSTATR register is set to 1, the flash sequencer enters the command-locked state. When one of the CFAE and DFAE flags in the FASTAT register is set to 1, the flash sequencer also enters the command lock state. In the command-locked state, the flash sequencer can accept only status clearing command or forced end command.
The status clear command is used to clear the command-locked state (see section 55.8.3.5, Recovery from the Command-Locked State). To clear the CFAE, DFAE, and CMDLK flags in the FASTAT register, and the ILGLERR, ILGCOMERR, FESETERR, SECERR, OTEERR, ERSERR, and PRGERR flags in the FSTATR register in the command-locked state, the status clear command is available.
The FLWEERR flag cannot only be cleared by the status clearing command, but can only be cleared by the forced end command.

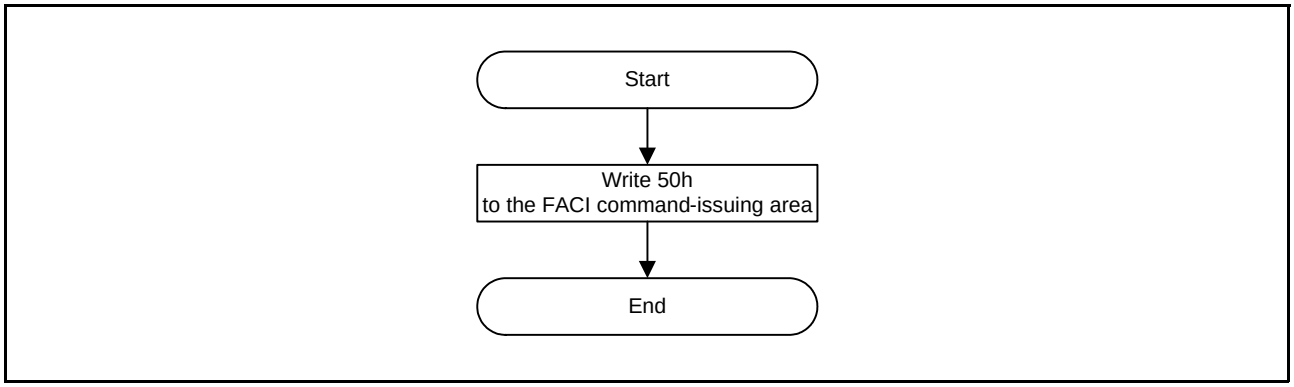
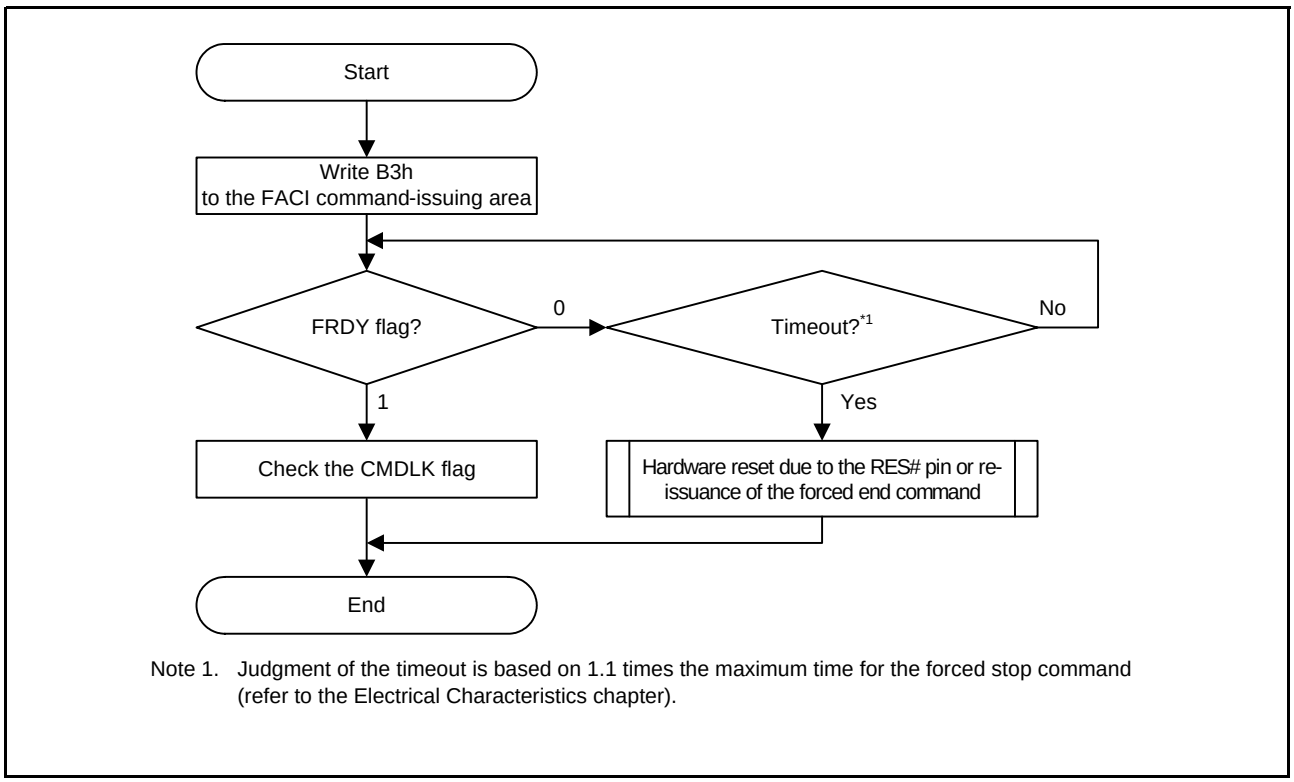


Figure 55.23 Usage of the Status Clear Command

55.8.3.12 Forced Stop Command

The forced stop command forcibly ends command processing by the flash sequencer. Although this command halts command processing in higher speed than the P/E suspension command, values from the area where programming or erasure was in progress are not guaranteed. Furthermore, resumption of processing is not possible. Processing of programming or erasure that was terminated by the forced stop command is also defined as one round of programming. Executing a forced stop command also initializes the whole FCU and a part of the FACI, and the FASTAT and FSTATR registers. Accordingly, this command can be used in the procedure for recovery from the command-locked state and in processing in response to a time-out of the flash sequencer (see section 55.8.3.5, Recovery from the Command-Locked State).



Note 1. Judgment of the timeout is based on 1.1 times the maximum time for the forced stop command (refer to the Electrical Characteristics chapter).

Figure 55.24 Usage of the Forced Stop Command

How to Use the Forced Stop Command when a Command is being issued

When processing is terminated by the forced stop command while a timeout is generated based on the DBFULL bit judgment of the program command, writing to the FACI command issuing area may be handled as the data written by the program command. In this case, read the FACI command issuing area and generate a command lock intentionally, then issue the forced stop command according to the method of returning from the command lock state. A command lock can be generated although the access size for reading the FACI command issuing area is in 8-, 16-, or 32-bit units.

55.8.3.13 Blank Check Command

Values read from data flash memory that has been erased but not yet been programming again (i.e. that is in the non-programmed state) are undefined. Use the blank check command when you need to confirm that an area is in the non-programmed state.

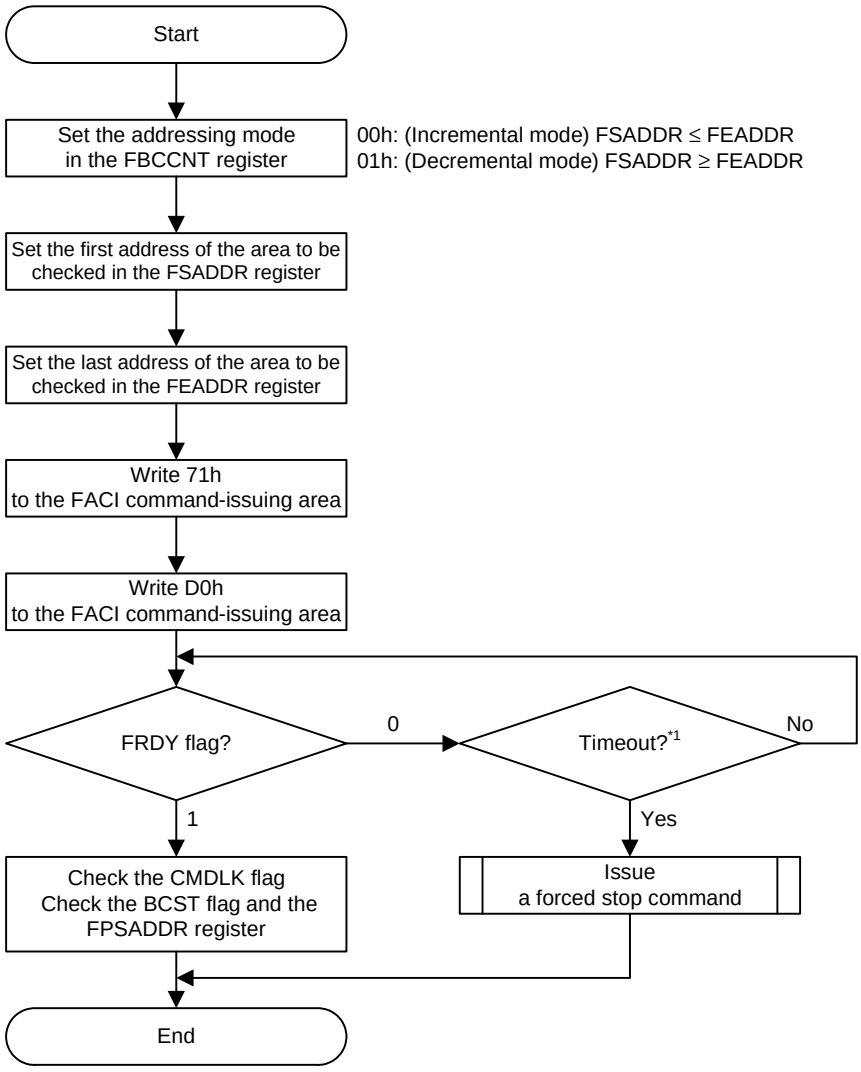
Before issuing a blank check command, set addressing mode, start and end addresses of the target area for blank checking to the FBCCNT, FSADDR, and FEADDR registers.

When the FBCCNT.BCDIR bit is 1, the value specified in the FSADDR register must be set at least the value specified in the FEADDR register.

When the FBCCNT.BCDIR bit is 0, the value specified in the FSADDR register must be the value specified in the FEADDR register or less.

When the settings of the FBCCNT.BCDIR bit, FSADDR register, and FEADDR register are inconsistent, the flash sequencer enters the command-locked state. The size of the target area for blank checking is in the range from 4 bytes to 8 Kbytes and is set in units of 4 bytes.

Write 71h and D0h to the FACI command-issuing area to start blank checking. Completion of processing can be confirmed by the FSTATR.FRDY flag. At the end of processing, the result of blank checking is stored in the FBCSTAT.BCST flag. If the target area for blank checking includes areas where programming has been completed, the flash sequencer stores the address of the programmed data that it first detected in the FPSADDR register.

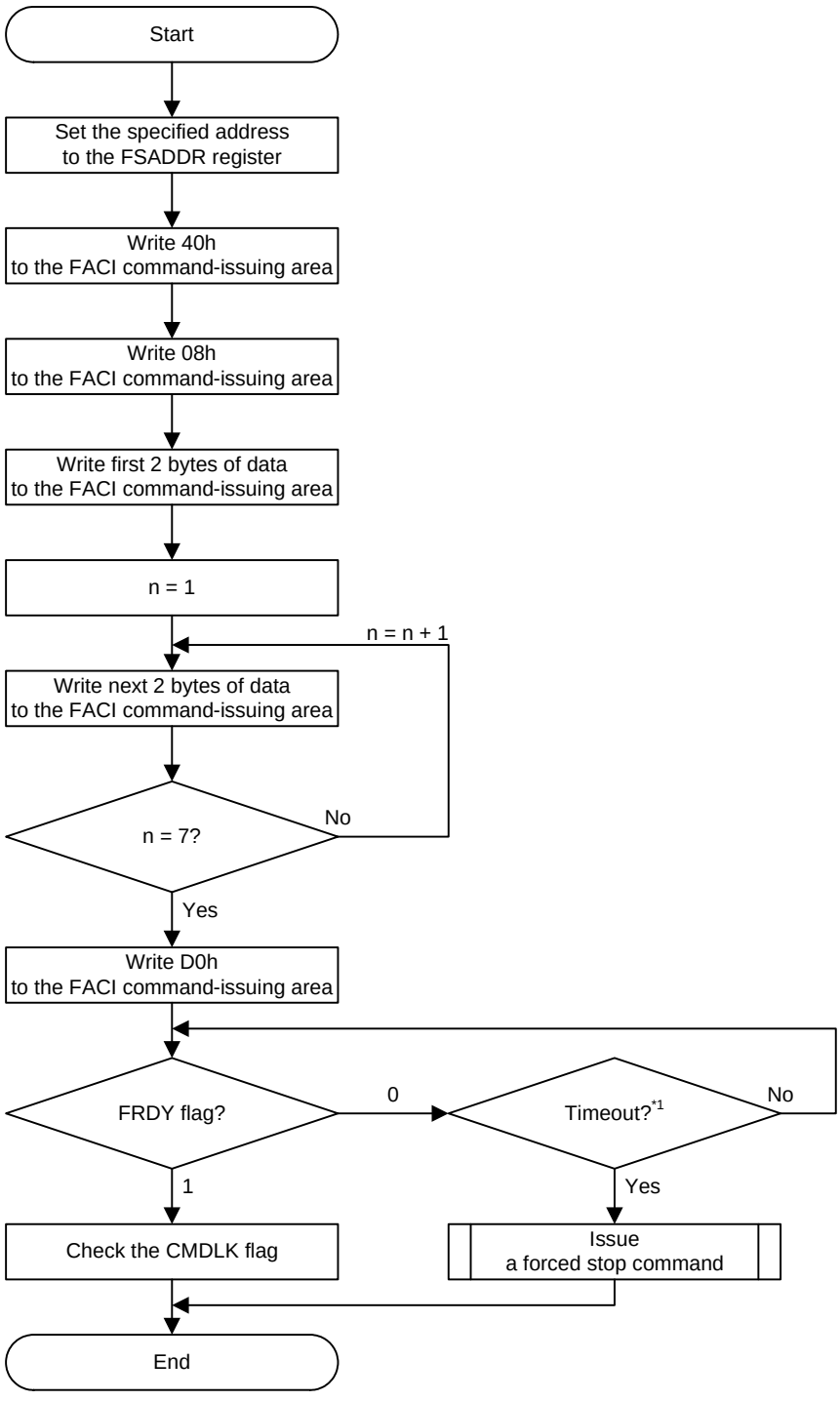


Note 1. Judgment of the timeout is based on 1.1 times the blank checking time (max.) for the entire target area. For the blank checking time, refer to the Electrical Characteristics chapter.

Figure 55.25 Usage of the Blank Check Command

55.8.3.14 Configuration Set Command

The configuration set command is used to set the option-setting memory (configuration setting area).
Before issuing a configuration set command, set the specified address (shown in Table 55.15) in the FSADDR register.
Writing D0h to the FACI command-issuing area in the final access for issuing the FACI command starts processing of the configuration set command.



Note 1. Timeout is judged to be after 185 ms (when FCLK ≥ 20 MHz), 205 ms (when FCLK = 15 MHz), or 340 ms (when FCLK = 4 MHz).

Figure 55.26 Usage of the Configuration Set Command

The correspondence between the possible target data for configuration setting and the address value set in the FSADDR register is shown in Table 55.15. For details on the FSADDR register, see section 55.4.9, FACI Command Start Address Register (FSADDR).

Table 55.15 Address Used by Configuration Set Command

Address	FSADDR Register Value	Setting Data	Operation of additional writing		Timing when the Setting is Enabled
			FAW.FSPR bit is 1	FAW.FSPR bit is 0	
FE7F 5D00h	00FF 5D00h	Option function select register 0 (OFS0), option function select register 1 (OFS1), endian select register (MDE)	Writable	Writable	At a reset
FE7F 5D10h	00FF 5D10h	TM identification data register (TMINF)	Writable	Writable	At a reset
FE7F 5D20h	00FF 5D20h	Bank select register (BANKSEL)	Writable	Writable	At a reset
FE7F 5D40h	00FF 5D40h	Serial programmer command control register (SPCC), TM enable flag register (TMEF)	Writable*1 (from 1 to 0 only)	Writable*1 (from 1 to 0 only)	When a reset or command is executed*3
FE7F 5D50h	00FF 5D50h	OCD/serial programmer ID setting register (OSIS)	Writable	Writable	At a reset
FE7F 5D64h	00FF 5D60h	Flash access window setting register (FAW)*2	Writable	Not writable*2	When a reset or command is executed
FE7F 5D70h	00FF 5D70h	ROM code protect register (ROMCODE)	Writable	Writable	At a reset

Note 1. Once these bits are set to 0, the bits cannot be restored to 1 by using the configuration setting command.
Note 2. The FAW.FSPR bit cannot be restored to 1 by using the configuration setting command once it is set to 0. Therefore, setting the access window and start-up area select flags again becomes impossible. (when the configuration setting command is issued to the address of FE7F 5D64h, the command is locked.) Exercise extra caution when handling the FAW.FSPR bit.
Note 3. The setting in the serial programmer command control register (SPCC) is enabled after a reset. The setting of the TM enable flag register (TMEF) is enabled when a reset or command is executed.

55.9 Suspend Operation

Reading from the flash memory is not possible during programming or erasure if the conditions for background operation given in Table 55.25 are not satisfied. When a P/E suspend command is issued to suspend the programming or erasure of the flash memory, reading from the flash memory is enabled. Regarding P/E suspend commands, there are one suspend command mode for programming and two suspend command modes for erasure (suspension priority mode and erasure priority mode). To resume suspended programming or erasure, the P/E resume command is available.

55.10 Protection Functions

55.10.1 Software Protection

Software protection disables programming and erasure for the flash memory through the settings of control registers. If an attempt is made to issue an FACI command against software protection, the flash sequencer enters the command-locked state.

55.10.1.1 Protection through FWEPROR

Programming cannot proceed in any mode unless the FWEPROR.FLWE[1:0] bits are set to 01b.

55.10.1.2 Protection through FENTRYR

When the FENTRYR register is set to 0000h, the flash sequencer enters read mode. In read mode, FACI commands cannot be accepted. If an attempt is made to issue an FACI command in read mode, the flash sequencer enters the command-locked state.

55.10.2 Error Protection

Error protection detects erroneous issuance of FACI commands, unauthorized access, and flash sequencer malfunction. FACI command acceptance is disabled (command-locked state) in response to the detection of these errors. The flash memory cannot be programmed or erased while the flash sequencer is in the command-locked state. For release from the command-locked state, issue a status clear or forced stop command. The status clear command can only be used while the FSTATR.FRDY flag is 1. The forced stop command can be used regardless of the value of the FSTATR.FRDY flag. Generation of a flash access error (FIFERR) interrupt detects malfunction. An FIFERR interrupt is generated under any of the following conditions.

- When the flash sequencer enters the command lock state (when the FASTAT.CMDLK flag is 1) while the FAEINT.CMDLKIE bit is 1
- When violation in access to the data flash memory occurred (the FASTAT.DFAE flag is 1) while the FAEINT.DFAEIE bit is 1
- When violation in access to the code flash memory occurred (when the FASTAT.CFAE flag is 1) while the FAEINT.CFAEIE bit is 1

When the flash sequencer enters the command-locked state in response to a command other than the P/E suspend command during programming or erasure processing, the flash sequencer continues the processing for programming or erasure. In this state, the P/E suspend command cannot be used to suspend the processing for programming or erasure. If a command is issued in the command-locked state, the ILGLERR and ILGCOMERR flags in the FSTATR register becomes 1 and the other flags retain the values set due to previous error detection.
Table 55.16 shows error protection types and status bit values after error detection.

Table 55.16 Error Protection Type (1/2)

Error Type	Description	ILGCOMERR	FESETERR	SECERR	OTERR	ILGLERR	ERSERR	PRGERR	FLWEERR	CFAE	DFAE
FENTRYR setting error	The FENTRYR setting is AA81h	0	1	0	0	1	0	0	0	0	0
	The FENTRYR setting at suspension disagrees with that at resumption	0	1	0	0	1	0	0	0	0	0
Illegal command error	Access with an undefined unit was attempted in the first access by the FACI command (not writing of a byte)	1	0	0	0	1	0	0	0	0	0
	Access with an undefined code was attempted in the first access by the FACI command	1	0	0	0	1	0	0	0	0	0
	The value specified in the last access of the multiple-access FACI command is not D0h.	1	0	0	0	1	0	0	0	0	0
	The value “N” (see Table 55.11) specified in the second write access of an FACI command in the programming or configuration setting command is wrong.	1	0	0	0	1	0	0	0	0	0
	When a blank check command has been issued under one of the following conditions • When the FBCCNT.BCDIR bit is 0, and the value in FSADDR > that in FEADDR • When the FBCCNT.BCDIR bit is 1, and the value in FEADDR > that in FSADDR • The setting range of bits 16 to 0 in FEADDR is between 0 2000h and 1 FFFFh	1	0	0	0	1	0	0	0	0	0/1 *1
	When a multi-block erase command has been issued under one of the following settings. • The value in FSADDR > that in FEADDR • The setting range of bits 16 to 0 in FEADDR is between 0 2000h and 1 FFFFh	1	0	0	0	1	0	0	0	0	0/1 *1
	An FACI command not acceptable in each mode has been issued (see Table 55.12)	1	0	0	0	1	0	0	0	0	0
	Program or block erase command has been issued to the area protected by the area protection	1	0	0	0	1	0	0	0	0	0
	A program command has been issued to the area where the erasure has been suspended during the processing of erase suspension	1	0	0	0	1	0	0	0	0	0
	An FACI command has been issued when command acceptance conditions are not satisfied (see Table 55.13)	0/1 *2	0/1 *2	0/1 *2	0/1 *2	1	0/1 *2	0/1 *2	0/1 *2	0/1 *2	0/1 *2
Erasure error	An error occurs during erasure	0	0	0	0	0	1	0	0	0	0
Programming error	An error occurs during programming	0	0	0	0	0	0	1	0	0	0
Code flash memory access violation	When a program command or block erase command has been issued under the following settings in code flash memory P/E mode • The setting value of bits 23 to 0 in FSADDR is within the range from 00 0000h to DF FFFFh	0	0	0	0	1	0	0	0	1	0
	When a configuration program command has been issued under the following settings in code flash memory P/E mode • The setting value of bits 9 to 0 in the FSADDR register ranges from 000h to 0FFh, or from 180h to 3FFh	0	0	0	0	1	0	0	0	1	0

Table 55.16 Error Protection Type (2/2)

Error Type	Description	ILGCOMERR	FESETERR	SECERR	OTERR	ILGLERR	ERSERR	PRGERR	FLWEERR	CFAE	DFAE
Data flash memory access violation	When a program command or block erase command has been issued under the following settings in data flash memory P/E mode • The setting value of bits 16 to 0 in FSADDR is within the range from 0 2000h to 1 FFFFh	0	0	0	0	1	0	0	0	0	1
	When a multi-block erase command or blank check command has been issued under the following settings in data flash memory P/E mode • The setting value of bits 16 to 0 in FSADDR is within the range from 0 2000h to 1 FFFFh	1	0	0	0	1	0	0	0	0	1
Security error	When the FAW.FSPR bit is 0, a configuration setting command is issued for the settings of access window and the FAW.BTFLG bit	0	0	1	0	1	0	0	0	0	0
Others	The FACI command-issuing area has been accessed in read mode	0	0	0	1	1	0	0	0	0	0
	The FACI command-issuing area has been read in code flash memory P/E mode or data flash memory P/E mode	0	0	0	1	1	0	0	0	0	0
FLWE error	When the FACI command is processed, disable programming or erasure by using the FWEPROR register*3	0	0	0	0	0	0/1	0/1	1	0	0

Note 1. If the setting value in the FSADDR register matches the condition for violations in access to the data flash memory, the FASTAT.DFAE bit is set to 1.
Note 2. This is the value when a command is executed.
Note 3. For details on the FWEPROR register, see section 55.4.5, Flash P/E Protect Register (FWEPROR).

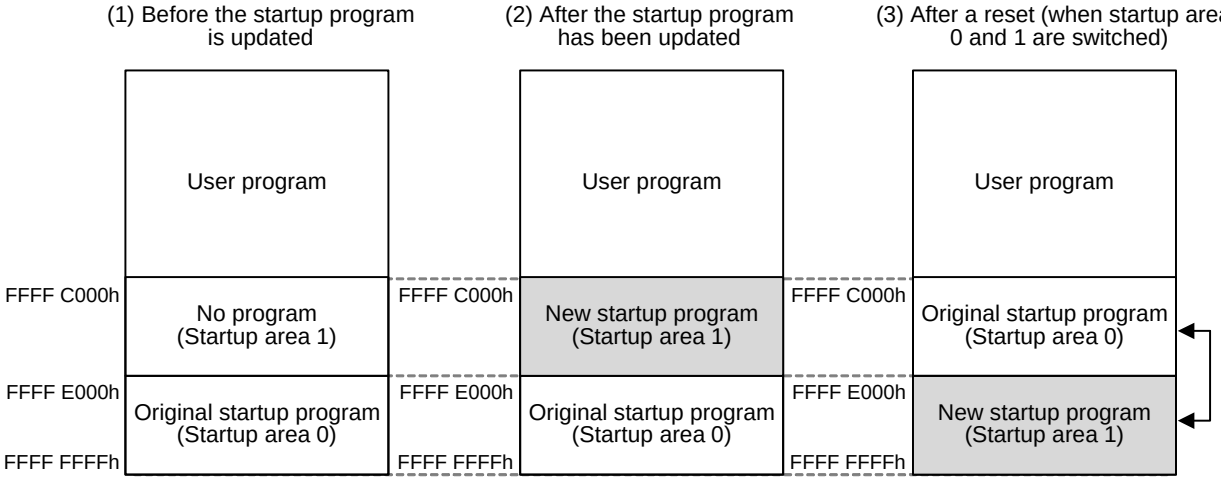
55.10.3 Start-Up Program Protection

Protection of the startup program is for protection of the program to be started after a reset (the startup program). This function provides a way to safely update the startup program when rewriting is suspended during a reset.

The startup area is 8 Kbytes in size and is assigned to the user area in the code flash memory. This function uses the values of the FAW.BTFLG bit and the FSUACR.SAS[1:0] bits to change the area where the startup program is stored in block units (see Figure 55.27 to Figure 55.30).

In protection of the startup program, the state of the selection of the startup area can be fixed by the access window protection bit (FAW.FSPR). However, the FAW.FSPR bit never be restored to 1 once the flag is set to 0. Exercise extra caution when handling the FAW.FSPR bit.

In addition, this protection cannot be used when dual mode is selected by the bank mode switching function (when the MDE.BANKMD[2:0] bits are 000b).



Note: The addresses are used for reading, programming, and erasure. Startup area 0 is erased when a block erase command is issued for the address FFFF C000h with allocation shown under (3).

(1) Before the startup program is updated
When the FAW.BTFLG bit is 1 and the SAS[1:0] bits are 0xb, the address range from FFFF C000h to FFFF DFFFh is selected as startup area 1 and the address range from FFFF E000h to FFFF FFFFh is selected as startup area 0.

(2) After the startup program has been updated
A new startup program can be written to startup area 1. At this time, startup area 0 will still contain the previous startup program, so rewriting by starting up from startup area 0 remains possible if rewriting to the startup area 1 fails.

(3) After a reset (when startup areas 0 and 1 are switched)
When rewriting startup area 1 is successfully completed, switch the assignment of startup areas 0 and 1 by rewriting the startup area selection bit (the FAW.BTFLG bit) with the configuration setting command. Afterwards, the program is started up from startup area 1 after a reset.

Figure 55.27 Concept of Protection of the Startup Program

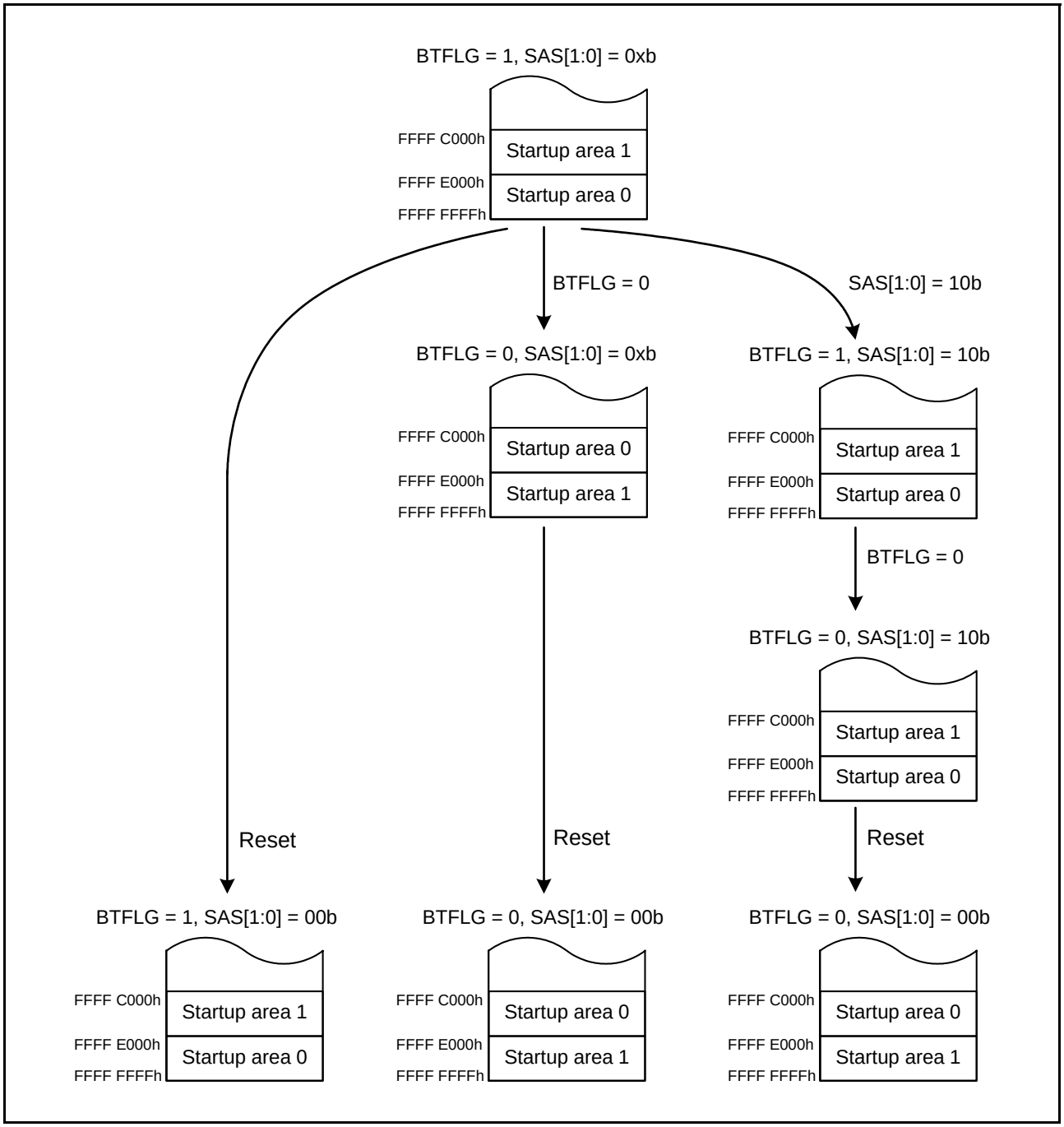


Figure 55.28 Example 1 of Transitions for Startup Program Protection Settings

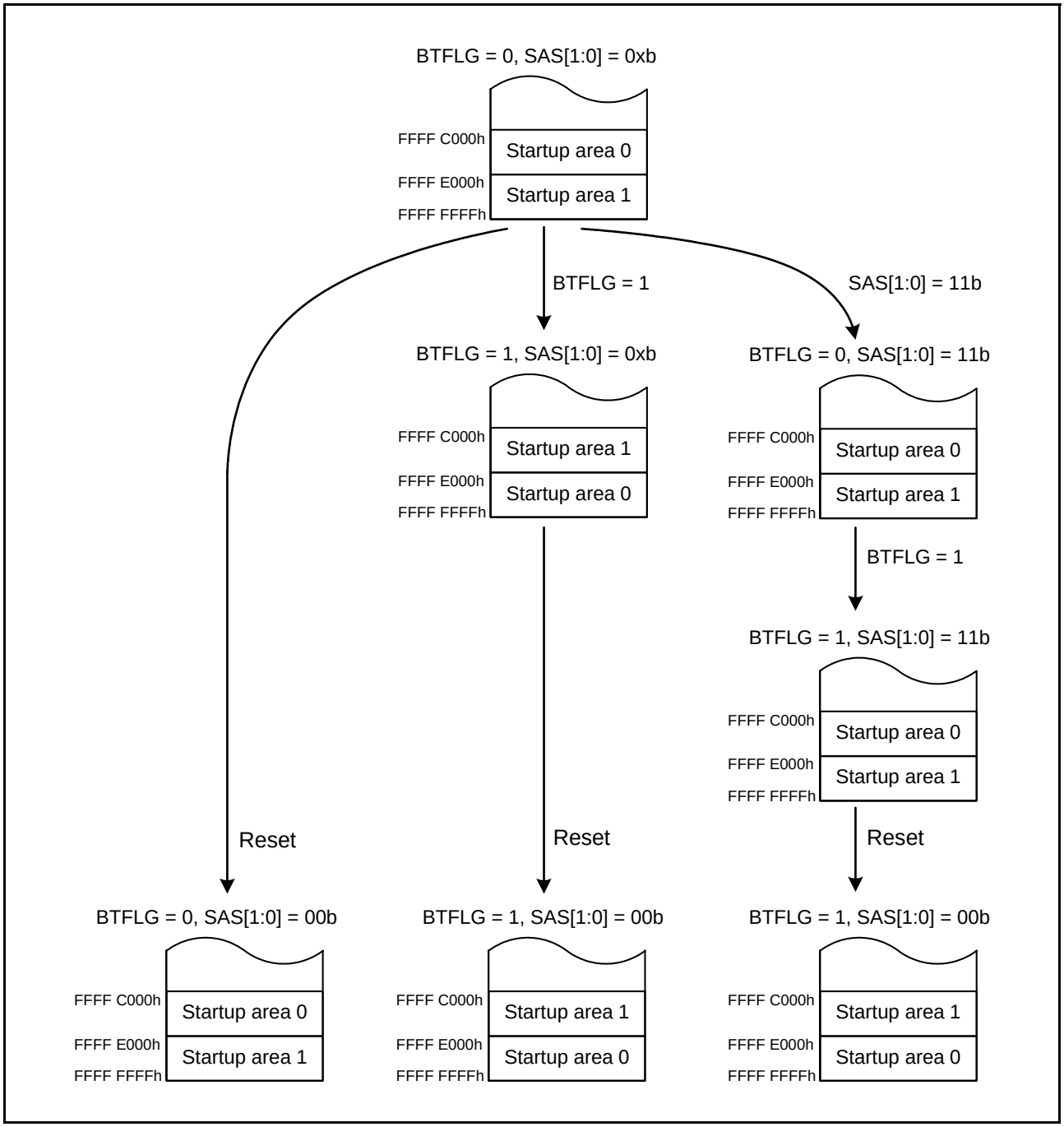


Figure 55.29 Example 2 of Transitions for Startup Program Protection Settings

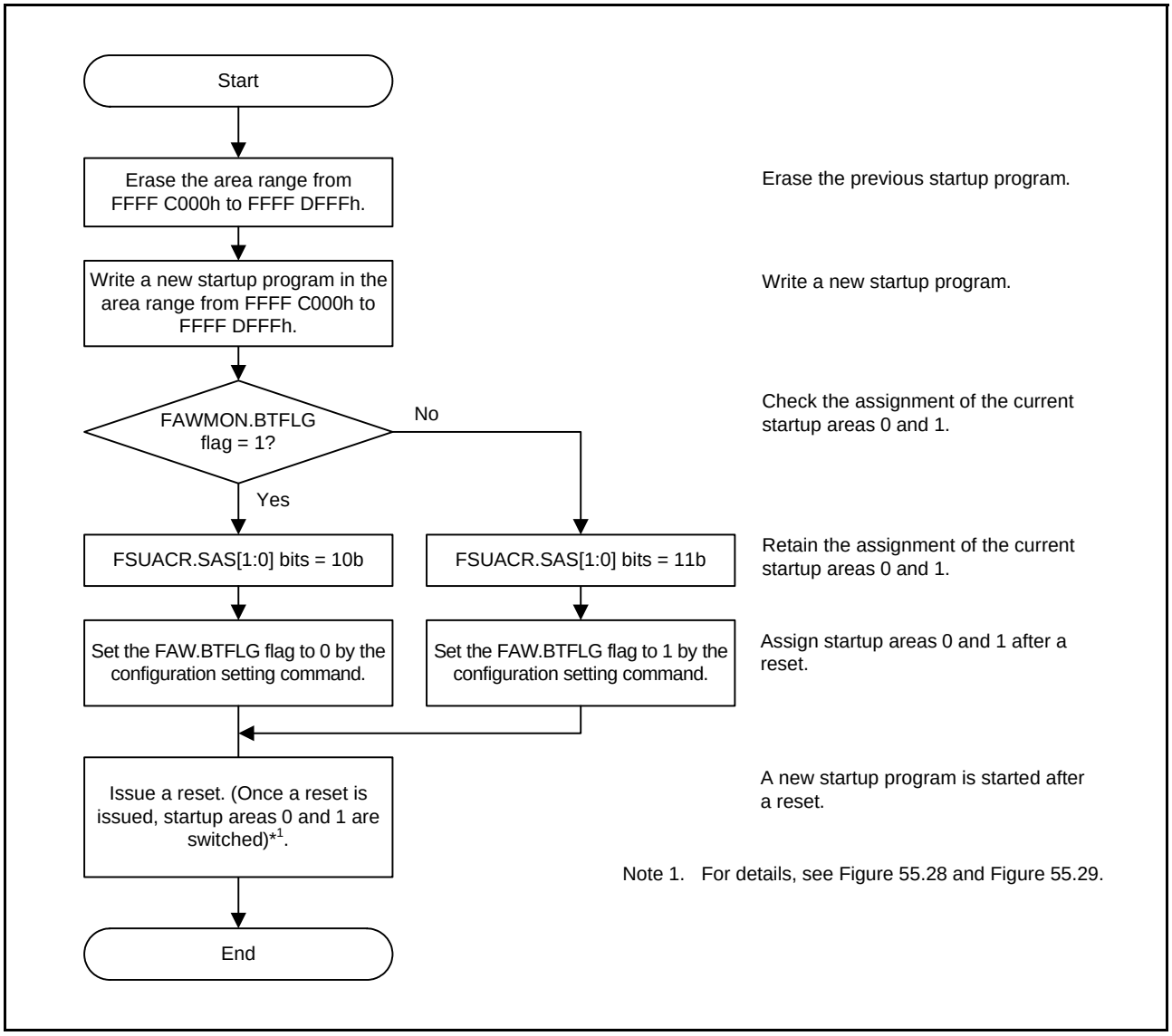


Figure 55.30 Example of Flowchart for Selecting Startup Area

55.10.4 Protection by Area Protection

Issuing an FACI command that programs or erases the area set outside the access window leads to the command-locked state. The access window is valid only in the user area of the code flash memory. The access window is valid in self-programming mode or serial programming mode.
The access window can be set in block units.
Figure 55.31 shows the Area Protection Overview.

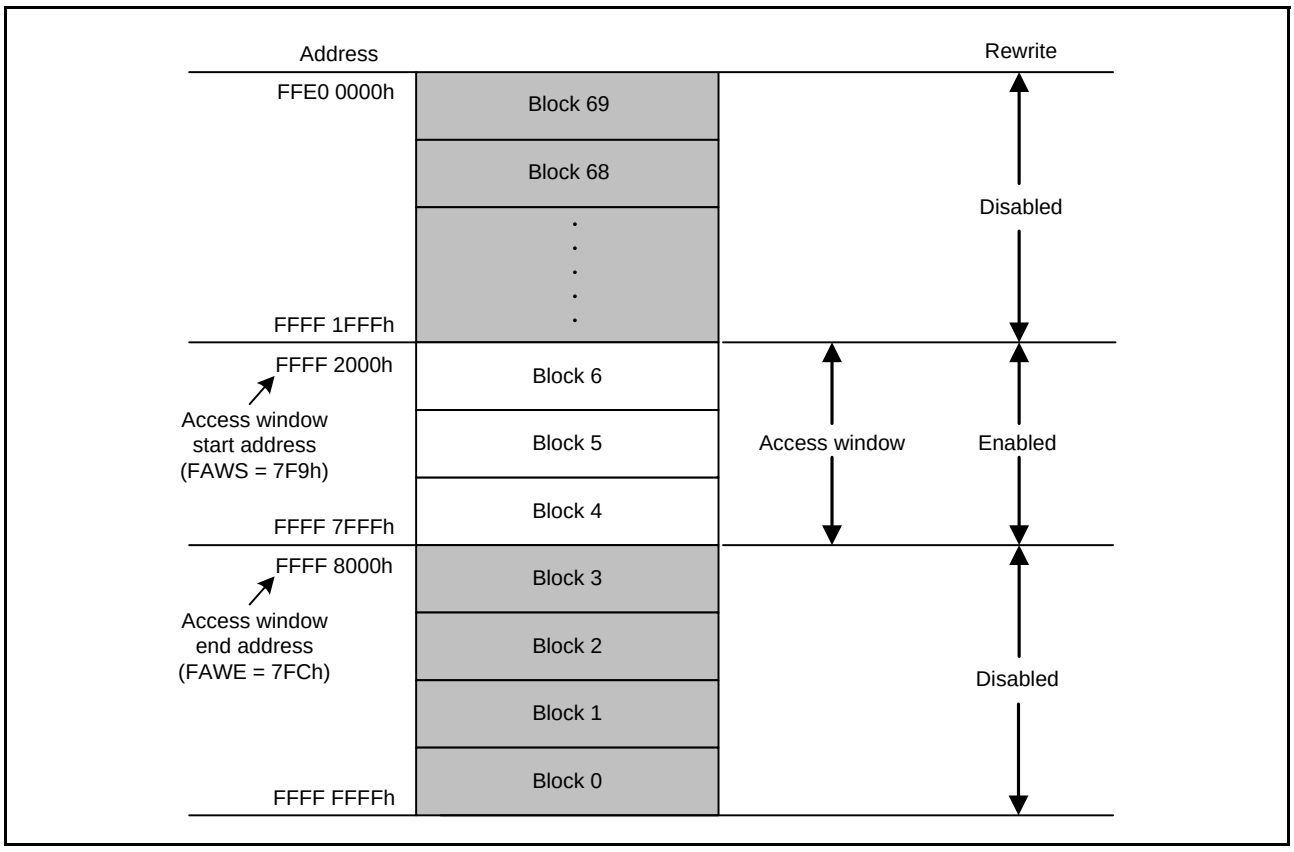


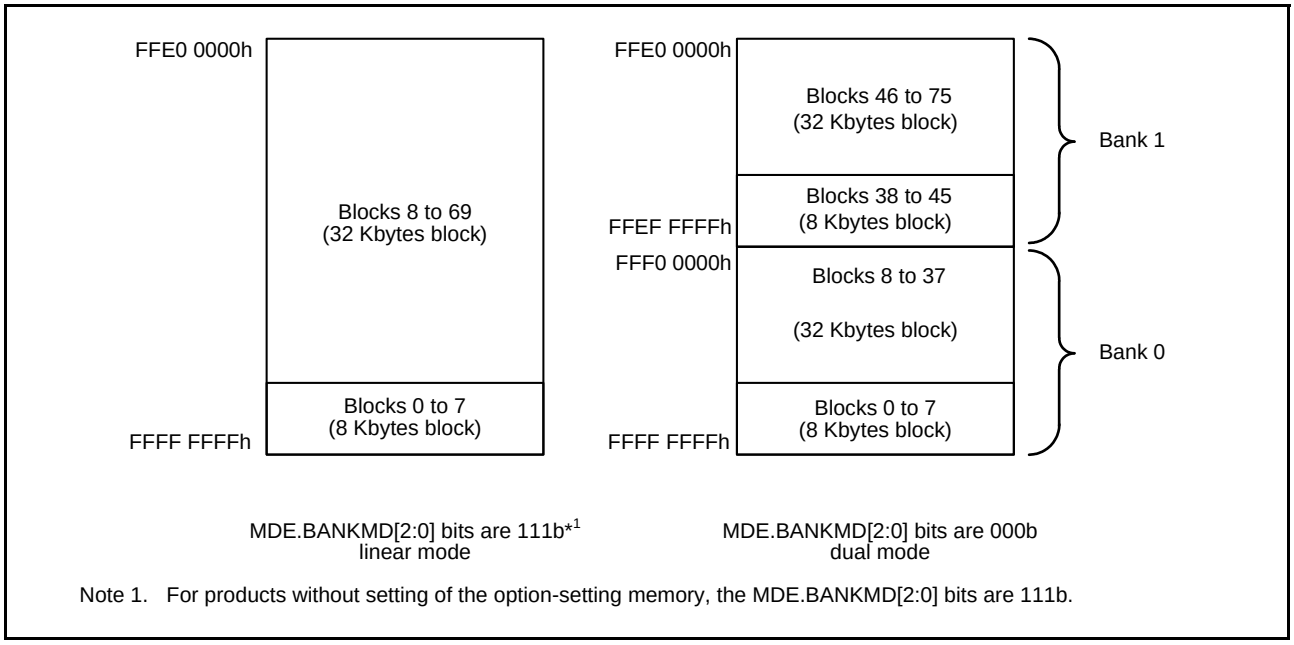
Figure 55.31 Area Protection Overview (When Blocks 4 to 6 are Set as the Access Window in Products with 2 Mbytes of Code Flash Memory)

55.10.5 Dual Bank Function

This protection uses the functions of bank mode switching and startup bank selection to update a program while a user program is running and to provide a safe method of updating in cases where programming is suspended during a reset.

55.10.5.1 Switching Bank Modes

The bank mode switching function selects either linear mode in which the user area in the code flash memory is used as one area, or dual mode in which the user area is divided into two bank areas. Figure 55.32 shows an example of flow of switching bank modes. A reset after setting the MDE.BANKMD[2:0] bits in the option setting memory determines the mode of the bank mode switching function. Selecting dual mode enables the startup bank selection function.
When dual mode is selected by bank mode switching function (the MDE.BANKMD[2:0] bits are 000b), start-up program protection function cannot be used.

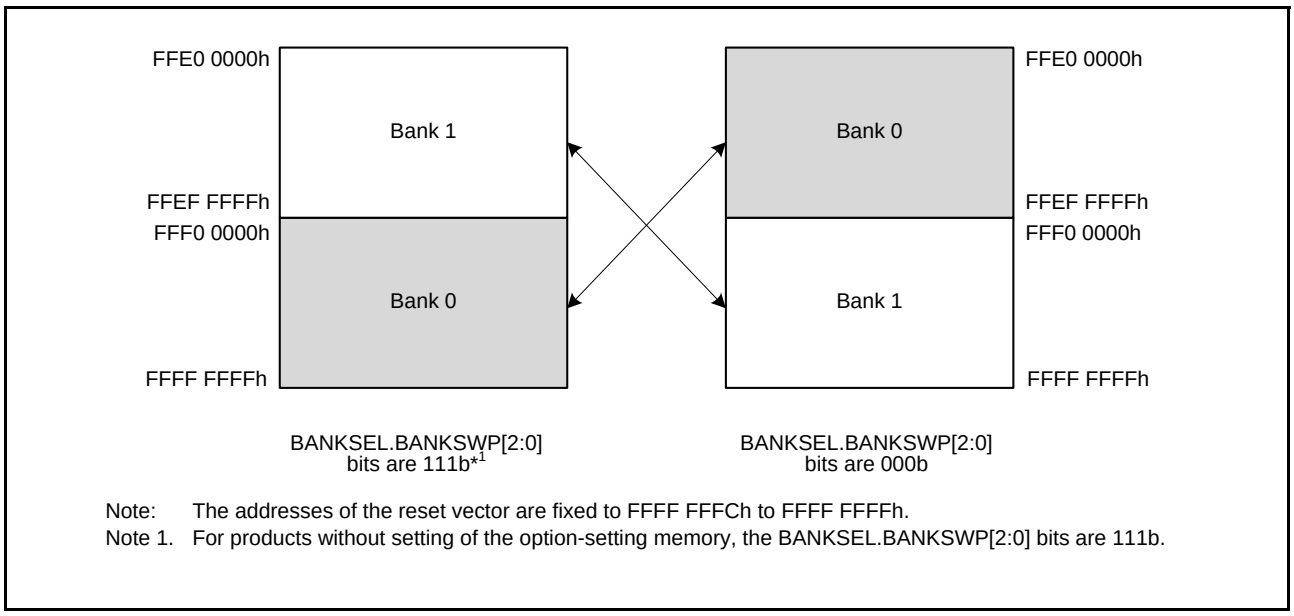


Note 1. For products without setting of the option-setting memory, the MDE.BANKMD[2:0] bits are 111b.

Figure 55.32 Example of Flow of Switching Bank Modes (For Products with 2 Mbytes of Code Flash Memory)

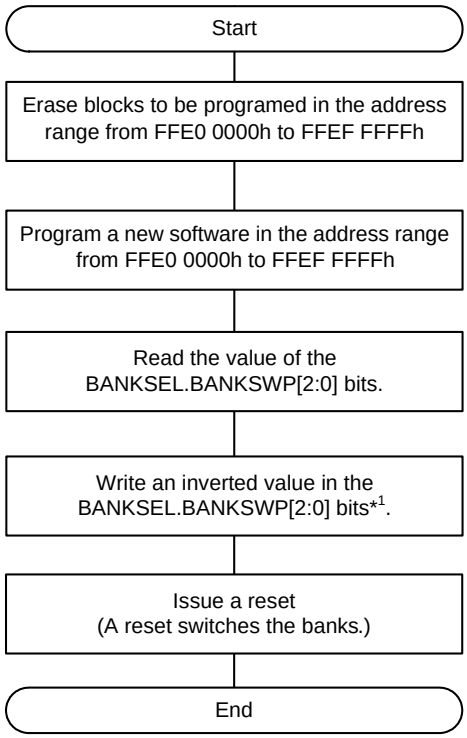
55.10.5.2 Selecting the Startup Bank

Startup bank selection provides a way to safely update the program by selecting a bank area to be started in dual mode (when the MDE.BANKMD[2:0] bits are 000b) when programming is suspended during a reset. Figure 55.33 is a schematic view of startup bank selection and Figure 55.34 shows an example of the flow of startup bank selection. A reset after setting the value of the BANKSEL.BANKSWP[2:0] bits in the option-setting memory changes the addresses of banks 0 and 1 and booting up a program proceeds from the updated area. When the address is switched by using startup bank selection, the P/E target for the FACI commands is also switched. This function is invalid in linear mode.



Note: The addresses of the reset vector are fixed to FFFF FFFCh to FFFF FFFFh.
Note 1. For products without setting of the option-setting memory, the BANKSEL.BANKSWP[2:0] bits are 111b.

Figure 55.33 Example of Startup Bank Selection (For Products with 2 Mbytes of Code Flash Memory)



Note 1. Set the inverse of the read value of the BANKSEL.BANKSWP[2:0] bits (000b or 111b).

Figure 55.34 Example of Startup Bank Selection Flow (For Products with 2 Mbytes of Code Flash Memory)

55.11 Boot Mode

There are three serial programming modes; the boot mode (SCI interface) with SCI, the boot mode (USB interface) with USB, and the boot mode (FINE interface) with FINE. Table 55.17 lists the I/O pins used in boot mode.

Table 55.17 I/O Pins Used in Boot Mode

Pin Name	I/O	Mode to be Used	Use
MD	Input	Boot mode (SCI interface) Boot mode (USB interface)	Selection of operating mode
PC7/UB	Input		Selection of boot mode (SCI interface) or boot mode (USB interface)
P30/RXD1	Input	Boot mode (SCI interface)	For host communication (to receive data through SCI)
P26/TXD1	Output		For host communication (to transmit data through SCI)
USB0_DP, USB0_DM	I/O	Boot mode (USB interface)	Data input/output of USB
P16/USB0_VBUS	Input		Detection of connection and disconnection of USB cables
P35/UPSEL	Input		Selection of USB bus-powered mode or self-powered mode
MD/FINED	I/O	Boot mode (FINE interface)	Selection of operating mode, FINE data I/O

55.11.1 Boot Mode (SCI Interface)

In boot mode (SCI interface), the host sends control commands and data for programming, and the flash memory is programmed or erased accordingly. An on-chip SCI handles transfer between the host and this MCU in asynchronous mode. Tools for transmission of control commands and the data for programming must be prepared in the host.
When this MCU is activated in boot mode (SCI interface), the program on the dedicated area the MCU is executed. The boot program automatically adjusts the bit rate of the SCI and controls programming/erasure by receiving control commands from the host.
Figure 55.35 shows the system configuration for operations in boot mode (SCI interface).

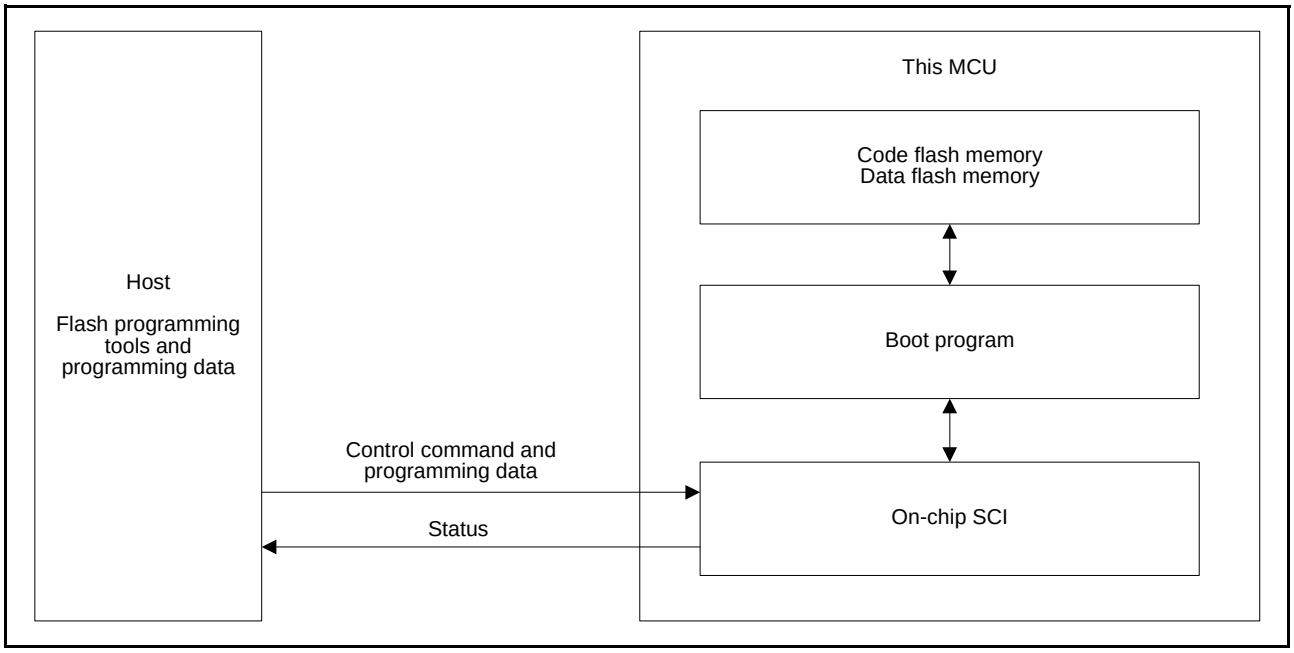


Figure 55.35 System Configuration for Operations in Boot Mode (SCI Interface)

55.11.2 Boot Mode (USB Interface)

In boot mode (USB interface), the flash memory can be programmed or erased by sending control commands and program data from the host. An on-chip USB is used for communications between the host and this MCU. The host requires tools for sending control commands and data for programming. Figure 55.36 shows the configuration of a system for use in boot mode (USB interface).

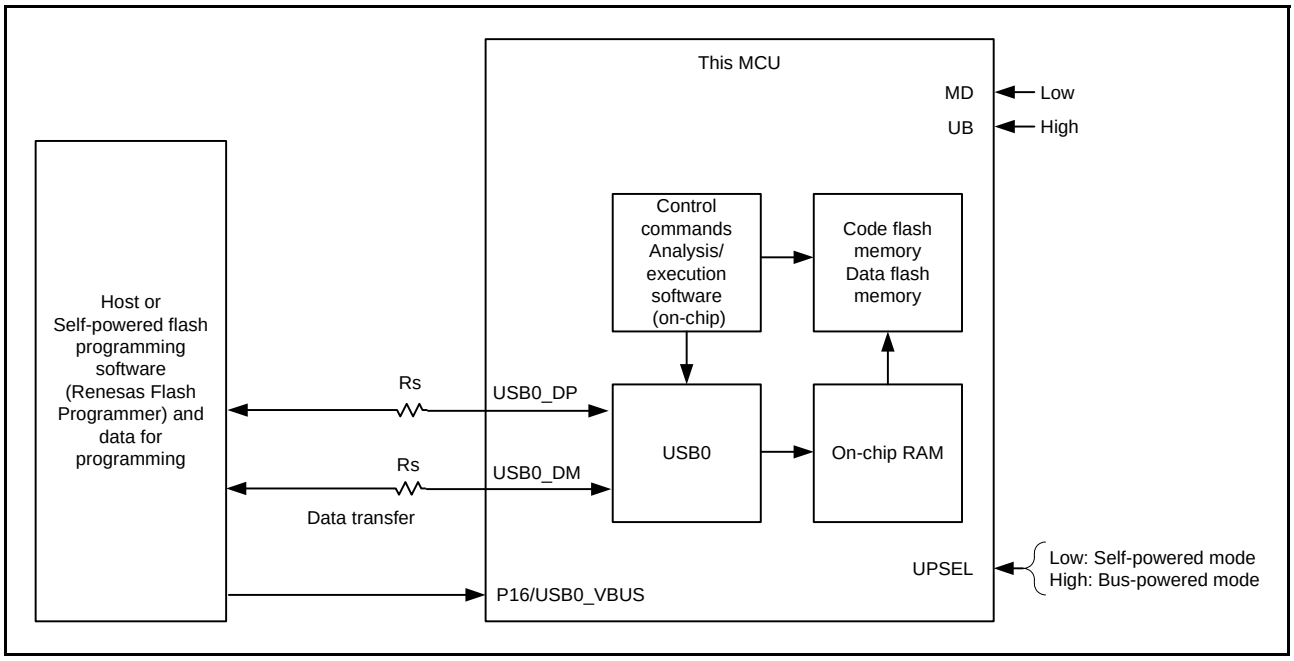


Figure 55.36 System Configuration in Boot Mode (USB Interface)

In boot mode (USB interface), self-powered mode or bus-powered mode is selected for operation. Release from the reset state while the low level is being applied to the MD and UPSEL pins and the high level is being applied to the UB pin selects self-powered mode. Release from the reset state while the low is being applied to the MD pin and the high level is being applied to the UB and UPSEL pins are high selects bus-powered mode.
Table 55.18 shows the enumeration information for each mode, and Table 55.19 shows the device class information.

Table 55.18 Enumeration Information

USB Specification	Ver.2.0 (Full-Speed)	
Maximum current	Self-powered mode (pins P35 and UPSEL = Low)	100 mA
	Bus-powered mode (pins P35 and UPSEL = High)	500 mA

Table 55.19 Device Class Information

Device Class	Communication Device Class (CDC)
Sub class	Abstract Control Model (ACM)

55.11.3 Boot Mode (FINE Interface)

The flash memory can be programmed and erased using the FINE in boot mode (FINE interface). The flash memory can be rewritten.

55.11.3.1 Operating Conditions in Boot Mode (FINE Interface)

FINE is used to communicate with the serial programmer in boot mode (FINE Interface).
Figure 55.37 shows an Example of Pin Connections in Boot Mode (FINE Interface). Table 55.20 lists Pin Handling in Boot Mode (FINE Interface).
The example of pin connections shown in Figure 55.37 is a simplified circuit. Operations are not guaranteed in all systems.

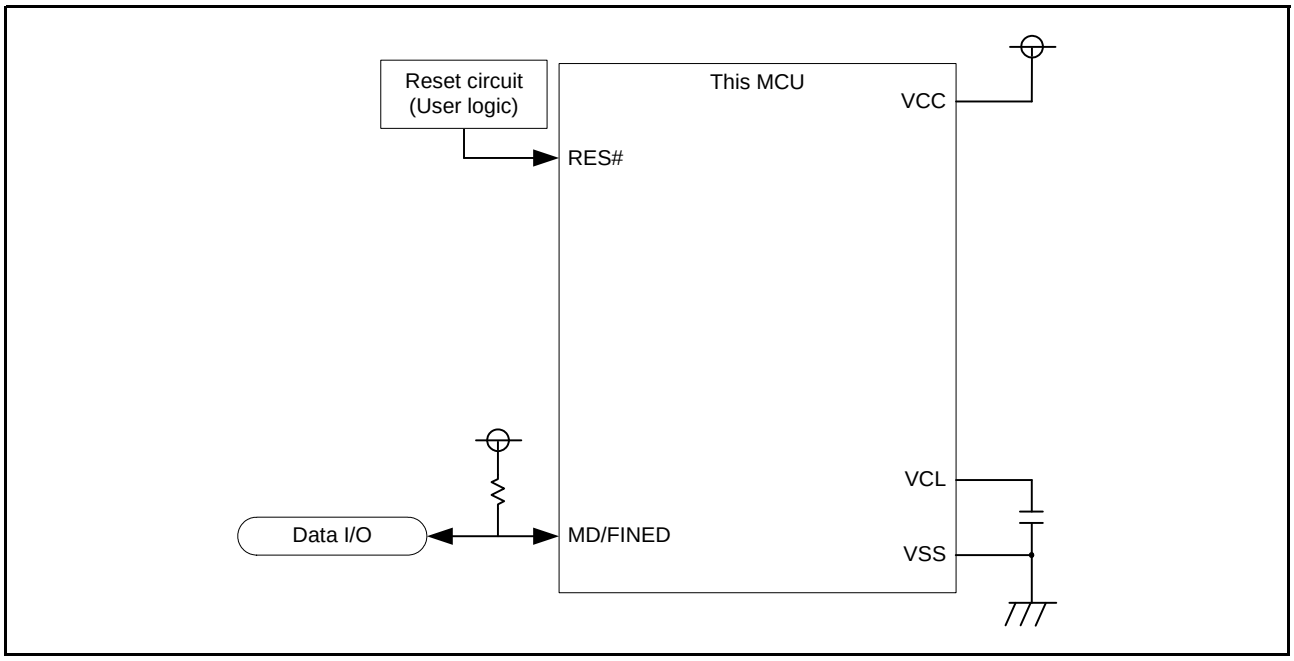


Figure 55.37 Example of Pin Connections in Boot Mode (FINE Interface)

Table 55.20 Pin Handling in Boot Mode (FINE Interface)

Pin Name	Name	I/O	Function
VCC, VSS	Power supply input	Input	Input 2.7 V or higher to the VCC pin. Input 0 V to the VSS pin.
VCL	Decoupling capacitor connect pin	—	Connect to the VSS pin via a 0.22-µF multilayer ceramic capacitor for stabilizing the internal voltage.
MD	Operating mode control/ data I/O	I/O	Connect the VCC pin via a resistor (pull up).
RES#	Reset input	Input	Reset pin. Connect to the reset circuit.

55.12 On-Chip Debugger ID Code Protection

This function is used to prohibit connection with the on-chip debugger (OCD). When an OCD is connected, the ID code of the OSIS register, which is written on the option-setting memory, is used for judging the ID codes.
The ID code sent from the OCD is judged if it matches the ID code in the OSIS register. If the codes match, connection with the OCD proceeds. If they do not match, the OCD cannot be connected.

55.13 Serial Programmer ID Code Protection

This function is used to prohibit connection with the serial programmer. When a serial programmer is connected, the control and ID codes of the OSIS register, which are written on the option-setting memory, are used for judgment of the ID codes.
Control or ID codes sent from serial programmers are checked to see if they match the control or ID code in the OSIS register. If the codes match, connection to the serial programmer proceeds. If they do not match, connection with the serial programmer does not proceed. If the control code is 45h and the codes do not match three times in a row, all data in the flash memory are erased.*1

Note 1. When the FAW.FSPR bit is 0, data in the flash memory are not erased.

55.14 ROM Code Protection

ROM code protection is a function to prohibit reading from or programming/erasure to the flash memory when the parallel programmer is used.
For details, see section 7.2.10, ROM Code Protection Register (ROMCODE).

55.15 Boot Mode Communications Protocol

This section describes the communications protocol for use in boot mode. When developing a serial programmer, use this communications protocol to control it.

55.15.1 How to Start the Chip Up in Boot Mode

(1) How to Start the Chip Up in Boot Mode (SCI Interface)

The chip starts up in boot mode (SCI interface) if both the MD and UB pins are at the low level on release from the reset state (i.e. when the level on the RES# pin changes from low to high). A waiting time of at least 400 ms is required while the RES# pin is held at the high level after the chip starts up in boot mode (SCI interface) until communications with the MCU can proceed.

Figure 55.38 shows the states of pins up to communications in boot mode (SCI interface) becoming possible.

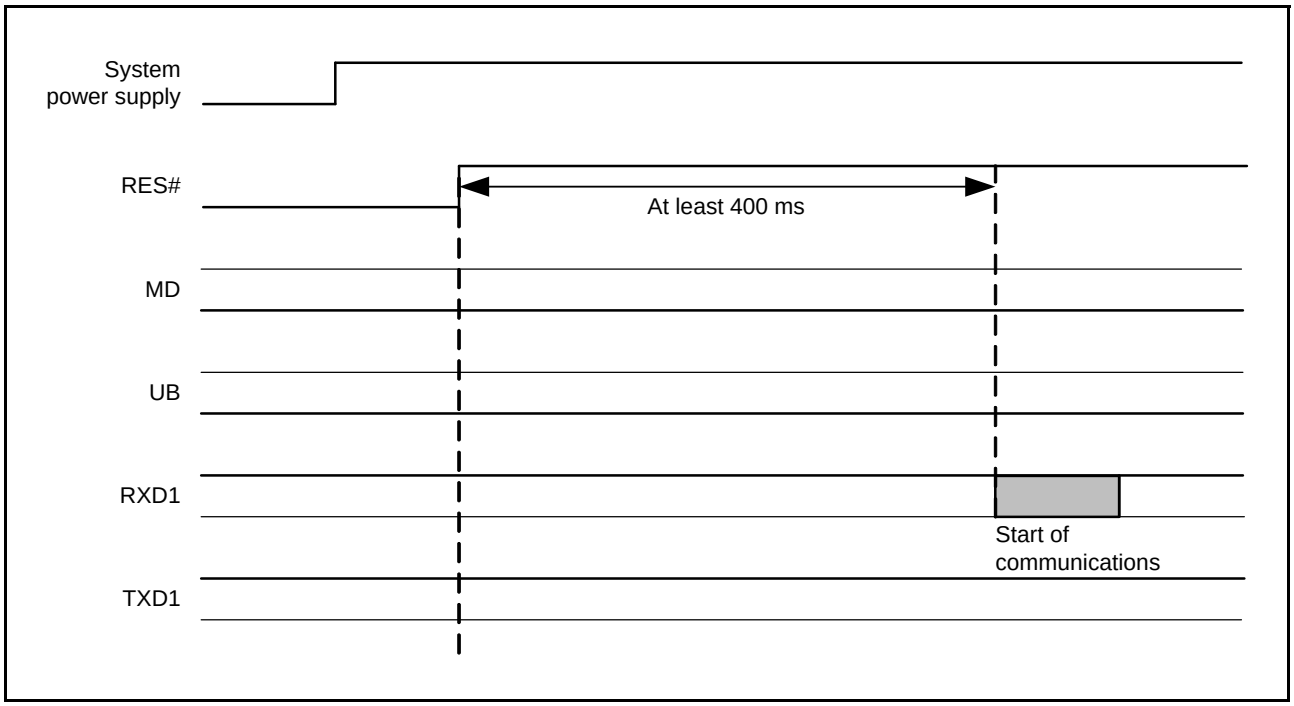


Figure 55.38 States of Pins Up to Communications in Boot Mode (SCI Interface) Becoming Possible

(2) How to Start the Chip Up in Boot Mode (USB Interface)

The chip starts up in boot mode (USB interface) if the MD pin is at the low level and UB pin is at the high level on release from the reset state (i.e. when the level on the RES# pin changes from low to high).

55.15.2 State Transitions in Boot Mode

55.15.2.1 State Transitions in Boot Mode (SCI Interface)

Figure 55.39 shows a flowchart for transition to boot mode (SCI interface).

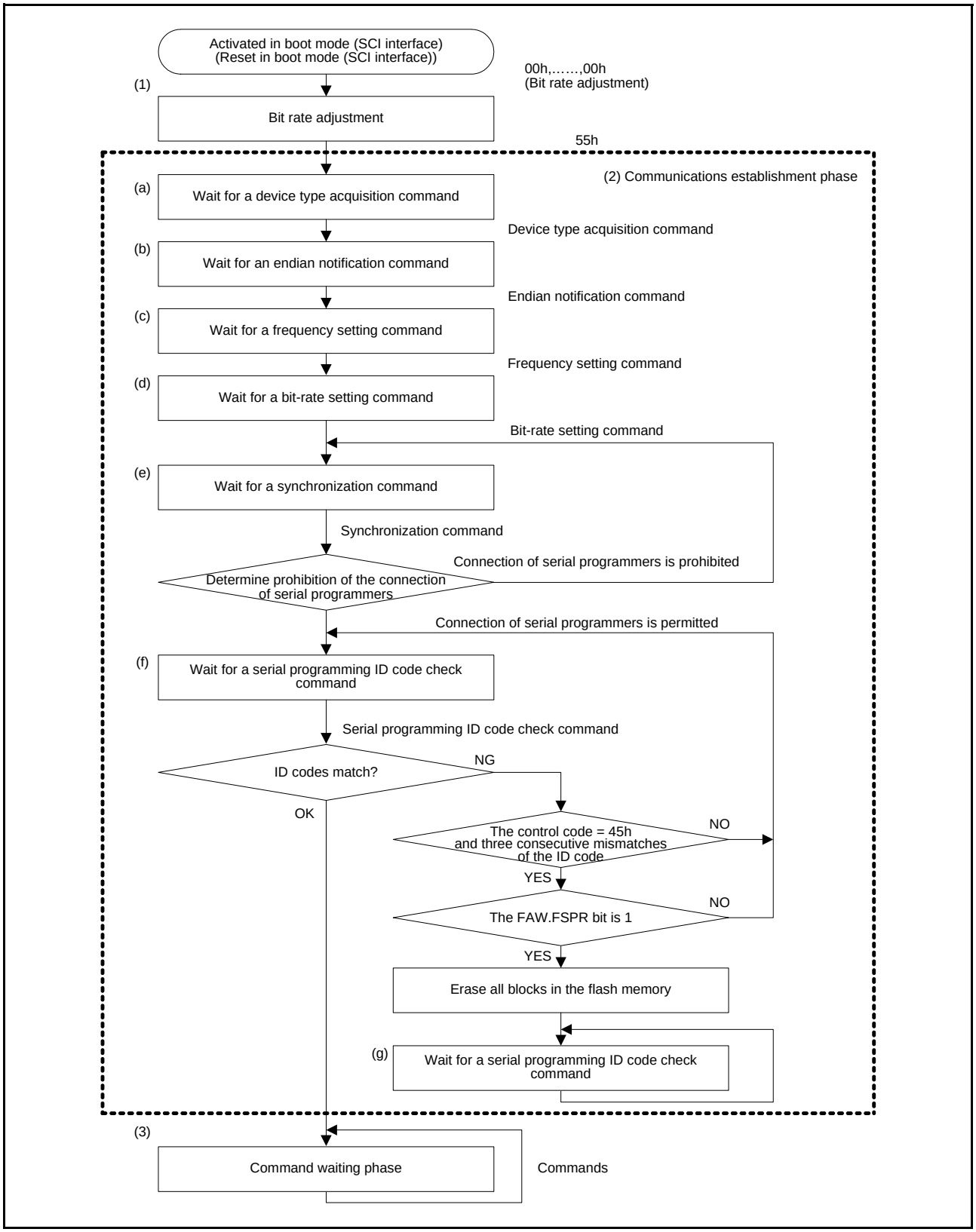


Figure 55.39 Flowchart for Transition to Boot Mode (SCI Interface)

(1) Matching the bit rates

When this MCU is activated in boot mode, the bit rate of the SCI is automatically adjusted to match that of the host. On completion of this automatic bit rate adjustment, this MCU transmits the value 00h to the host. On subsequent correct reception of the value 55h sent from the host, this MCU enters the communications establishment phase. For details on matching of the bit rates, see section 55.15.3, Automatic Adjustment of the Bit Rate.

(2) Communications establishment phase

The device, endian, frequency, and bit rate are selected in this phase. In this phase, the serial programmer ID code protection is judged. For the commands to use in the communications establishment phase, see section 55.15.5, Communications Establishment Phase.

(a) Waiting for a device type acquisition command

In this state, the MCU is waiting for a device type acquisition command to be sent from the host. When it receives a device type acquisition command, the state shifts to waiting for an endian notification command. For details of the device type acquisition command, see section 55.15.9, Device Type Acquisition Command.

(b) Waiting for an endian notification command

In this state, the MCU is waiting for an endian notification command to be sent from the host. When it receives an endian notification command, the state shifts to waiting for a frequency setting command. For details of the endian notification command, see section 55.15.10, Endian Notification Command.

(c) Waiting for a frequency setting command

In this state, the MCU is waiting for a frequency setting command to be sent from the host. When it receives a frequency setting command, the state shifts to waiting for a bit-rate setting command. For details of the frequency setting command, see section 55.15.11, Frequency Setting Command.

(d) Waiting for a bit-rate setting command

In this state, the MCU is waiting for a bit-rate setting command to be sent from the host. When it receives a bit-rate setting command, the state shifts to waiting for a synchronization command. For details of the bit-rate setting command, see section 55.15.12, Bit-Rate Setting Command.

(e) Waiting for a synchronization command

In this state, the MCU is waiting for a synchronization command to be sent from the host. When it receives a synchronization command, it transitions to the waiting state of the serial programming ID code check command. If the MCU has been set to prohibit the connection of a serial programmer, this MCU transmits an error code to indicate that connecting a serial programmer is prohibited and remains in the state of waiting for a synchronization command. For details of the synchronization command, see section 55.15.13, Synchronization Command.

(f) Waiting for a serial programming ID code check command

Wait for the serial programming ID code check command to be sent from the host. The control code or ID code that was sent is compared with the control code or ID code in the OSIS register. The device enters the command waiting phase if they match. If they do not match, the next transition is back to the state of waiting for a serial programming ID code check command.
If the control code is 45h and the codes do not match three times in a row, all data in the flash memory are erased.*1
For details of the ID code check command, see section 55.15.15, Serial Programming ID Code Check Command.

Note 1. When the FAW.FSPR bit is 0, the blocks in those areas are not erased.

(g) Waiting for a serial programming ID code check command (after erasure)

After all blocks in the flash memory are erased, reboot the MCU in boot mode.

(3) Phase of waiting for commands

In this state, programming and erasure proceed in accord with commands from the host. For details of the commands that can be issued in the command waiting phase, see section 55.15.6, Command Waiting Phase.

55.15.2.2 State Transitions in Boot Mode (USB Interface)

Figure 55.40 shows a flowchart for transition to boot mode (USB interface).

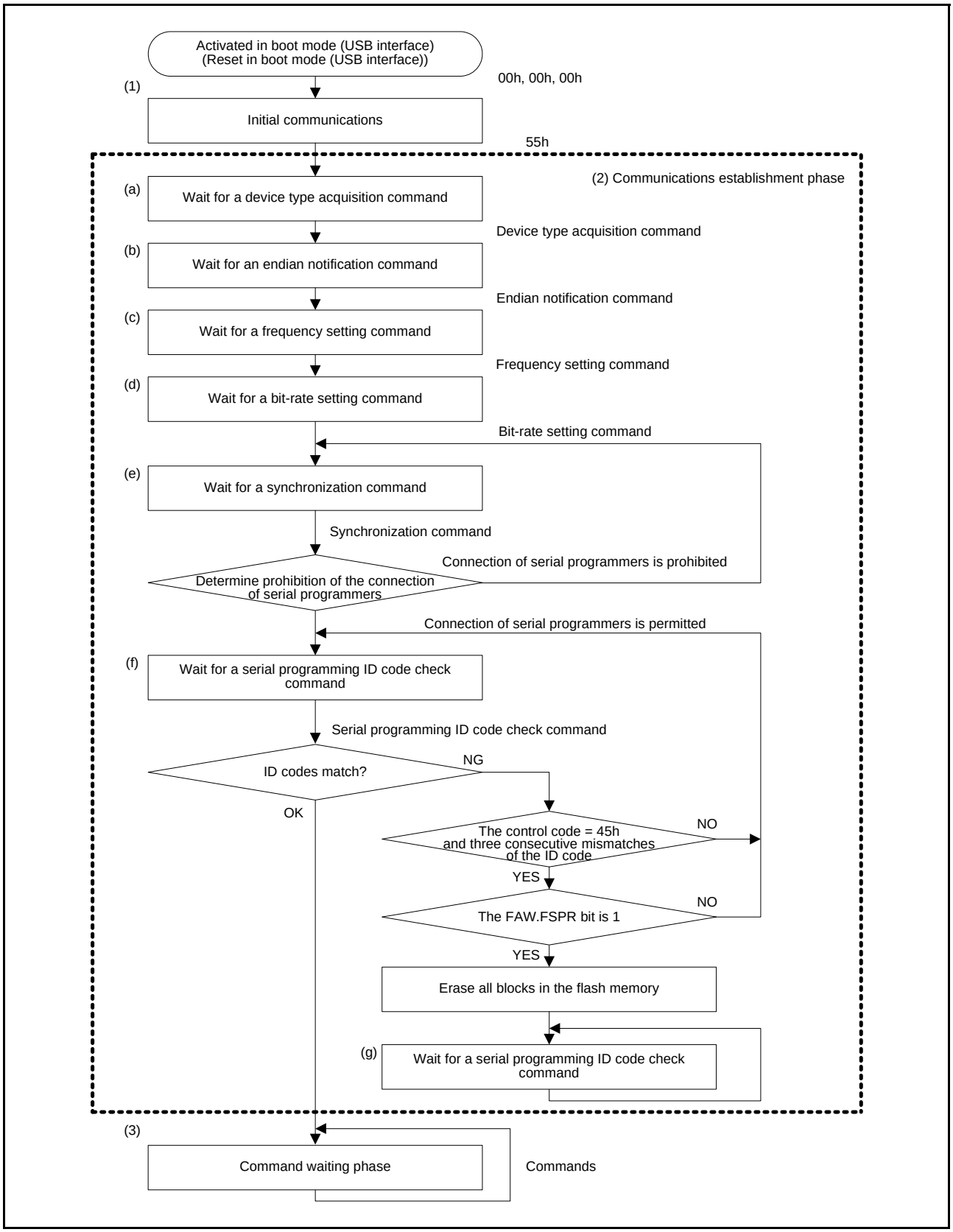


Figure 55.40 Flowchart for Transition to Boot Mode (USB Interface)

(1) Initial Communications

Enumeration starts when this MCU is activated in boot mode (USB interface).
Once the process of enumeration between the host and this MCU is completed, send the value 00h from the host to this MCU three times.
This MCU ignores transmission of the value 00h from the host from the fourth time and onward.
This MCU returns 00h to the host when receiving the value 00h three times. On subsequent correct reception of the value 55h sent from the host, this MCU enters the communications establishment phase.

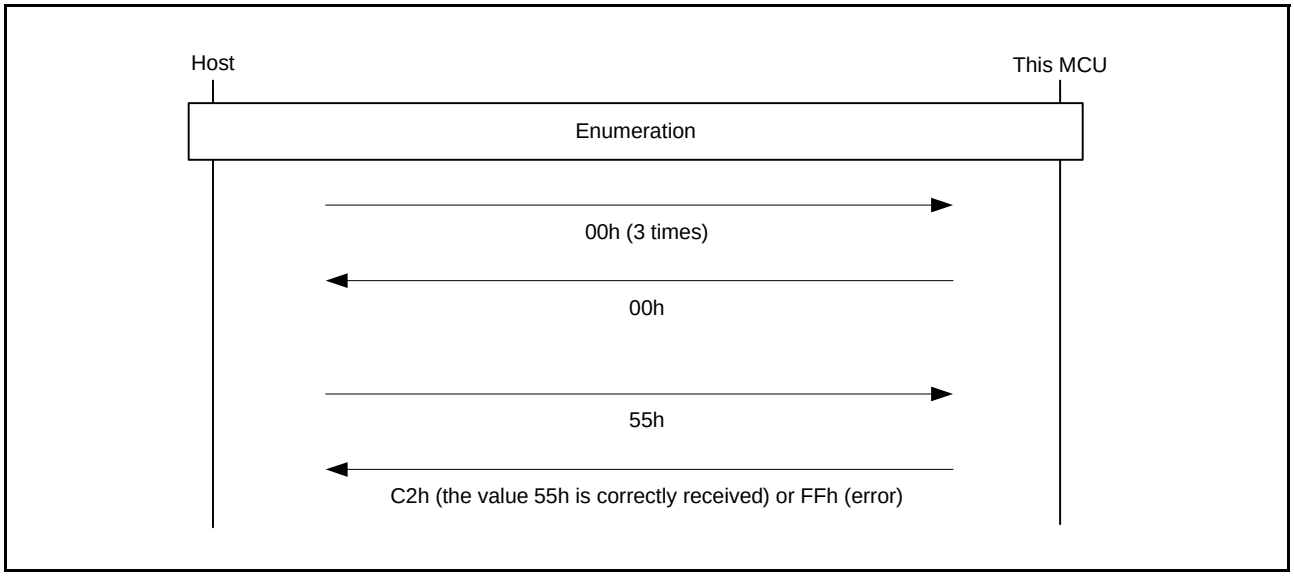


Figure 55.41 Initial Communications Between the Host and This MCU

(2) Communications establishment phase

The device, endian, frequency, and bit rate are selected in this phase. In this phase, the serial programmer ID code protection is judged. For the commands to use in the communications establishment phase, see section 55.15.5, Communications Establishment Phase.

(a) Waiting for a device type acquisition command

In this state, the MCU is waiting for a device type acquisition command to be sent from the host. When it receives a device type acquisition command, the state shifts to waiting for an endian notification command. For details of the device type acquisition command, see section 55.15.9, Device Type Acquisition Command.

(b) Waiting for an endian notification command

In this state, the MCU is waiting for an endian notification command to be sent from the host. When it receives an endian notification command, the state shifts to waiting for a frequency setting command. For details of the endian notification command, see section 55.15.10, Endian Notification Command.

(c) Waiting for a frequency setting command

In this state, the MCU is waiting for a frequency setting command to be sent from the host. When it receives a frequency setting command, the state shifts to waiting for a bit-rate setting command. For details of the frequency setting command, see section 55.15.11, Frequency Setting Command.

(d) Waiting for a bit-rate setting command

In this state, the MCU is waiting for a bit-rate setting command to be sent from the host. When it receives a bit-rate setting command, the state shifts to waiting for a synchronization command. For details of the bit-rate setting command, see section 55.15.12, Bit-Rate Setting Command.

(e) Waiting for a synchronization command

In this state, the MCU is waiting for a synchronization command to be sent from the host. When it receives a synchronization command, it transitions to the waiting state of the serial programming ID code check command. If the MCU has been set to prohibit the connection of a serial programmer, this MCU transmits an error code to indicate that connecting a serial programmer is prohibited and remains in the state of waiting for a synchronization command. For details of the synchronization command, see section 55.15.13, Synchronization Command.

(f) Waiting for a serial programming ID code check command

In this state, the MCU is waiting for a serial programming ID code check command to be sent from the host. The control code or ID code that was sent is compared with the control code or ID code in the OSIS register. The device enters the command waiting phase if they match. If they do not match, the next transition is back to the state of waiting for a serial programming ID code check command.
If the control code is 45h and the codes do not match three times in a row, all data in the flash memory are erased.*1
For details of the ID code check command, see section 55.15.15, Serial Programming ID Code Check Command.

Note 1. When the FAW.FSPR bit is 0, the blocks in those areas are not erased.

(g) Waiting for a serial programming ID code check command (after erasure)

After all blocks in the flash memory are erased, reboot the MCU in boot mode.

(3) Phase of waiting for commands

In this state, programming and erasure proceed in accord with commands from the host. For details of the commands that can be issued in the command waiting phase, see section 55.15.6, Command Waiting Phase.

55.15.3 Automatic Adjustment of the Bit Rate

When this MCU is booted up in boot mode (SCI interface), asynchronous transfer by the SCI is used to measure the periods at low level of consecutive bytes with value 00h that are sent from the host. While the period at low level is being measured, set the host's SCI transfer format to 8-bit data, one stop bit, no parity, and a transfer rate of 9,600 bps. This MCU measures the periods at low level in the signal from the host, adjusts the bit rate of its SCI, and then sends the value 00h to the host.

If reception of the value 00h by the host is successful, the host responds by sending the value 55h to this MCU. If successful reception of 00h by the host is not possible, reboot this MCU in boot mode, and then repeat the process of automatically adjusting the bit rate. If reception of the value 55h by this MCU is successful, it responds by sending C2h to the host, and if successful reception of 55h by this MCU is not possible, it responds by sending FFh to the host.

If this MCU is started up in boot mode (USB interface), the bit rate is not automatically adjusted.

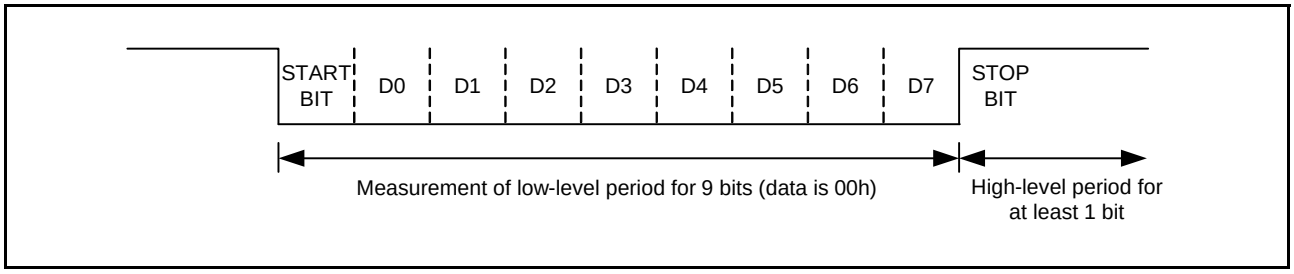


Figure 55.42 Transfer Format Used by SCI in Automatic Adjustment of Bit Rate

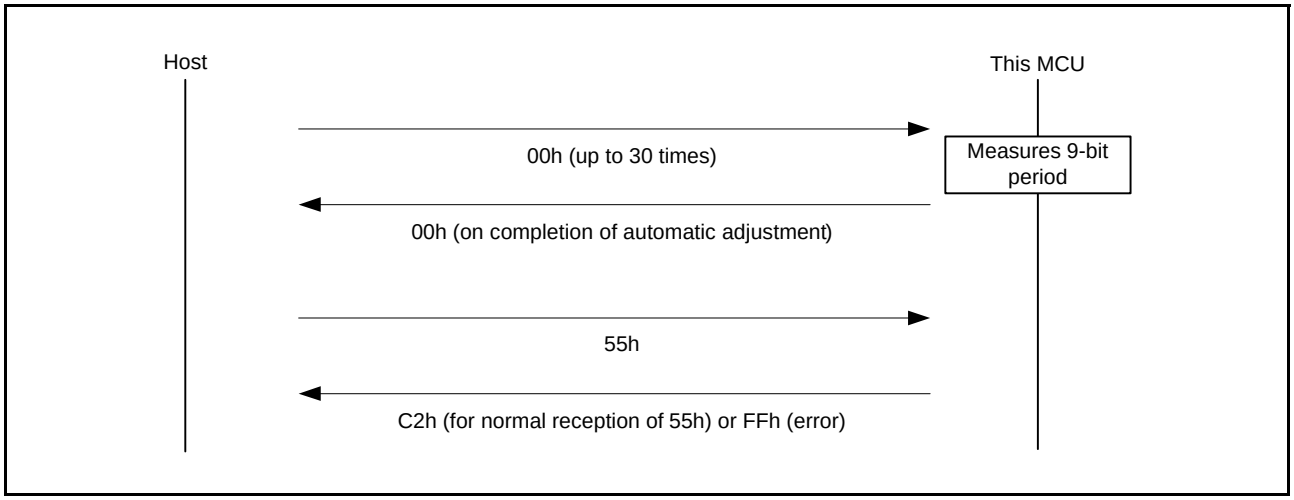


Figure 55.43 Sequence of Transfer between Host and This MCU

For the host's SCI bit rate, ensure that SCI communications proceed under the conditions given in Table 55.21.

Table 55.21 Conditions for Automatic Bit-Rate Adjustment

Bit Rate of SCI in Host
9,600 bps

55.15.4 Packet Format

(1) Command packet

The host sends commands to this MCU in the format below.

SOH	LNH	LNL	COM	Command information (variable length) (up to 255 bytes)	SUM	ETX

Symbol	Code	Description
SOH	01h	Start of a packet (1 byte)
LNH	—	Packet length (length of COM and the command information) (8 to 15 bits) (1 byte)
LNL	—	Packet length (length of COM and the command information) (0 to 7 bits) (1 byte)
COM	—	Command code (1 byte)
Command information	—	Command information (up to 255 bytes)
SUM*1	—	Two's complement of the sum of values of LNH, LNL, COM, and the command information (1 byte)
ETX	03h	End of a packet (1 byte)

Note 1. SUM indicates the 1 byte of data that produces 00h as the result of adding LNH, LNL, COM, the command information, and SUM itself.

(2) Status packet and data packet

Data transmission proceeds between the host and this MCU in the format below.

SOD	LNH	LNL	RES	Data (variable length) (up to 1024 bytes)	SUM	ETB or ETX

Symbol	Code	Description
SOD	81h	Start of a packet (1 byte)
LNH	—	Packet length (length of RES and the data) (8 to 15 bits) (1 byte)
LNL	—	Packet length (length of RES and the data) (0 to 7 bits) (1 byte)
RES	—	Response code (1 byte)
Data	—	Data (up to 1024 bytes)
SUM*1	—	Two's complement of the sum of values of LNH, LNL, RES, and the data (1 byte)
ETB	17h	End of a packet (1 byte)
ETX	03h	End of the last packet (1 byte)

Note 1. SUM indicates the 1 byte of data that produces 00h as the result of adding LNH, LNL, RES, the data, and SUM itself.

55.15.5 Communications Establishment Phase

Table 55.22 lists the commands available in the commands establishment phase.
The synchronization command and ID authentication mode acquisition command can also be used in the command waiting phase.

Table 55.22 Commands Available in the Communications Establishment Phase

Command Name	Function
Device type acquisition	Transmits the oscillation frequency and CPU operating frequency (in Hz) supported by boot mode to the host.
Endian notification	Indicates whether big-endian or little-endian is to be used.
Frequency setting	Sets the values of the oscillation frequency and CPU operating frequency (in Hz).
Bit-rate setting	Changes the bit rate.
Synchronization	This command is used in processing for communications synchronization. It is also used when confirming whether the MCU can accept commands.
ID authentication mode acquisition	This command sends the enabled state of serial programmer ID code protection to the host.
Serial programming ID code check	Determines whether the control code and ID code written in the option-setting memory matches the control code and ID code sent by the host.

In the communications establishment phase, send commands from the host in the order of the device type acquisition, endian notification, frequency setting, bit-rate setting, and synchronization commands according to the responses to commands. Send the ID authentication mode acquisition or serial programming ID code check command following the synchronization command.
If commands are issued in an incorrect order or other commands are issued, this MCU returns a response indicating a flow error.

55.15.6 Command Waiting Phase

Table 55.23 lists the commands available in the command waiting phase.
The synchronization command and ID authentication mode acquisition command can also be used in the communications establishment phase.

Table 55.23 Commands Available in the Command Waiting Phase

Command Name	Function
Synchronization	See Table 55.22.
Blank check	Check that a selected area is blank.
Block erase	Erases a selected single block.
Area erase	Erases the specified area.
Programming	Programs the selected area.
Read	Reads data from a selected area.
ID authentication mode acquisition	See Table 55.22.
Simple addition checksum	Calculates the sum of values in a selected area.
Configuration clearing	Erases the set value of the option-setting memory and the TM target areas.
Configuration programming	Programs the set value in the option-setting memory.
Configuration reading	Reads the set value of the option-setting memory.
Acquisition of the number of the area information	Obtains the number of area information of the flash memory.
Acquisition of the area information	Obtains the area information of the flash memory.

If the host has sent an undefined command, this MCU returns a response indicating an error in the form of a non-supported command.

55.15.7 Command Transfer Sequence

Though the sequence of transfer differs from command to command, common transfer sequences are used for the commands that only make settings for this MCU and for the commands that obtain information on the settings in this MCU. However, as the contents of the command packet, status packet, and data packet differ for each command, see the sections on the individual commands for details.

(1) Common transfer sequence for the commands that only make settings

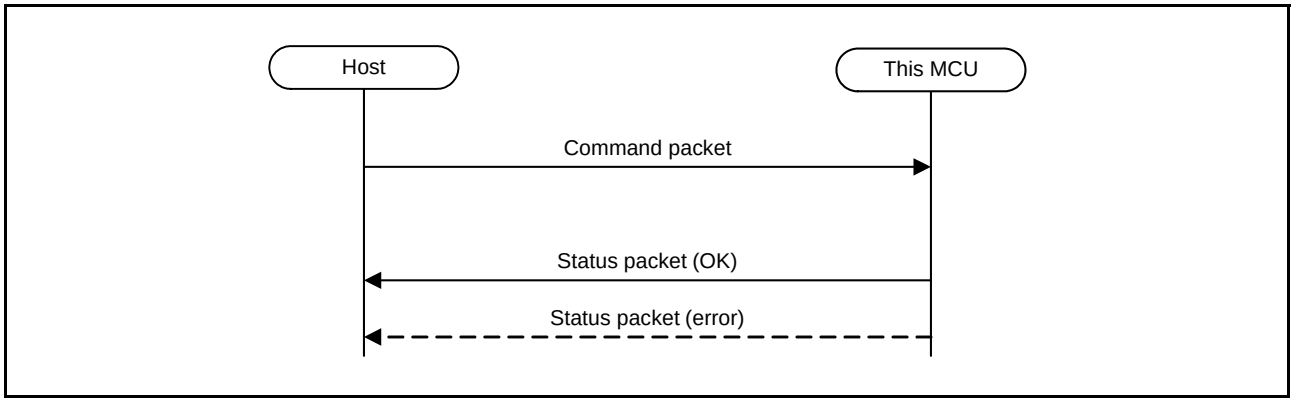


Figure 55.44 Common Transfer Sequence for the Commands that Only Make Settings

(2) Common transfer sequence for the commands that obtain information on settings

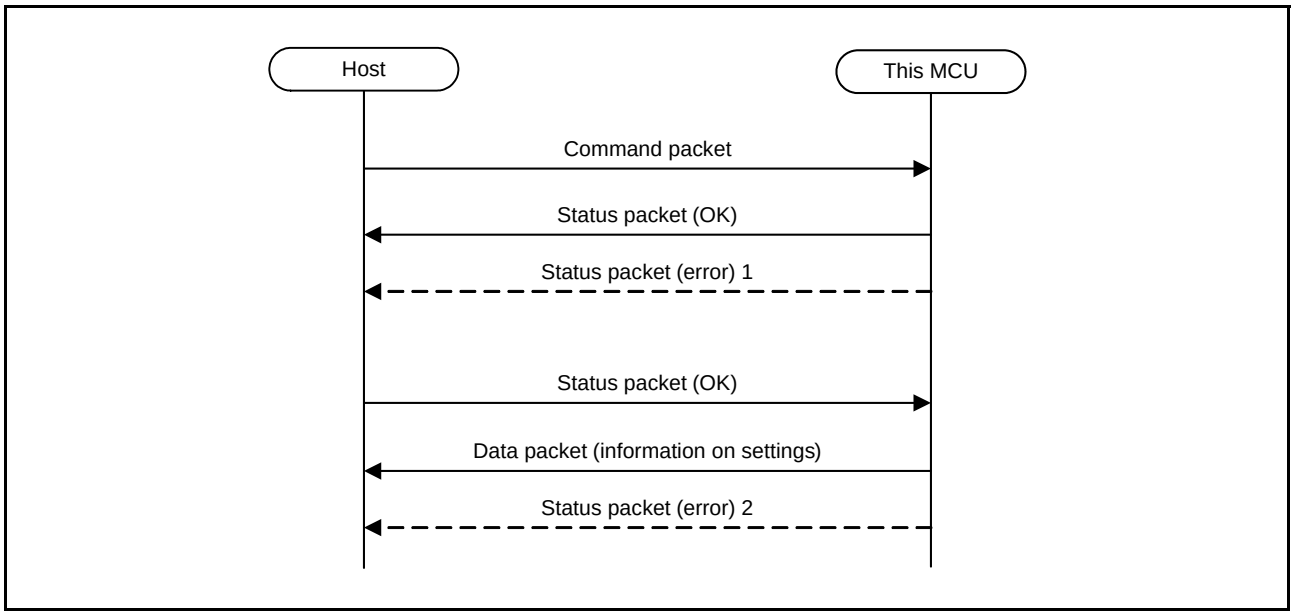


Figure 55.45 Common Transfer Sequence for the Commands that Obtain Information on Settings

Table 55.24 Common Transfer Sequence

Command Name	Common Transfer Sequence Type
Device type acquisition	Command that obtains information on settings
Endian notification	Command that only makes a setting
Frequency setting	Command that obtains information on settings
Bit rate setting	Not in a common transfer sequence
Synchronization	Command that only makes a setting
ID authentication mode acquisition	Command that obtains information on settings
Serial programming ID code check	Command that only makes a setting
Blank check	Command that only makes a setting
Block erase	Command that only makes a setting
Area erase	Command that only makes a setting
Programming	Not in a common transfer sequence
Read	Not in a common transfer sequence
Simple addition checksum	Command that obtains information on settings
Configuration clearing	Command that only makes a setting
Configuration programming	Not in a common transfer sequence
Configuration reading	Command that obtains information on settings
Acquisition of the number of the area information	Command that obtains information on settings
Acquisition of the area information	Command that obtains information on settings

For the command transfer sequences that are not in a common transfer sequence, see the sections on the individual commands.

55.15.8 Non-supported Commands

When this MCU receives an undefined command packet, it returns an unsupported error (C0h) and enters the command waiting state.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: Packet length (8 to 15 bits)
LNL: Packet length (0 to 7 bits)
COM: Command code*1
SUM: Sum of values
ETX: 03h

Note 1. Command code other than those specified in Table 55.23.

(2) Status packet structure

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 80h | COM (command code)
ERR: Error code
C0h (“non-supported” error)
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

55.15.9 Device Type Acquisition Command

This command is used to make the MCU send the input frequency and system clock frequency (in Hz) supported by boot mode (SCI interface).
This command can only be accepted in the communications establishment phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 38h
SUM: C7h
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	TYP	OSA	OSI	CPA	CPI	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 19h
RES: 38h (OK)
TYP: Type code (8 bytes)*1
OSA: Maximum input frequency (4 bytes)
OSI: Minimum input frequency (4 bytes)
CPA: Maximum system clock frequency (4 bytes)
CPI: Minimum system clock frequency (4 bytes)
SUM: Sum of values
ETX: 03h

An example of the values sent is given below.

Maximum input frequency = 16,000,000 Hz
- OSA (1st byte): 00h
- OSA (2nd byte): F4h
- OSA (3rd byte): 24h
- OSA (4th byte): 00h

Minimum input frequency = 16,000,000 Hz
- OSI (1st byte): 00h
- OSI (2nd byte): F4h
- OSI (3rd byte): 24h
- OSI (4th byte): 00h

Maximum system clock (ICLK) = 120,000,000 Hz
- CPA (1st byte): 07h
- CPA (2nd byte): 27h
- CPA (3rd byte): 0Eh
- CPA (4th byte): 00h

Minimum system clock: (ICLK) = 120,000,000 Hz
- CPI (1st byte): 07h
- CPI (2nd byte): 27h
- CPI (3rd byte): 0Eh
- CPI (4th byte): 00h

In boot mode (USB interface), transfer is at the input frequency when the value of OSA or OSI is in use (20 MHz or 24 MHz).

Note 1. Reserved data

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 38h (OK)
SUM: C7h
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: B8h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: B8h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

55.15.10 Endian Notification Command

This command is used to inform the MCU of the endian (big or little).
Specify either endian as the endian information according to the data to be programmed.
This command can only be accepted in the communications establishment phase.

(1) Command packet structure

SOH	LNH	LNL	COM	END	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 02h
COM: 36h
END: Endian information
00h (big-endian)
01h (little-endian)
SUM: Sum of values
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 36h (OK)
SUM: C9h
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: B6h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D7h (endian error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

55.15.11 Frequency Setting Command

This command is used to set the values of oscillation frequency and CPU operating frequency (in Hz).
In boot mode (SCI or FINE interface), the HOCO runs at 16 MHz and the ICLK runs at 120 MHz, so set the input frequency to 16 MHz and the system clock frequency to 120 MHz. Additionally, in boot mode (SCI or FINE interface), the FCLK and PCLKB run at 60 MHz, so the MCU returns the hexadecimal value corresponding to 60,000,000 to indicate the frequency of the peripheral module clock.
In boot mode (USB interface), the input frequency in use is 20 MHz or 24 MHz and the ICLK runs at 120 MHz, so set the input frequency to 20 MHz or 24 MHz and the system clock frequency to 120 MHz. Additionally, in boot mode (USB interface), the FCLK runs at 60 MHz and the UCLK runs at 48 MHz, so the MCU returns the hexadecimal value corresponding to 48,000,000 to indicate the frequency of the peripheral module clock.
This command can only be accepted in the communications establishment phase.

(1) Command packet structure

SOH	LNH	LNL	COM	OC1	OC2	OC3	OC4	CC1	CC2	CC3	CC4	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 09h
COM: 32h
OC1: Input frequency
OC2: Input frequency
OC3: Input frequency
OC4: Input frequency
CC1: System clock frequency
CC2: System clock frequency
CC3: System clock frequency
CC4: System clock frequency
SUM: Sum of values
ETX: 03h

Ex. When the input frequency is 16,000,000 Hz and the system clock frequency is 120,000,000 Hz, send the values as below.

OC1: 00h	CC1: 07h
OC2: F4h	CC2: 27h
OC3: 24h	CC3: 0Eh
OC4: 00h	CC4: 00h

(2) Data packet structure

SOD	LNH	LNL	RES	FQ1	FQ2	FQ3	FQ4	PF1	PF2	PF3	PF4	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 09h
RES: 32h
FQ1: System clock frequency
FQ2: System clock frequency
FQ3: System clock frequency
FQ4: System clock frequency
PF1: Peripheral clock frequency
PF2: Peripheral clock frequency
PF3: Peripheral clock frequency
PF4: Peripheral clock frequency
SUM: Sum of values
ETX: 03h

Ex. An example of the values sent is given below.
System clock frequency = 120,000,000 Hz
Peripheral clock frequency = 60,000,000 Hz

FQ1: 07h	PF1: 03h
FQ2: 27h	PF2: 93h
FQ3: 0Eh	PF3: 87h
FQ4: 00h	PF4: 00h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 32h (OK)
SUM: CDh
ETX: 03h

(4) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: B2h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D1h (input frequency error)
D2h (system clock (ICLK) frequency error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

55.15.12 Bit-Rate Setting Command

This command is used to change the bit rate after receiving bit rate data (in bps).
If an error occurs, the bit rate is not changed.
This command can only be accepted in the communications establishment phase.
In boot mode (USB or FINE interface), the bit rate is not changed. The bit rate should be any desired value.

(1) Procedure

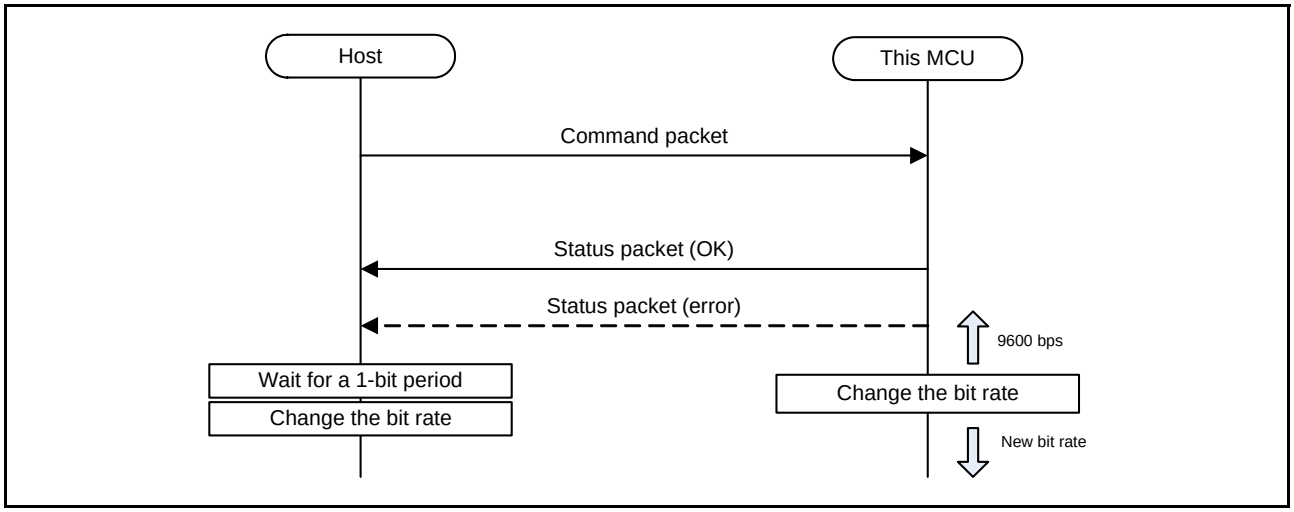


Figure 55.46 Bit-Rate Setting Command Transfer Sequence

(2) Command packet structure

SOH	LNH	LNL	COM	BR1	BR2	BR3	BR4	SUM	ETX

When the bit rate is 2,000,000 bps, send the values as below.
BR1: 00h
BR2: 1Eh
BR3: 84h
BR4: 80h

SOH: 01h
LNH: 00h
LNL: 05h
COM: 34h
BR1: Bit rate
BR2: Bit rate
BR3: Bit rate
BR4: Bit rate
SUM: Sum of values
ETX: 03h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 34h (OK)
SUM: CBh
ETX: 03h

(4) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: B4h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D4h (bit rate error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

55.15.13 Synchronization Command

This command is used in processing to synchronize communications.
It is also used when checking whether the MCU is ready to accept commands. If a serial programmer connection is prohibited, a serial programmer connection prohibition error is returned.
This command can be accepted in both the communications establishment and command waiting phases.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 00h
SUM: FFh
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 00h (OK)
SUM: FFh
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 80h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
DCh (serial programmer connection prohibition error)
SUM: Sum of values
ETX: 03h

55.15.14 ID Authentication Mode Acquisition Command

This command sends the enabled state of serial programmer ID code protection to the host.
This command can be accepted in both the communications establishment and command waiting phases.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM: 2Ch
SUM: D3h
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	MOD	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 2Ch (OK)
MOD: ID authentication information (1 byte)
00h (serial programmer ID code protection is enabled)
SUM: Sum of values
ETX: 03h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 2Ch (OK)
SUM: D3h
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: ACh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: ACh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

55.15.15 Serial Programming ID Code Check Command

This command sends the result when the control and ID codes set in the OSIS register match those received from the host.

This command can be accepted in the communications establishment phase. When the serial programmer ID code protection is enabled, the MCU does not enter the command waiting phase unless processing in response to this command ends normally.

If the control code is 45h, the codes do not match three times in a row, all data in the flash memory are erased.*1

Note 1. When the FAW.FSPR bit is 0, data in the flash memory are not erased.

(1) Command packet structure

SOH	LNH	LNL	COM	IDC	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 11h
COM: 30h
IDC: Control code and ID code (16 bytes)*1
SUM: Sum of values
ETX: 03h

Note 1. Send the values as below.
<ID code>
ID = 128'h0F0E0D0C0B0A09080706050403020100
(Control code: 00h, ID code 2: 01h, ID code 3: 02h, ... , ID code 16: 0Fh)
<Data for transmission>

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte	…
00h	01h	02h	03h	04h	05h	06h	07h	…

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 30h (OK)
SUM: CFh
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: B0h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
DBh (ID code mismatch error)
E1h (Error of erasure)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

55.15.16 Blank Check Command

This command checks whether the specified area is blank.
Specify addresses aligned with 128-byte boundaries for the code flash memory and with 4-byte boundaries for the data flash memory. The option-setting memory cannot be specified. When trusted memory (TM) is enabled, an error occurs if the area for blank-checking include any part of the area being handled as TM.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SHH	SHL	SLH	SLL	EHH	EHL	ELH	ELL	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 09h
COM: 10h
SHH: Blank check start address (24 to 31 bits)
SHL: Blank check start address (16 to 23 bits)
SLH: Blank check start address (8 to 15 bits)
SLL: Blank check start address (0 to 7 bits)
EHH: Blank check end address (24 to 31 bits)
EHL: Blank check end address (16 to 23 bits)
ELH: Blank check end address (8 to 15 bits)
ELL: Blank check end address (0 to 7 bits)
SUM: Sum of values
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 10h (OK)
SUM: EFh
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 90h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D0h (address error)
E0h (non-blank error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

55.15.17 Block Erase Command

This command is used to erase a specified single block.
Specify the block for erasure as the first address in the block.
When the TM function is enabled, blocks targeted for TM target area cannot be erased. The option-setting memory cannot be erased.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SHH	SHL	SLH	SLL	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 05h
COM: 12h
SHH: First address in the block for erasure (24 to 31 bits)
SHL: First address in the block for erasure (16 to 23 bits)
SLH: First address in the block for erasure (8 to 15 bits)
SLL: First address in the block for erasure (0 to 7 bits)
SUM: Sum of values
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 12h (OK)
SUM: EDh
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 92h (error)
ERR: Error code
- C1h (packet error)
- C2h (checksum error)
- C3h (flow error)
- D0h (address error)
- DAh (protection error)
- E1h (erase error)
- E7h (flash sequencer error)

SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

55.15.18 Area Erase Command

The area erase command erases the specified area successively, block by block, in ascending order of addresses.
The target area can be selected from the code flash memory or data flash memory.
When the TM function is enabled, blocks other than those being handled as TM are erased.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	ARE	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 02h
COM: 50h
ARE: Area
00h (code flash memory)
20h (data flash memory)
SUM: Sum of values
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 50h (OK)
SUM: AFh
ETX: 03h

(3) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	SUM	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: D0h (Error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D5h (area error)
DAh (protection error)
E1h (erase error)
E7h (flash sequencer error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

55.15.19 Programming Command

This command sets up the MCU for data to be programmed to its flash memory and specifies the area where the data are to be programmed.
Specify the data length in 128-byte units for the code flash memory and 4-byte units for the data flash memory. In addition, the program start address must be specified in 128-byte aligned for the code flash memory and 4-byte aligned for data flash memory. When the TM function is enabled, areas including the TM target area cannot be programmed.
This command can only be accepted in the command waiting phase.

(1) Procedure

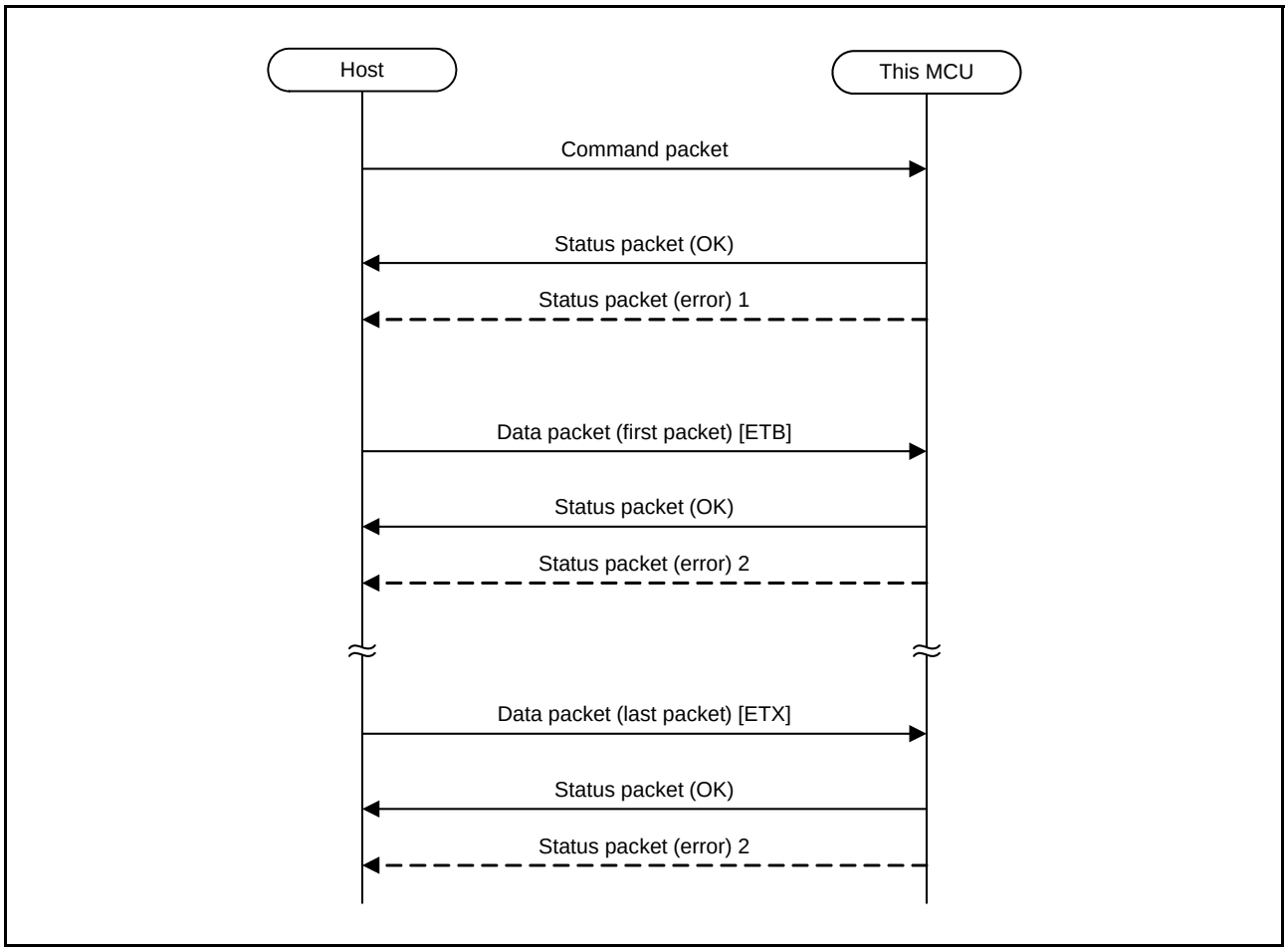


Figure 55.47 Programming Command Transfer Sequence

(2) Command packet structure

SOH	LNH	LNL	COM	SHH	SHL	SLH	SLL	EHH	EHL	ELH	ELL	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 09h
COM: 13h
SHH: Program start address (24 to 31 bits)
SHL: Program start address (16 to 23 bits)
SLH: Program start address (8 to 15 bits)
SLL: Program start address (0 to 7 bits)
EHH: Program end address (24 to 31 bits)
EHL: Program end address (16 to 23 bits)
ELH: Program end address (8 to 15 bits)
ELL: Program end address (0 to 7 bits)
SUM: Sum of values
ETX: 03h

(3) Data packet structure

SOD	LNH	LNL	RES	Data	SUM	ETB or ETX

SOD: 81h
LNH: Data length + 1 (8 to 15 bits)
LNL: Data length + 1 (0 to 7 bits)
RES: 13h (OK)
Data: Program data
SUM: Sum of values
ETB: 17h
ETX: 03h

(4) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 13h (OK)
SUM: ECh
ETX: 03h

(5) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 93h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D0h (address error)
DAh (protection error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(6) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 93h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
E2h (program error)
E7h (flash sequencer error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

55.15.20 Read Command

This command is used to read data from the specified area in the flash memory and send it to the host.
The minimum unit for reading is 1 byte. When the TM target area is read while the TM function is enabled, 0 is read. The option-setting memory cannot be specified.
This command can only be accepted in the command waiting phase.

(1) Procedure

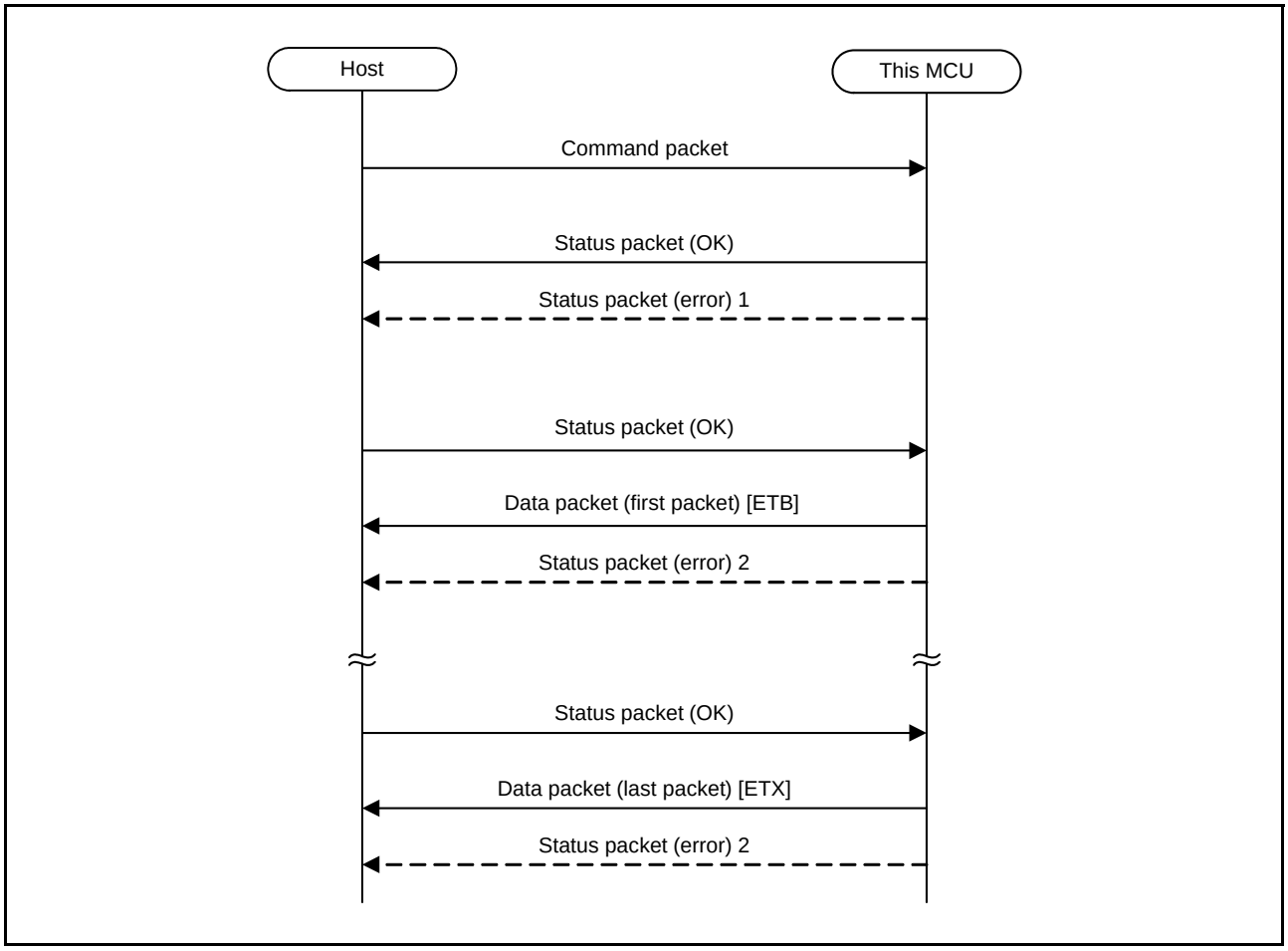


Figure 55.48 Read Command Transfer Sequence

(2) Command packet structure

SOH	LNH	LNL	COM	SHH	SHL	SLH	SLL	EHH	EHL	ELH	ELL	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 09h
COM: 15h
SHH: Read start address (24 to 31 bits)
SHL: Read start address (16 to 23 bits)
SLH: Read start address (8 to 15 bits)
SLL: Read start address (0 to 7 bits)
EHH: Read end address (24 to 31 bits)
EHL: Read end address (16 to 23 bits)
ELH: Read end address (8 to 15 bits)
ELL: Read end address (0 to 7 bits)
SUM: Sum of values
ETX: 03h

(3) Data packet structure

SOD	LNH	LNL	RES	Data	SUM	ETB	or	ETX

SOD: 81h
LNH: Data length + 1 (8 to 15 bits)
LNL: Data length + 1 (0 to 7 bits)
RES: 15h (OK)
Data: Read data
SUM: Sum of values
ETB: 17h
ETX: 03h

(4) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 15h (OK)
SUM: EAh
ETX: 03h

(5) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 95h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
D0h (address error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(6) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 95h (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

55.15.21 Configuration Clearing Command

This command is used to clear the option-setting memory (configuration setting area).
This command also erases the TM target areas. Therefore, when this command is in use, do not set the TM target areas outside the access window when setting an access window. The TM function is disabled after a reset.
While the FAW.FSPR bit is valid, a protection error occurs.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM:1Ch
SUM: E3h
ETX: 03h

(2) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 1Ch (OK)
SUM: E3h
ETX: 03h

(3) Status packet structure, error occurrence

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: 9Ch (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
C3h (flow error)
DAh (protection error)
E0h (blank error)
E1h (erase error)
E2h (program error)
E7h (flash sequencer error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

55.15.22 Simple Addition Checksum Command

This command is used to calculate the sum of values in the specified area and send the result to the host. While the TM function is enabled, however, the values in the TM target areas are not included in the calculation.
The target area of this command can be selected from the code flash memory or data flash memory. Calculation is by simple addition. If the size of each area is less than 8×2^n Kbytes, the area range from the final address of each area to 8×2^n Kbyte area is supplemented by FFh. The initial value is 0 and the sum of values in the specified area is obtained by adding the values of all bytes. If this command is issued for the data flash memory that includes erased blocks, the result is undefined. When a simple addition checksum is to be executed for the data flash memory, make sure that data have been written throughout the specified area.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	ARE	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 02h
COM: 4Dh
ARE: Area information
- 00h (code flash memory)
- 20h (data flash memory)

SUM: Sum of values
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	SD1	SD2	SD3	SD4	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 05h
RES: 4Dh (OK)
SD1: Sum of values
SD2: Sum of values
SD3: Sum of values
SD4: Sum of values
- If the sum of values is 01234567h, the settings are as follows.
- SD1 = 01h
- SD2 = 23h
- SD3 = 45h
- SD4 = 67h

SUM: Sum of values
ETX: 03h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 4Dh (OK)
SUM: B2h
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: CDh (error)
ERR: Error code
- C1h (packet error)
- C2h (checksum error)
- C3h (flow error)
- D5h (area error)

SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: CDh (error)
ERR: Error code
C1h (packet error)
C2h (checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

55.15.23 Configuration Programming Command

This command is used to receive data to be written to the option-setting memory and to program the data in the specified area.
Programming must be specified in the unit acquired by using the area information acquisition command.
This command can only be accepted in the command waiting phase.
Programming in the area including the FAW register while the FAW.FSPR bit is enabled leads to a protection error.
Once the FAW.FSPR bit has been enabled, the bit cannot be disabled.
For details of the FAW.FSPR bit, refer to section 7.2.9, Flash Access Window Setting Register (FAW) in section 7, Option-Setting Memory (OFSM).
Rewriting the SPCC.SPE bit while the connection of serial programmer is prohibited leads to normal termination, but programming does not proceed.
In addition, this MCU cannot be connected to a serial programmer at a reset while connection to a serial programmer is disabled.
Writing to the TMEF.TMEF[2:0] bits, TMINF register, and MDE.BANKMD[2:0] bits while the TMEF.TMEF[2:0] bits are 000b in linear mode leads to normal termination, but programming does not proceed.
Writing to the TMEF.TMEF[2:0] bits, TMINF register, and MDE.BANKMD[2:0] bits while the TMEF.TMEF[2:0] bits or the TMEF.TMEFDB[2:0] bits are 000b in dual mode leads to normal termination, but programming does not proceed.

(1) Procedure

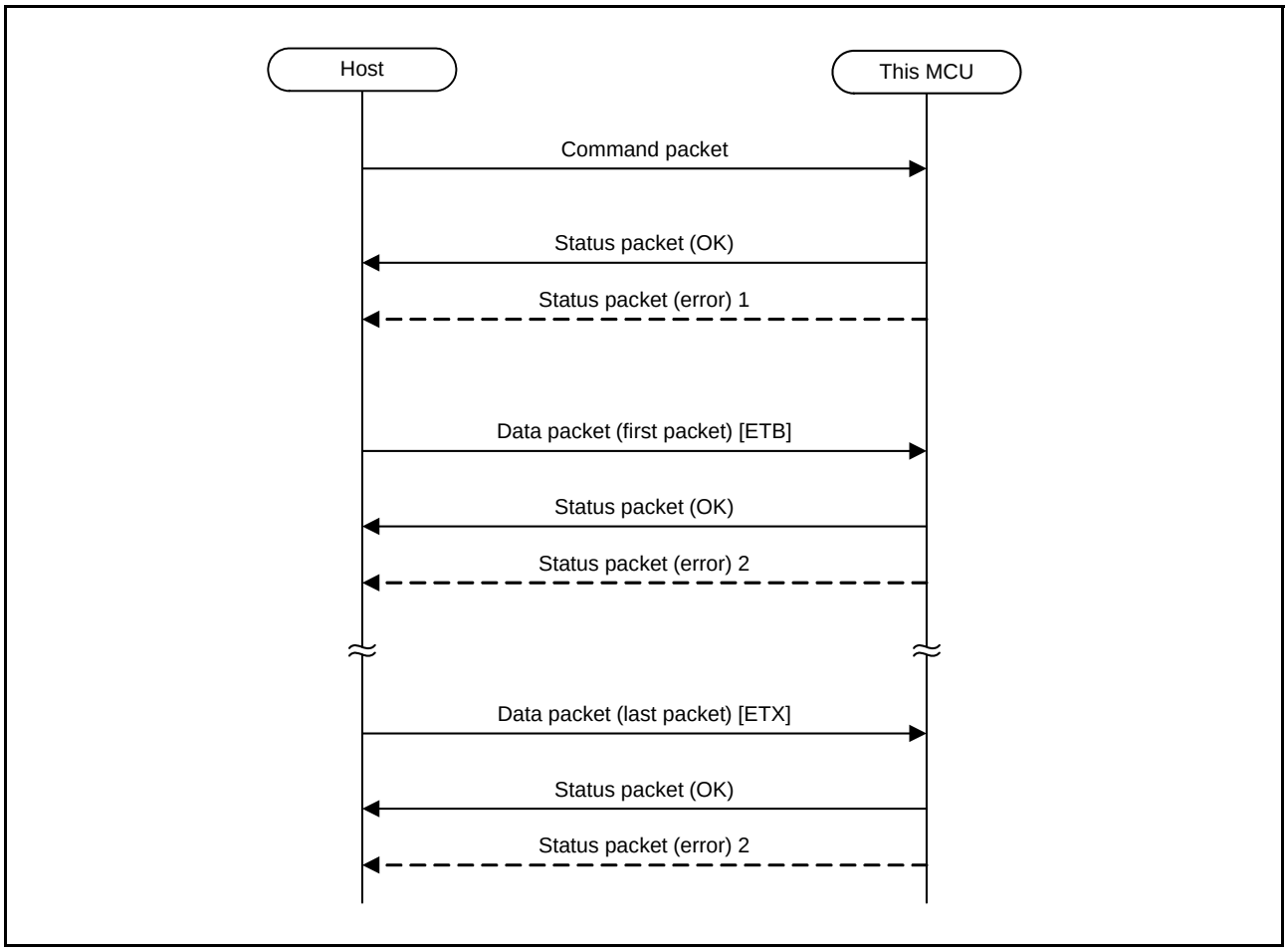


Figure 55.49 Programming Command Transfer Sequence

(2) Command packet structure

SOH	LNH	LNL	COM	SHH	SHL	SLH	SLL	EHH	EHL	ELH	ELL	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 09h
COM:51h
SHH: Start address (24 to 31 bits)
SHL: Start address (16 to 23 bits)
SLH: Start address (8 to 15 bits)
SLL: Start address (0 to 7 bits)
EHH: End address (24 to 31 bits)
EHL: End address (16 to 23 bits)
ELH: End address (8 to 15 bits)
ELL: End address (0 to 7 bits)
SUM: Sum of values
ETX: 03h

(3) Data packet structure

SOD	LNH	LNL	RES	Data	SUM	ETB	or	ETX

SOD: 81h
LNH: Data length + 1 (8 to 15 bits)
LNL: Data length + 1 (0 to 7 bits)
RES: 51h (OK)
Data: Program data*1
SUM: Sum of values
ETB: 17h
ETX: 03h

Note 1. The data for programming must be sent from the host in order from that for the lowest address to that for the highest address.

<Write data>

Address	0h	1h	2h	3h	4h	5h	6h	7h	...
Data	00h	01h	02h	03h	04h	05h	06h	07h	...

<Program data>

1st byte	**2nd byte**	**3rd byte**	**4th byte**	**5th byte**	**6th byte**	**7th byte**	**8th byte**	**...**
00h	01h	02h	03h	04h	05h	06h	07h	...

(4) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 51h (OK)
SUM: AEh
ETX: 03h

(5) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: D1h (Error)
ERR: Error code
C1h (Packet error)
C2h (Checksum error)
C3h (Flow error)
D0h (address error)
DAh (protection error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(6) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: D1h (Error)
ERR: Error code
C1h (Packet error)
C2h (Checksum error)
E1h (Erase error)
E2h (Program error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

55.15.24 Configuration Reading Command

This command is used to read data from the area specified by the option-setting memory and to transmit the data to the host.

The minimum unit for reading is 4 bytes.

This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SHH	SHL	SLH	SLL	EHH	EHL	ELH	ELL	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 09h
COM:52h
SHH: Start address (24 to 31 bits)
SHL: Start address (16 to 23 bits)
SLH: Start address (8 to 15 bits)
SLL: Start address (0 to 7 bits)
EHH: End address (24 to 31 bits)
EHL: End address (16 to 23 bits)
ELH: End address (8 to 15 bits)
ELL: End address (0 to 7 bits)
SUM: Sum of values
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	Data	SUM	ETB or ETX

SOD: 81h
LNH: Data length + 1 (8 to 15 bits)
LNL: Data length + 1 (0 to 7 bits)
RES: 52h (OK)
Data: Read data*1
SUM: Sum of values
ETB: 17h
ETX: 03h

Note 1. The data is read in the order from the lower address to the higher address.

<Program data>

Address	0h	1h	2h	3h	4h	5h	6h	7h	...
Data	00h	01h	02h	03h	04h	05h	06h	07h	...

<Read data>

1st byte	**2nd byte**	**3rd byte**	**4th byte**	**5th byte**	**6th byte**	**7th byte**	**8th byte**	**...**
00h	01h	02h	03h	04h	05h	06h	07h	...

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 52h (OK)
SUM: ADh
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: D2h (Error)
ERR: Error code
C1h (Packet error)
C2h (Checksum error)
C3h (Flow error)
D0h (address error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: D2h (Error)
ERR: Error code
C1h (Packet error)
C2h (Checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

55.15.25 Acquisition of the Number of the Area Information Command

This command is used to transmit the number of area that the MCU has to the host.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 01h
COM:53h
SUM: ACh
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	NOA	SUM	ETX

SOD: 81h
LNH:00h
LNL: 02h
RES: 53h (OK)
NOA: Number of area information (1 byte)
05h (in linear mode)
08h (in dual mode)
SUM: Sum of values
ETX: 03h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 53h (OK)
SUM: ACh
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: D3h (Error)
ERR: Error code
C1h (Packet error)
C2h (Checksum error)
C3h (Flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: D3h (Error)
ERR: Error code
C1h (Packet error)
C2h (Checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

55.15.26 Area Information Acquisition Command

This command is used to transmit information of the specified area to the host.
The address of the erase command, programming command, or configuration program command must be specified according to the range and unit (alignment) returned by this command.
This command can only be accepted in the command waiting phase.

(1) Command packet structure

SOH	LNH	LNL	COM	NUM	SUM	ETX

SOH: 01h
LNH: 00h
LNL: 02h
COM:54h
NUM: Area number [0 to NOA-1]
SUM: C5h
ETX: 03h

(2) Data packet structure

SOD	LNH	LNL	RES	KOA	SAD	EAD	EAU	WAU	SUM	ETX

SOD: 81h
LNH:00h
LNL: 12h
RES: 54h (OK)
Kind of the area
KOA =
00h (code flash memory)
20h (data flash memory)
30h (option-setting memory)
40h (Trusted Memory disabled)
41h (Trusted Memory enabled)
SAD: Start address
EAD: End address
EAU: Erase access unit (alignment) [Byte]
WAU: Write access unit (alignment) [Byte]
SUM: Sum of values
ETX: 03h

Data packet in linear mode

NUM	KOA	SAD	EAD	EAU	WAU	Description
00h	00h	FFFF 0000h	FFFF FFFFh	0000 2000h	0000 0080h	Code flash memory blocks 0 to 7 (8-Kbyte block)
01h	00h	FFE0 0000h	FFFE FFFFh	0000 8000h	0000 0080h	Code flash memory blocks 8 to 69 (32-Kbyte block)
02h	20h	0010 0000h	0010 1FFFh	0000 0040h	0000 0004h	Data flash memory
03h	30h	FE7F 5D00h	FE7F 5D7Fh	0000 0080h	0000 0010h	Option-setting memory
04h	40h	FFFE 0000h	FFFE FFFFh	0000 8000h	0000 0080h	Trusted Memory blocks 8 and 9 disabled
	41h	FFFE 0000h	FFFE FFFFh	0000 8000h	0000 0080h	Trusted Memory blocks 8 and 9 enabled

Settings of the TMEF.TMEF[2:0] bits and TMEF.TMEFDB[2:0] bits and setting of KOA while NUM is 04h in linear mode

TMEF.TMEF[2:0] bits	TMEF.TMEFDB[2:0] bits	KOA while NUM is 04h
111b	Don't care	40h
000b	Don't care	41h

Data packet in dual mode

NUM	KOA	SAD	EAD	EAU	WAU	Description
00h	00h	FFFF 0000h	FFFF FFFFh	0000 2000h	0000 0080h	Code flash memory blocks 0 to 7 (8-Kbyte block)
01h	00h	FFF0 0000h	FFFE FFFFh	0000 8000h	0000 0080h	Code flash memory blocks 8 to 37 (32-Kbyte block)
02h	00h	FFEF 0000h	FFEF FFFFh	0000 2000h	0000 0080h	Code flash memory blocks 38 to 45 (8-Kbyte block)
03h	00h	FFE0 0000h	FFEE FFFFh	0000 8000h	0000 0080h	Code flash memory blocks 46 to 75 (32-Kbyte block)
04h	20h	0010 0000h	0010 1FFFh	0000 0040h	0000 0004h	Data flash memory
05h	30h	FE7F 5D00h	FE7F 5D7Fh	0000 0080h	0000 0010h	Option-setting memory
06h	40h	FFFE 0000h	FFFE FFFFh	0000 8000h	0000 0080h	Trusted Memory blocks 8 and 9 disabled
	41h	FFFE 0000h	FFFE FFFFh	0000 8000h	0000 0080h	Trusted Memory blocks 8 and 9 enabled
07h	40h	FFEE 0000h	FFEE FFFFh	0000 8000h	0000 0080h	Trusted Memory blocks 46 and 47 disabled
	41h	FFEE 0000h	FFEE FFFFh	0000 8000h	0000 0080h	Trusted Memory blocks 46 and 47 enabled

Settings of the TMEF.TMEF[2:0] bits and TMEF.TMEFDB[2:0] bits and setting of KOA while NUM is 06h or 07h in dual mode

TMEF.TMEF[2:0] bits	TMEF.TMEFDB[2:0] bits	KOA while NUM is 06h or 07h
111b	111b	40h
000b	000b	41h

(3) Status packet structure, normal termination

SOD	LNH	LNL	RES	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 01h
RES: 54h (OK)
SUM: ABh
ETX: 03h

(4) Status packet structure, error occurrence 1

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: D4h (Error)
ERR: Error code
C1h (Packet error)
C2h (Checksum error)
C3h (Flow error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

(5) Status packet structure, error occurrence 2

SOD	LNH	LNL	RES	ERR	SUM	ETX

SOD: 81h
LNH: 00h
LNL: 02h
RES: D4h (Error)
ERR: Error code
C1h (Packet error)
C2h (Checksum error)
SUM: Sum of values
ETX: 03h

After the error code is returned, the chip returns to the command waiting state.

55.15.27 Usage Example

(1) Example of the Procedure for Reprogramming

Figure 55.50 shows an example of the procedure for reprogramming. Figure 55.51 shows an example of the method of changing to dual mode.

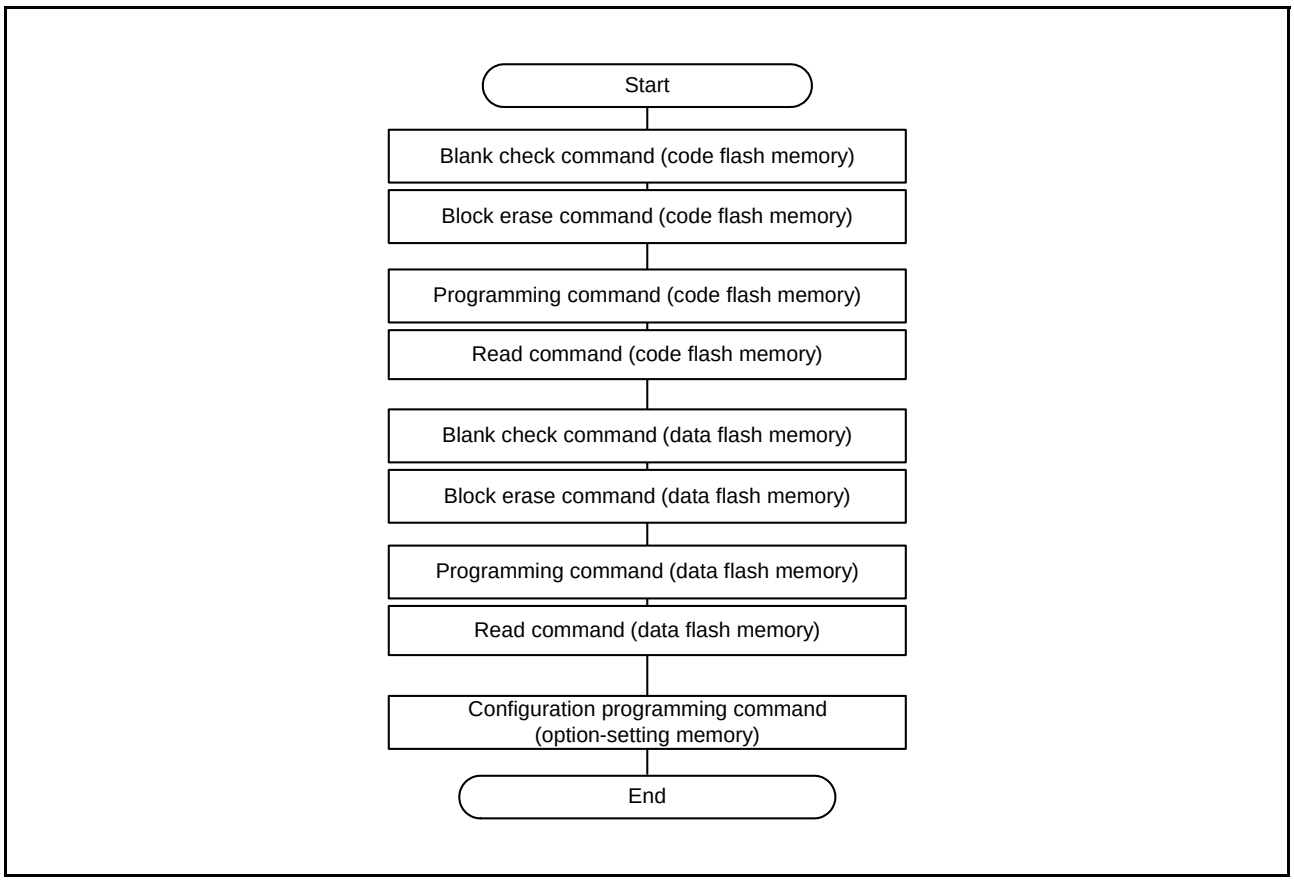


Figure 55.50 Example of Reprogramming Procedure

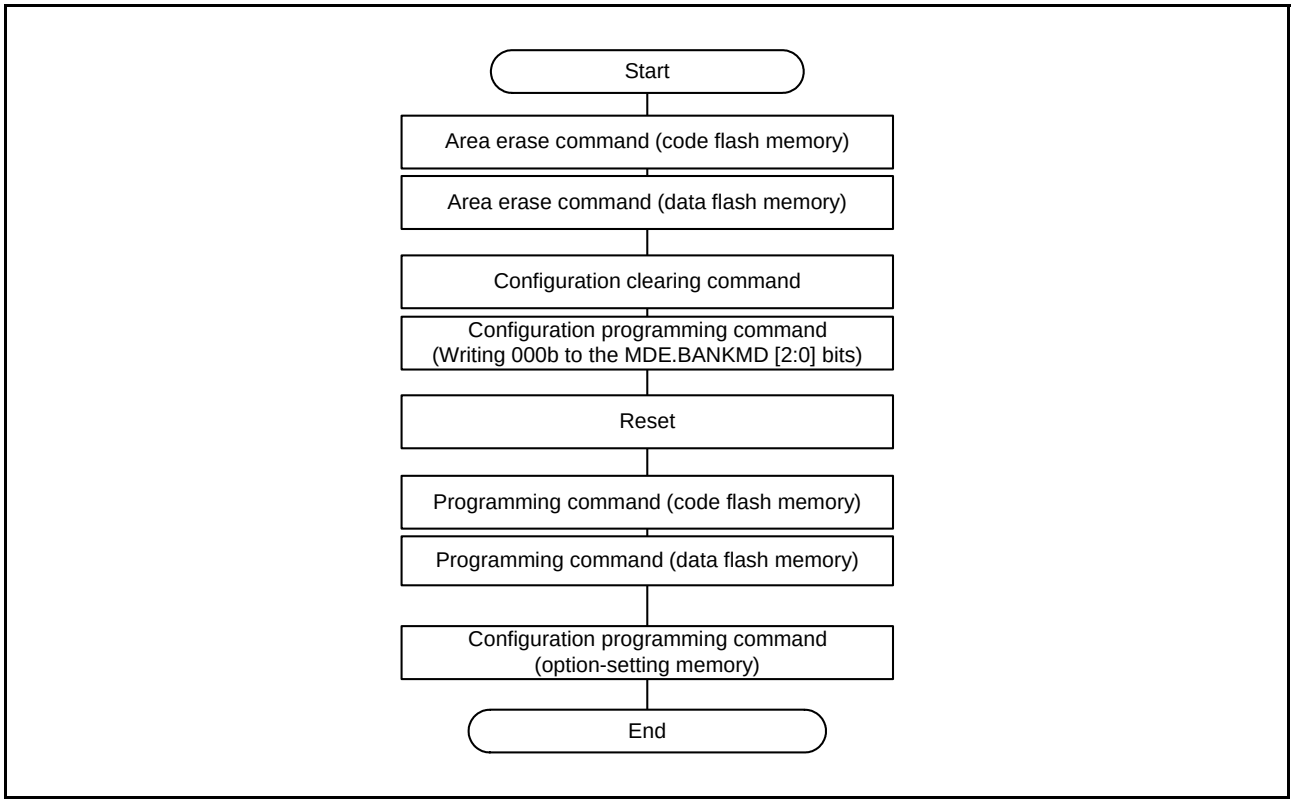


Figure 55.51 Rewriting when Switching Between Different Modes (from Linear to Dual)

55.16 Using the Serial Programmer for Rewriting

A dedicated serial programmer can be used to rewrite flash memory in boot mode.

(1) Serial Programming

By providing a connector on the board, it is possible to program the MCU with a serial programmer while the MCU is mounted on the board.

55.16.1 Environments for Serial Programming

The recommended environments for rewriting the flash memory of the MCU are described below.

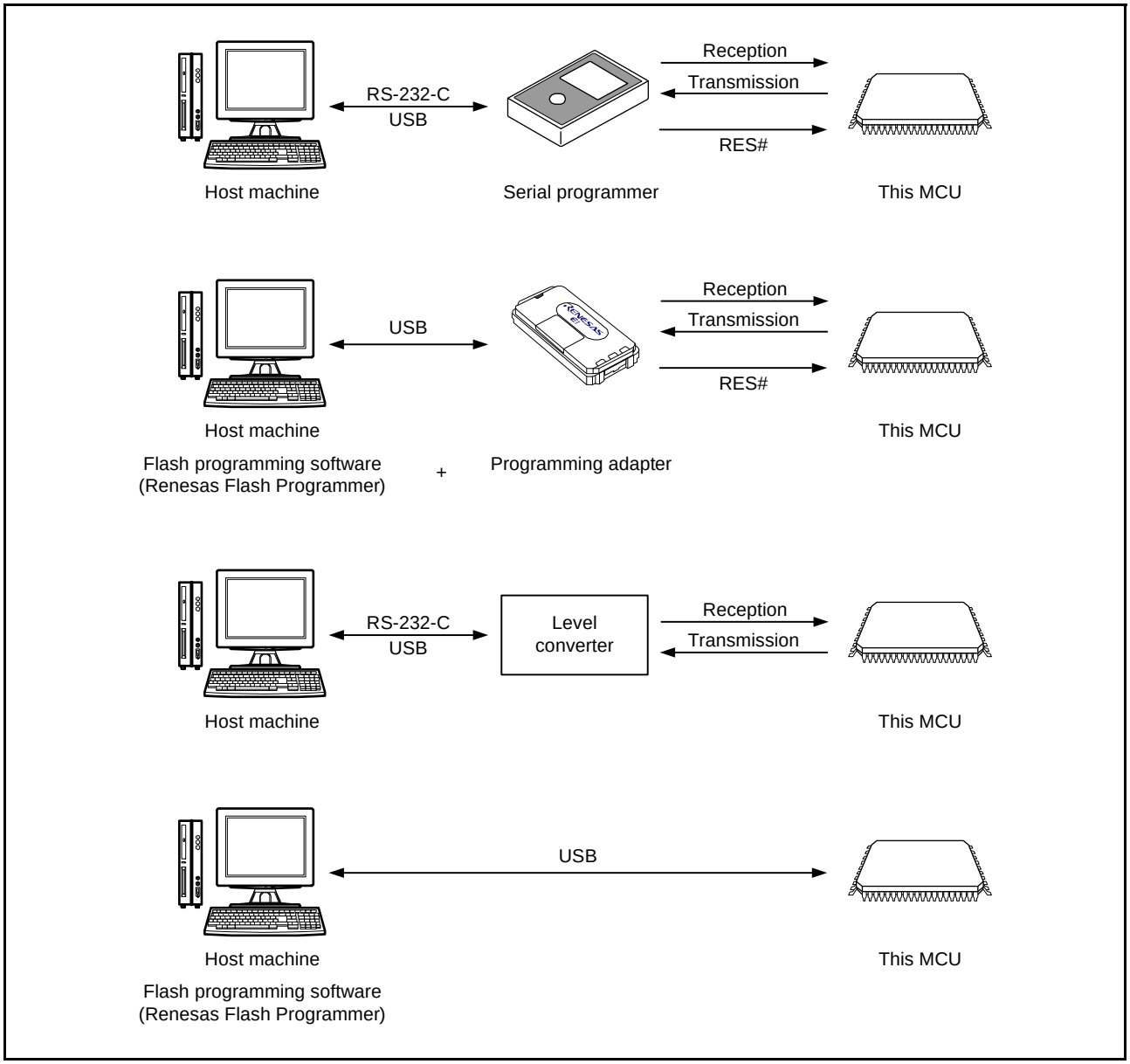


Figure 55.52 Environments for Rewriting the Flash Memory

Note: For details of the serial programmers, refer to the manual of each serial programmer; for details of the Renesas Flash Programmer flash programming software, refer to the Renesas Flash Programmer Flash Programming Software User's Manual.

55.17 Programming through Self-Programming

55.17.1 Overview

This MCU supports programming of the flash memory by the user program itself. The FACI commands can be used with user programs for writing to the flash memory. This allows upgrading of user programs and rewriting of constant data fields.

When the data flash memory is programmed, the background operation facility makes it possible to execute a programming program from the code flash memory to program the data flash memory. Furthermore, the programming program can be copied to internal RAM or external memory in advance of the programming operation, and executed from the given destination to program the data flash memory.

Background operation is also available for use when the address ranges of the area of code flash memory to be programmed and the area of code flash memory to be read satisfy particular conditions (see Table 55.25). At the time of self-programming in this case, a programming program in code flash memory can be used to program the code flash memory. Also, the programming program can be copied to internal RAM or external memory in advance of the programming operation, and executed from the given destination to program the code flash memory. This is useful when the address ranges do not satisfy the conditions for background operation.

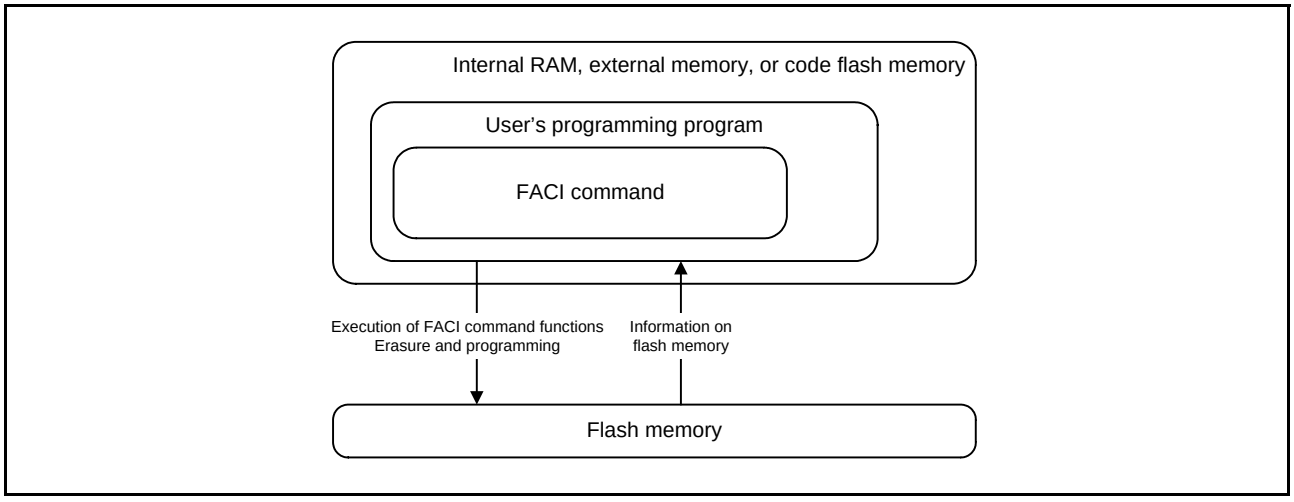


Figure 55.53 Schematic View of Self-Programming

55.17.2 Background Operation

The BGO is can be used to execute the flash rewrite routine on the code flash memory when the data flash memory or other area of code flash memory is rewritten.
Background operations can be used when the combination of the flash memory for rewriting and the flash memory for reading is any of those listed below.

Table 55.25 Conditions under which Background Operation is Usable

		Range for Rewriting	Range for Reading
Common to linear and dual modes		Data flash memory	Code flash memory
Linear mode	Products with 2 Mbytes of code flash memory	First half (1.0 Mbyte) of the code flash memory (addresses FFE0 0000h to FFEF FFFFh)	Second half (1.0 Mbyte) of the code flash memory (addresses FFF0 0000h to FFFF FFFFh) Data flash memory
		Second half (1.0 Mbyte) of the code flash memory (addresses FFF0 0000h to FFFF FFFFh)	First half (1.0 Mbyte) of the code flash memory (addresses FFE0 0000h to FFEF FFFFh) Data flash memory
	Products with 1.5 Mbytes of code flash memory	First half (0.5 Mbytes) of the code flash memory (addresses FFE8 0000h to FFEF FFFFh)	Second half (1.0 Mbyte) of the code flash memory (addresses FFF0 0000h to FFFF FFFFh) Data flash memory
		Second half (1.0 Mbyte) of the code flash memory (addresses FFF0 0000h to FFFF FFFFh)	First half (0.5 Mbytes) of the code flash memory (addresses FFE8 0000h to FFEF FFFFh) Data flash memory
Dual mode	When the BANKSEL.BANKSWP[2:0] bits are 111b:	Bank 1 area of the code flash memory	Bank 0 area of the code flash memory Data flash memory
	When the BANKSEL.BANKSWP[2:0] bits are 000b:	Bank 0 area of the code flash memory	Bank 1 area of the code flash memory Data flash memory

55.18 Reading Flash Memory

55.18.1 Reading Code Flash Memory

Special settings are not required to read code flash memory after release from the reset state. Data can simply be read out through access to addresses in the code flash memory.
When reading code flash memory that has been erased but not yet been programming again (i.e. that is in the non-programmed state), all bits are read as 1.

55.18.2 Reading Data Flash Memory

Special settings are not required to read data flash memory after release from the reset state. Data can simply be read out through access to addresses in the data flash memory.
Values read from data flash memory that has been erased but not yet been programming again (i.e. that is in the non-programmed state) are undefined. Use blank checking when you need to confirm that an area is in the non-programmed state.

55.19 Trusted Memory

This MCU has a trusted memory (hereafter called TM) facility to prevent the reading of blocks 8 and 9 and blocks 46 and 47 (in dual mode) in the code flash memory by third party software. When set as trusted memory, these areas are suitable for storing software that handles processing for encryption algorithms, device control software that is associated with proprietary or confidential know-how, purchased middleware, and so on.
Table 55.26 lists the specifications of the TM function, Table 55.27 lists access restrictions within the TM target area when the TM function is enabled, and Figure 55.54 shows the cases where the CPU is able to operate in relation to the TM target area.

Table 55.26 TM Specifications

Item	Description
TM target area	Linear mode: Blocks 8 and 9 in the code flash memory (64 Kbytes in total) Dual mode: Blocks 8 and 9 in the code flash memory (64 Kbytes in total) and blocks 46 and 47 (64 Kbytes in total)
Access restrictions when TM is enabled	See Table 55.27, Restrictions on Access to the TM Target Area while the TM function is Enabled.
How to run program code when the TM function is enabled	When the TM function is enabled, starting to run program code in an area being handled as TM is only possible with a branch instruction from program code outside the areas being handled as TM.
Interrupt processing during the execution of program code in an area being handled as TM while the TM function is enabled	Both the acceptance of requests for interrupt processing and return from interrupt processing are possible.
Security function	Enabling the TM function restricts access to program code in the areas for handling as TM to instruction fetching only
Protection functions	• Restrictions on data access to the TM target area when the TM function is enabled*1 • Once enabled, the TM function prevents its own disabling until the areas being handled as TM are erased. • Once enabled, the TM function prevents further writing to the areas being handled as TM.

Note 1. Access to data in operations that include the borders of the TM target areas is also not allowed.

Table 55.27 Restrictions on Access to the TM Target Area while the TM function is Enabled

Type of Access	CPU	DMAC/DTC/EXDMAC
Instruction fetching	Yes	—
Access to data	No*1	No

Note: The same restrictions apply to on-chip debuggers as to the DMAC in the table above. For the operation of the OCD you are using in relation to the TM target area, see the manual of the given OCD.
Note 1. Place data to which access will be required in areas other than the TM target areas.

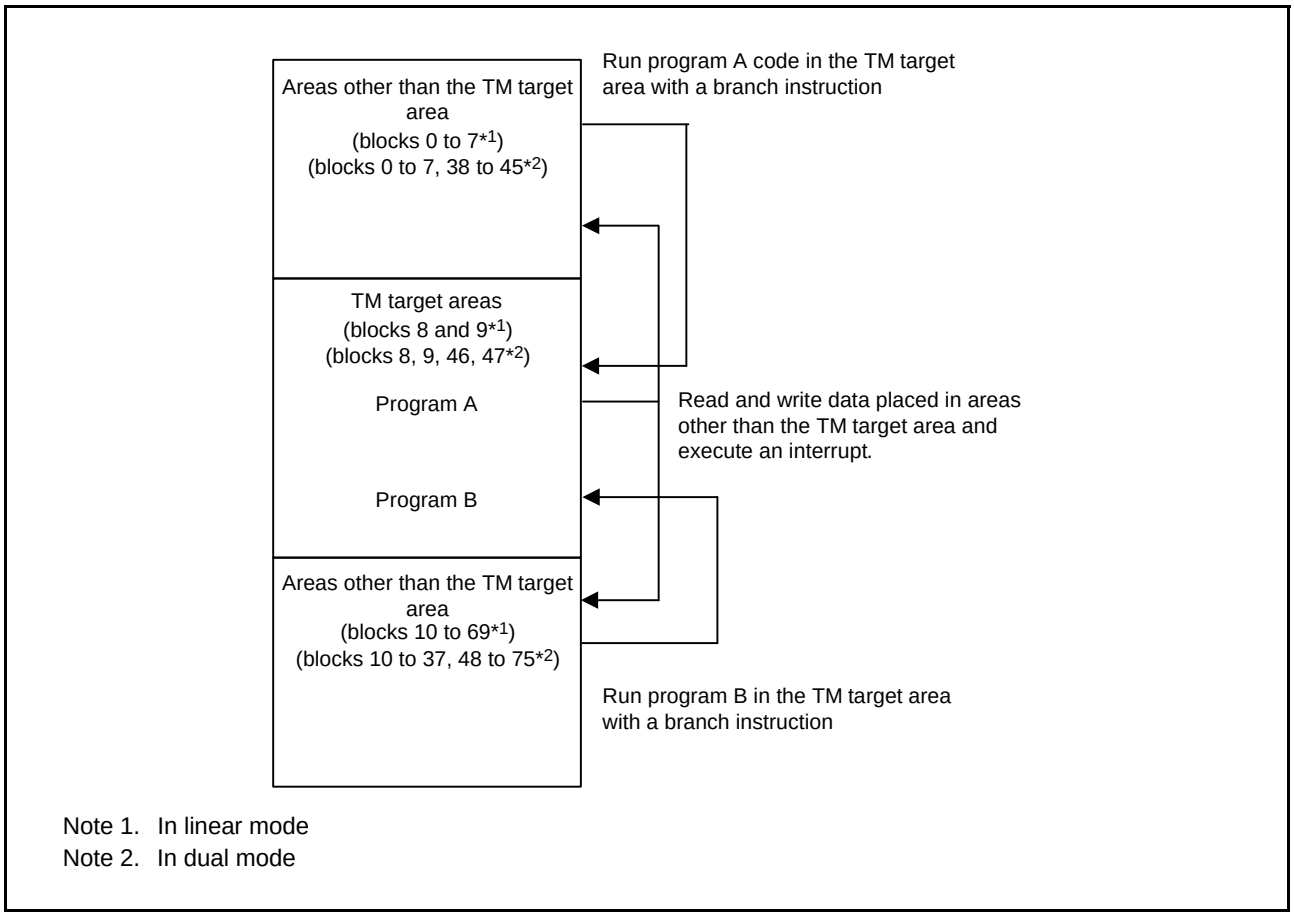


Figure 55.54 Cases where CPU is Allowed to Operate in Relation to the TM Target Area when the TM Function is Enabled

55.19.1 Allocating Program Code to the TM Target Area

When the TM function is enabled, implement countermeasures in the form of software for the TM target area as required as further measures to prevent the running of programs for access to consecutive addresses in the TM target areas from areas outside the TM target areas.

55.19.2 How to Enable the TM Function

55.19.2.1 By Self-Programming

After writing to blocks 8 and 9 of the code flash memory, i.e. the TM target areas (after programming blocks 8, 9, 46, and 47 in dual mode), use the configuration setting command of the FACI to enable the TM function.
Figure 55.55 is a flowchart of the procedure for enabling the TM function by self-programming.

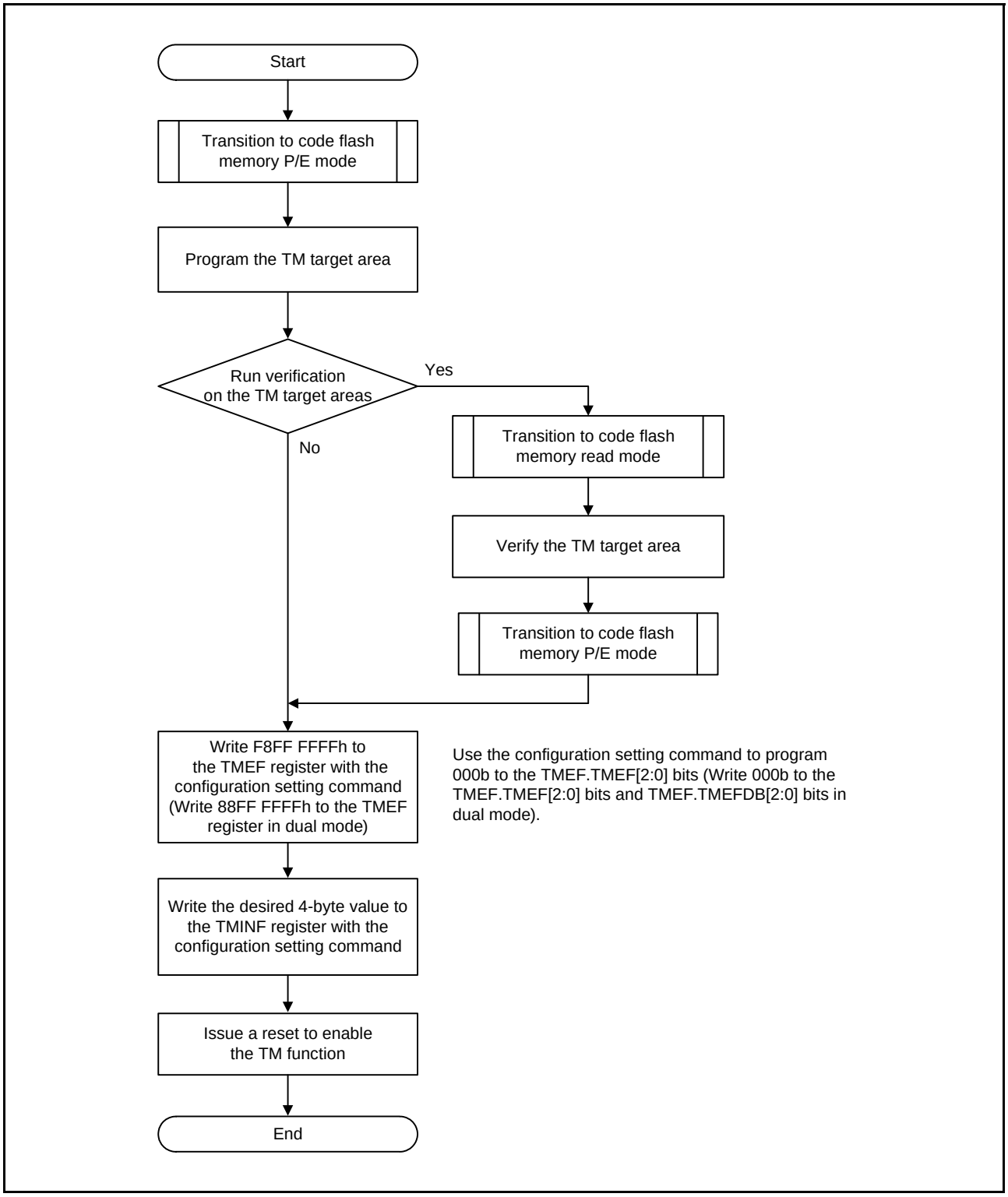


Figure 55.55 Flowchart for Enabling the TM Function by Self-Programming

55.19.2.2 By Using Boot Mode

In boot mode, use the configuration program command among the boot commands to enable the TM function after writing to blocks 8 and 9 of the code flash memory (or after writing to blocks 8, 9, 46, and 47 in dual mode).
For the configuration program command among the boot commands, see section 55.15.23, Configuration Programming Command.
Figure 55.56 is a flowchart of enabling the TM function in boot mode.

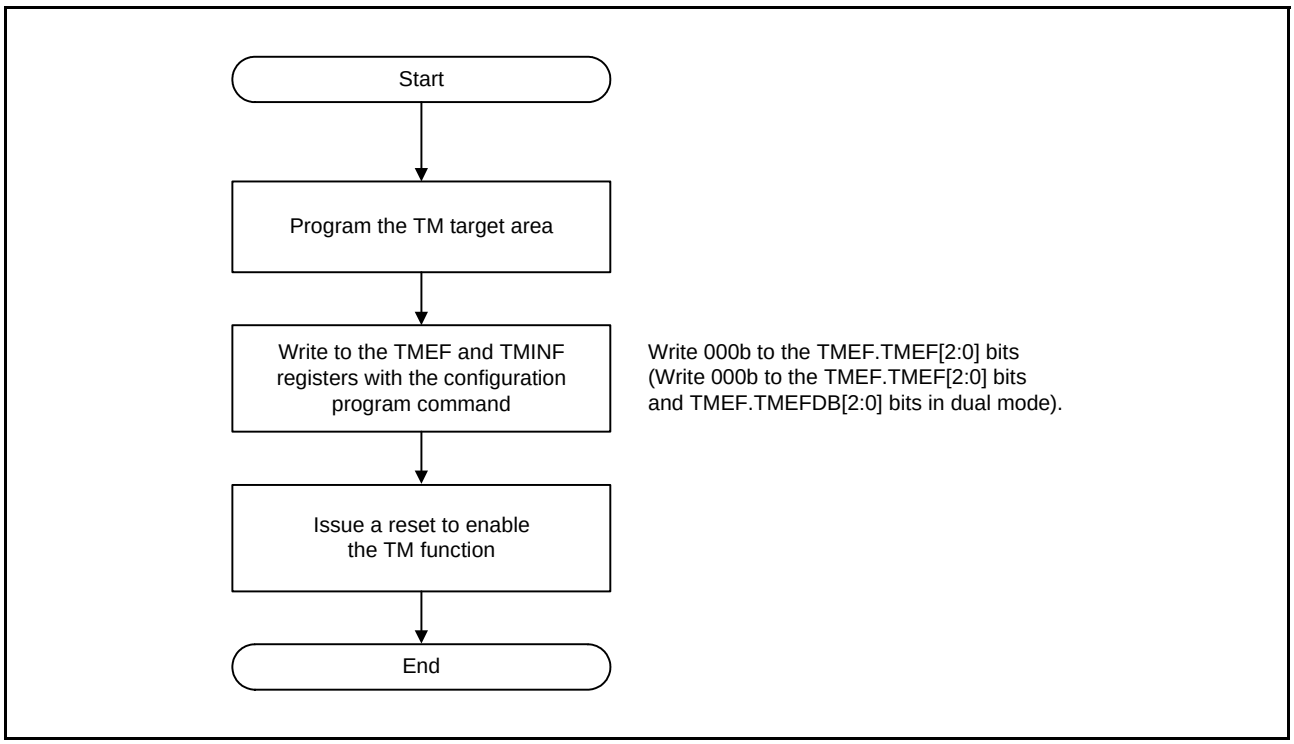
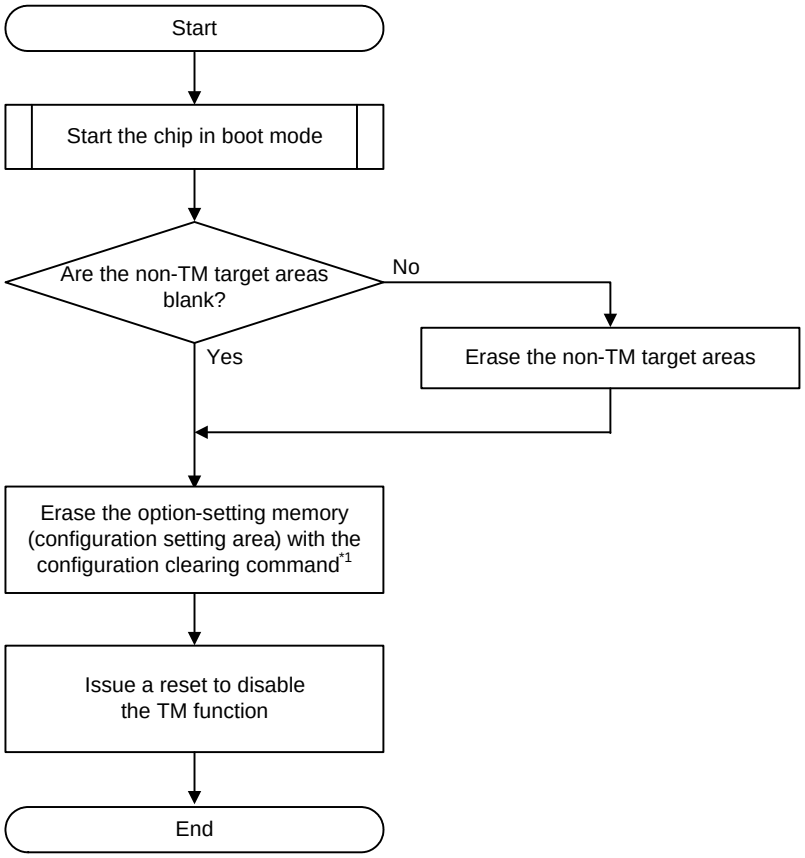


Figure 55.56 Flowchart for Enabling the TM Function by Using Boot Mode

55.19.3 How to Disable the TM function

The TM function cannot be disabled unless the TM target areas are first erased with the configuration clearing command. Do not use the configuration clearing command for this purpose unless the TM function is to be disabled.
Figure 55.57 is a flowchart of disabling the TM function in boot mode.



Note 1. The TM target area and the option-setting memory are erased by issuing the configuration clearing command.

Figure 55.57 Flowchart for Disabling the TM Function in Boot Mode

55.19.4 Notes on Enabling the TM function

55.19.4.1 Protection against Access to the TM Target Areas

While the TM function is enabled, the only type of access to the TM target areas is instruction fetching by the CPU, so do not allocate data to the TM target areas.
If the CPU, DMAC, DTC, or EXDMAC, or OCD attempt access to data in the TM target areas while the TM function is enabled, values are always read as 0 instead of the actual values.

55.19.4.2 Further Writing to the TM Target Areas

Further writing to the TM target areas is not allowed as long as the TM function is enabled.
If you do need to write further data, follow the procedure described in section 55.19.3, How to Disable the TM function, to disable the TM function, write the modified or extended data to blocks 8 and 9 of the code flash memory (blocks 8, 9, 46, and 47 in dual mode), and then follow the procedure described in section 55.19.2, How to Enable the TM Function, to enable the TM function.

55.19.4.3 Executing the Configuration Clearing Command

Follow the procedure in section 55.19.3, How to Disable the TM function, before issuing the configuration clearing command.

55.19.4.4 When the MPU Setting is for Access to the TM Target Areas

When the TM function is enabled, even if the MPU is set to allow access to the TM target areas, the TM function takes priority.

55.19.4.5 FACI Block Erase Command for the TM Target Areas

There are no special restrictions on block erasure of the TM target areas with the FACI block erase command.
Accordingly, since each area corresponds to an erase block, the areas can be erased by issuing the block erase command.

55.19.4.6 Conditions for Correct Operation of the TM Function

The TM function operates normally under the conditions prescribed in section 56, Electrical Characteristics.

55.20 Usage Notes

(1) Reading Area Where Programming/Erasure was Interrupted and Area Targeted for Suspension

The data stored in the area where programming or erasure has been suspended or the area where programming or erasure has been suspended by using the suspend command are undefined. To avoid faulty operation caused by reading undefined data, take care not to fetch instructions or read data from areas where programming or erasure was suspended and where programming or erasure was suspended by using the suspend command.

(2) Suspension During Programming/Erasure

When processing of programming/erasure is stopped by issuing the P/E suspend command, the programming/erasure processing can be resumed by issuing the P/E resume command. If the flash sequencer enters the command-locked state for any reason and issues the forced stop command after the suspended processing is normally completed and the ERSSPD flag or PRGSPD flag is set to 1, the suspended processing cannot be resumed. In addition, the values in the area where the processing was suspended are not guaranteed. Erase that area.

(3) Prohibition of Additional Programming

Programming a given area of the code flash memory or data flash memory twice is not possible. To program the code flash memory or data flash memory where has been programed, erase the target area. Programming can be added to the option-setting memory.

(4) Resets During Programming/Erasure, or Blank Checking

In the case of a reset due to the signal on the RES# pin during programming/erasure, or blank checking of the flash memory, wait for at least t_{RESWF} (see section 56, Electrical Characteristics) of the reset input period once the operating voltage is within the range stipulated in the electrical characteristics, then release the device from the reset state.

(5) Allocation of Vectors for Interrupts and Other Exceptions During Programming/Erasure

Generation of an interrupt or other exception during programming/erasure may lead to fetching of the vector from the code flash memory. Under conditions where BGO cannot be used, set the address of the vector to an address that is not in the code flash memory. Alternatively, make sure that no handling of interrupts or exceptions proceeds during programming/erasure.

(6) Items Prohibited During Programming/Erasure, or Blank Checking

High voltage is applied to the flash memory during programming/erasure, or blank checking. To prevent damage to the flash memory, do not perform the following operations.

- Have the operating voltage from the power supply go beyond the permitted range.
- Change the FWEPROR.FLWE[1:0] bits.
- Change the SYSCR0.ROME bit.
- Change the OPCCR.OPCM[2:0] bits.
- Change the SCKCR.FCK[3:0] and PCLKB[3:0] bits.
- Change the SCKCR3.CKSEL[2:0] bits.
- Change the RSTCKCR.RSTCKEN bit.
- Transition to the all module clock stop mode, software standby mode, or deep software standby mode.

(7) Notes on Program Execution in Boot Mode (USB Interface)

- An oscillator is usable in boot mode (USB interface) when its frequency is 20 or 24 MHz and when the setting value of the main clock oscillator driving ability 2 switching bits (MOFCR.MODRV2[1:0]) are 00b in the result of the matching test conducted by the oscillator manufacturer (the recommended setting value).

- To ensure that the power supply is stable during programming/erasure of flash memory, do not connect a cable via a bus-powered HUB.

(8) Programming/Erasure in Low-Speed Operating Modes 1 and 2

Do not programming/erasure the flash memory when low-speed operating mode 1 or 2 is selected with the operating power control register (OPCCR).

56. Electrical Characteristics

56.1 Absolute Maximum Ratings

Table 56.1 Absolute Maximum Rating

Conditions: VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V

Item		Symbol	Value	Unit
Power supply voltage		VCC, VCC_USB	–0.3 to +4.0	V
V_{BATT} power supply voltage		V_{BATT}	–0.3 to +4.0	V
Analog power supply voltage		AVCC0, AVCC1*1	–0.3 to +4.0	V
Reference power supply voltage		VREFH0	–0.3 to AVCC0 + 0.3 (up to 4.0)	V
Input voltage	Ports for 5 V tolerant: P12 to P17, P20, P21, P30 to P33, P67, P73, PC0 to PC3, and PJ3	V_{in}	–0.3 to VCC + 4.0 (up to 5.8)	V
	TAMPI0 to TAMPI2, RTCIC0 to RTCIC2, EXCIN*2		–0.3 to +4.0	
	Port for 5 V tolerant: P07		–0.3 to AVCC0 + 4.0 (up to 5.8)	
	P03, P05, P40 to P47		–0.3 to AVCC0 + 0.3 (up to 4.0)	
	Other than above		–0.3 to VCC + 0.3 (up to 4.0)	
Junction temperature		T_j	–40 to +125	°C
Storage temperature		T_{stg}	–55 to +125	°C

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exceeded.

Note 1. Connect the AVCC0, AVCC1, and VCC_USB pins to VCC, and the AVSS0, AVSS1, and VSS_USB pins to VSS. When the A/D converter unit 0 is not to be used, connect the VREFH0 pin to VCC and the VREFL0 pin to VSS, respectively. Do not leave these pins open. Insert capacitors of high frequency characteristics between the AVCC0 and AVSS0 pins, or AVCC1 and AVSS1 pins. Place capacitors of about 0.1 µF as close as possible to every power supply pin and use the shortest and heaviest possible traces.

Note 2. The listed values apply when pins P30, P31, and P32 are set for the TAMPIn or RTCICn (n = 0 to 2) functions, and pin PJ3 is set for the EXCIN function.

56.2 Recommended Operating Conditions

Table 56.2 Recommended Operating Conditions (1)

Item		Symbol	Min.	Typ.	Max.	Unit
Power supply voltage*1		VCC	2.7	—	3.6	V
		VSS	—	0	—	
V_{BATT} power supply voltage		V_{BATT}	1.62*2	—	3.6	V
USB power supply voltage		VCC_USB	—	VCC	—	V
		VSS_USB	—	0	—	
Analog power supply voltage*1, *3		AVCC0	—	VCC	—	V
		AVSS0	—	0	—	
		AVCC1	—	VCC	—	
		AVSS1	—	0	—	
		VREFH0	2.7	—	AVCC0	
		VREFL0	—	0	—	
Input voltage	Ports for 5 V tolerant: P12 to P17, P20, P21, P30 to P33, P67, P73, PC0 to PC3, and PJ3	V_{in}	–0.3	—	VCC + 3.6 (up to 5.5)	V
	TAMPI0 to TAMPI2, RTCIC0 to RTCIC2, EXCIN*4		–0.3	—	3.9	
	SCLHS0, SDAHS0*5		–0.3	—	VCC + 0.3	
	Port for 5 V tolerant: P07		–0.3	—	AVCC0 + 3.6 (up to 5.5)	
	P03, P05, P40 to P47		–0.3	—	AVCC0 + 0.3	
	Other than above		–0.3	—	VCC + 0.3	
Operating temperature	D version	T_{opr}	–40	—	85	°C
	G version		–40	—	105	
Junction temperature	D version	T_j	–40	—	105	°C
	G version		–40	—	125	

Note 1. Comply with the following potential condition: VCC = AVCC0 = AVCC1 = VCC_USB
Note 2. The low CL crystal unit cannot be used when the V_{BATT} voltage is less than 2.0 V.
Note 3. For details, see section 50.6.11, Voltage Range of Analog Power Supply Pins.
Note 4. The listed values apply when pins P30, P31, and P32 are set for the TAMPIn or RTCICn (n = 0 to 2) functions, and pin PJ3 is set for the EXCIN function.
Note 5. The listed values apply when pins P12 and P13 are respectively set for the SCLHS0 and SDAHS0 functions in Hs-mode of the RIICHS.

Table 56.3 Recommended Operating Conditions (2)

Item	Symbol	Value
Decoupling capacitance to stabilize the internal voltage	C_{VCL}	0.22 µF ± 30%*1

Note 1. Use a multilayer ceramic capacitor whose nominal capacitance is 0.22 µF and a capacitance tolerance is ±30% or better.

56.3 DC Characteristics

Table 56.4 DC Characteristics (1)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
$T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	IRQ input pin*1 MTU input pin*1 POE3 input pin*1 TPU input pin*1 TMR input pin*1 CMTW input pin*1 SCI input pin*1 RSCIA input pin*1 CAN input pin*1 CAC input pin*1 ADTRG# input pin*1 QSPIX input pin*1 SSIE input pin*1 REMC input pin*1 RES#, NMI, TCK	V_{IH}	0.8 × VCC	—	—	V	
		V_{IL}	—	—	0.2 × VCC		
		ΔV_T	0.06 × VCC	—	—		
	RIIC input pin RIICHS input pin (except for SMBus)	V_{IH}	0.7 × VCC	—	—		
		V_{IL}	—	—	0.3 × VCC		
		ΔV_T	0.05 × VCC	—	—		
	TAMPIn/RTCICn pin EXCIN pin	V_{IH}	0.8 × V_{BKP}	—	—		
		V_{IL}	—	—	0.2 × V_{BKP}		
	Ports for 5 V tolerant*2	V_{IH}	0.8 × VCC	—	—		
		V_{IL}	—	—	0.2 × VCC		
	Other input pins excluding ports for 5 V tolerant	V_{IH}	0.8 × VCC	—	—		
		V_{IL}	—	—	0.2 × VCC		
High level input voltage (except for Schmitt trigger input pin)	MD pin, EMLE	V_{IH}	0.9 × VCC	—	—	V	
	EXTAL, RSPI input pin, RSPIA input pin, EXDMAC input pin, WAIT#, SDHI input pin		0.8 × VCC	—	—		
	D0 to D15		0.7 × VCC	—	—		
	RIIC, RIICHS (SMBus)		2.1	—	—		
Low level input voltage (except for Schmitt trigger input pin)	MD pin, EMLE	V_{IL}	—	—	0.1 × VCC	V	
	EXTAL, RSPI input pin, RSPIA input pin, EXDMAC input pin, WAIT#, SDHI input pin		—	—	0.2 × VCC		
	D0 to D15		—	—	0.3 × VCC		
	RIIC, RIICHS (SMBus)		—	—	0.8		

Note 1. This does not include the pins, which are multiplexed as ports for 5 V tolerant.
Note 2. P07, P12 to P17, P20, P21, P30 to P33, P67, P73, PC0 to PC3, and PJ3 are 5 V tolerant.

Table 56.5 DC Characteristics (2)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
$T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output high voltage	All output pins	V_{OH}	VCC – 0.5	—	—	V	I_{OH} = –1 mA
Output low voltage	All output pins (except for RIIC0 to RIIC2 pins and RIICHS0 pin)	V_{OL}	—	—	0.5	V	I_{OL} = 1.0 mA
	RIIC0 to RIIC2 output pin RIICHS0 output pin		—	—	0.4		I_{OL} = 3.0 mA
			—	—	0.6		I_{OL} = 6.0 mA
	RIIC0 output pin RIICHS0 output pin		—	—	0.4		I_{OL} = 15.0 mA (ICFER.FMPE = 1)
			—	0.4	—		I_{OL} = 20.0 mA (ICFER.FMPE = 1)
	RIICHS0 output pin		—	—	0.4		I_{OL} = 3.0 mA (ICFER.HSME = 1)
Input leakage current	RES#, MD pin, EMLE*1, BSCANP*1, NMI	| I_{in} |	—	—	1.0	µA	V_{in} = 0 V V_{in} = VCC
Three-state leakage current (off state)	Other than ports for 5 V tolerant	| I_{TSI} |	—	—	1.0	µA	V_{in} = 0 V V_{in} = VCC
	Ports for 5 V tolerant		—	—	5.0		V_{in} = 0 V V_{in} = 5.5 V
Input pull-up resistor	Other than P35	R_{PU}	10	—	100	kΩ	V_{in} = 0 V
Input pull-down resistor	EMLE, BSCANP	R_{PD}	10	—	100	kΩ	V_{in} = VCC
Pull-up current serving as the SCLHS0 current source	SCLHS0 pin (P12)	I_{CS}	3	—	12	mA	VCC = 3.0 to 3.6 V V_{in} = 0.3 × VCC to 0.7 × VCC
Input capacitance	All input pins (except for P12, P13, P16, P17, P20, P21, EMLE, BSCANP, USB0_DP, USB0_DM, USB1_DP and USB1_DM)	C_{in}	—	—	8	pF	Vbias = 0 V Vamp = 20 mV f = 1 MHz T_a = 25°C
	P12, P13, P16, P17, P20, P21, EMLE, BSCANP, USB0_DP, USB0_DM, USB1_DP and USB1_DM		—	—	16		
Power supply voltage in the backup domain		V_{BKP}	—	VCC	—	V	VCC ≥ $V_{DETBATT}$
			—	VBATT	—		VCC < $V_{DETBATT}$
Output voltage of the VCL pin		V_{CL}	—	1.18	—	V	

Note 1. The input leakage current value at the EMLE and BSCANP pins are only when V_{in} = 0 V.

Table 56.6 DC Characteristics (3)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
$T_a = T_{opr}$

Item				Symbol	D version		G version		Unit	Test Conditions
					Typ.	Max.	Typ.	Max.		
Supply current*1	High-speed operating mode	Full operation*2		I_{CC}*3	—	55	—	68	mA	ICLK = 120 MHz, PCLKA = 120 MHz, PCLKB = 60 MHz, PCLKC = 60 MHz, PCLKD = 60 MHz, FCLK = 60 MHz, BCLK = 120 MHz, BCLK pin = 60 MHz
		Normal operation	Peripheral module clocks are supplied *4		23	—	23	—		
			Peripheral module clocks are stopped *4, *5		13	—	13	—		
		Core Mark	Peripheral module clocks are stopped *4, *5		14.5	—	14.5	—		
		Sleep mode: Peripheral module clocks are supplied*4			20	38	20	51		
		All module clock stop mode (reference value)			9	26	9	39		
		Increased by BGO operation*8	Reading from the code flash memory while the data flash memory is being programmed		6	—	6	—		
			Reading from the code flash memory while the code flash memory is being programmed		7	—	7	—		
		Increased by Trusted Secure IP operation			—	15	—	15		
	Low-speed operating mode 1: Peripheral module clocks are stopped*4				1.6	—	1.6	—		All clocks 1 MHz
	Low-speed operating mode 2: Peripheral module clocks are stopped*4				1.6	—	1.6	—		All clocks 32.768 kHz
	Software standby mode				1.1	18	1.1	27		
	Deep software standby mode	Power is supplied to the standby RAM, USB resume detecting unit (USB0 only) and REMC			15.5	69	15.5	85	µA	
		Power is not supplied to the standby RAM, USB resume detecting unit (USB0 only) and REMC	Low power consumption function of the power-on reset circuit is disabled*6		11.5	42	11.5	54		
			Low power consumption function of the power-on reset circuit is enabled*7		4.9	32	4.9	47		
		Increase current by operating RTC	When a low C_L crystal is in use		1	—	1	—		
			When a standard C_L crystal is in use		2	—	2	—		
		Increase current by operating REMC	External clock (32 kHz) input		0.1	—	0.1	—		
			When a standard C_L crystal is in use		1.4	—	1.4	—		
	When the RTC is operating while VCC is not supplied (Only the RTC and sub-clock oscillator operate with the battery backup function)		When a low C_L crystal is in use		0.9	—	0.9	—		V_{BATT} = 2.0 V, VCC = 0 V
					1.6	—	1.6	—		V_{BATT} = 3.3 V, VCC = 0 V
			When a standard C_L crystal is in use		1.6	—	1.6	—		V_{BATT} = 1.62 V, VCC = 0 V
					1.7	—	1.7	—		V_{BATT} = 2.0 V, VCC = 0 V
					3.3	—	3.3	—		V_{BATT} = 3.3 V, VCC = 0 V
	Inrush current on release from deep software standby mode		Inrush current*9	I_{RUSH}	—	130	—	130	mA	
			Total inrush current*9	E_{RUSH}	—	1	—	1	µC	

Note 1. Supply current values are measured when all output pins are unloaded and all input pull-up resistors are disabled.

Note 2. Peripheral module clocks are supplied.
Note 3. I_{CC} depends on the f (ICLK) as follows (when ICLK/PCLKA : PCLKB/PCLKC/PCLKD : BCLK : BCLK pin = 2 : 1 : 2 : 1 and EXTAL = 12 MHz).
- D version
  I_{CC} max = 0.28 × f + 21.0 (full operation in high-speed operating mode)
  I_{CC} typ = 0.16 × f + 3.5 (normal operation in high-speed operating mode)
  I_{CC} typ = 0.20 × f + 1.4 (ICLK 1 MHz max) (low-speed operating mode 1)
  I_{CC} max = 0.14 × f + 21.0 (sleep mode)
- G version
  I_{CC} max = 0.31 × f + 30.0 (full operation in high-speed operating mode)
  I_{CC} typ = 0.16 × f + 3.5 (normal operation in high-speed operating mode)
  I_{CC} typ = 0.20 × f + 1.4 (ICLK 1 MHz max) (low-speed operating mode 1)
  I_{CC} max = 0.17 × f + 30.0 (sleep mode)

Note 4. Whether the peripheral module clocks are supplied or stopped is controlled only by the bit settings in the module stop control registers A to D.
Note 5. When the peripheral module clock is stopped, the settings of the clock frequency are as follows:
ICLK = 120 MHz and PCLKA = PCLKB = PCLKC = PCLKD = FCLK = BCLK = BCLK pin = 3.75 MHz (divided by 64).
Note 6. When the low power consumption function is disabled, the DEEPCUT[1:0] bits are set to 01b.
Note 7. When the low power consumption function is enabled, the DEEPCUT[1:0] bits are set to 11b.
Note 8. These are the increases during programming of the code flash memory after the code flash memory (limitations apply to the combinations of address ranges of the program area and the readable area) or the data flash memory has been programmed or erased.
Note 9. Reference value

Table 56.7 DC Characteristics (4)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
$T_a = T_{opr}$

Item			Symbol	D version			G version			Unit	Test Conditions
				Min.	Typ.	Max.	Min.	Typ.	Max.		
Analog power supply current*1, *2	During 12-bit A/D conversion (unit 0)		I_{AVCC0}	—	0.8	1	—	0.8	1	mA	
	During 12-bit A/D conversion (unit 1)		I_{AVCC1}	—	0.6	1	—	0.6	1	mA	
	During 12-bit A/D conversion (unit 1) + temperature sensor			—	0.7	1.1	—	0.7	1.1		
	Waiting for A/D and temperature sensor conversion (all units)		I_{AVCC}	—	0.9	1.4	—	0.9	1.4	mA	$I_{AVCC} = I_{AVCC0} + I_{AVCC1}$
	A/D and temperature sensor are in standby mode (all units)			—	1.4	6.7	—	1.4	9.0	µA	
Reference power supply current	During 12-bit A/D conversion (unit 0)		I_{VREFH0}	—	38	60	—	38	60	µA	
	Waiting for 12-bit A/D conversion (unit 0)			—	0.07	0.5	—	0.07	0.6		
	12-bit A/D converter in module stop status (unit 0)			—	0.07	0.4	—	0.07	0.5		
USB operating current (increase per channel)		Low speed	$I_{CCUSBLS}$	—	3.7	6.5	—	3.7	6.5	mA	
		Full speed	$I_{CCUSBFS}$	—	4.2	10	—	4.2	10	mA	
CTSU operating current			I_{CTSU}	—	100	—	—	100	—	µA	
RAM retension voltage			V_{RAM}	2.7	—	—	2.7	—	—	V	
VCC rising gradient			SrVCC	8.4	—	20000	8.4	—	20000	µs/V	
VCC falling gradient*3			SfVCC	8.4	—	—	8.4	—	—	µs/V	

Note 1. The reference power supply current is included in the power supply current value for 12-bit A/D converter (unit 1).
Note 2. The analog power supply current cannot be separated into I_{AVCC0} and I_{AVCC1} in the 48-pin products because AVCC0 and AVCC1 share the same pin.
Note 3. This applies when V_{BATT} is used.

Table 56.8 Permissible Output Currents

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
$T_a = T_{opr}$

Item			Symbol	Min.	Typ.	Max.	Unit
Permissible output low current (average value per pin)	All output pins*1	Normal drive	I_{OL}	—	—	2.0	mA
	All output pins*2	High drive		—	—	3.8	
	All output pins*3	High-speed interface high-drive		—	—	7.5	
Permissible output low current (max. value per pin)	All output pins*1	Normal drive	I_{OL}	—	—	4.0	mA
	All output pins*2	High drive		—	—	7.6	
	All output pins*3	High-speed interface high-drive		—	—	15	
Permissible output low current (total)	Total of all output pins		ΣI_{OL}	—	—	80	mA
Permissible output high current (average value per pin)	All output pins*1	Normal drive	I_{OH}	—	—	–2.0	mA
	All output pins*2	High drive		—	—	–3.8	
	All output pins*3	High-speed interface high-drive		—	—	–7.5	
Permissible output high current (max. value per pin)	All output pins*1	Normal drive	I_{OH}	—	—	–4.0	mA
	All output pins*2	High drive		—	—	–7.6	
	All output pins*3	High-speed interface high-drive		—	—	–15	
Permissible output high current (total)	Total of all output pins		ΣI_{OH}	—	—	–80	mA

Caution: To protect the LSI's reliability, the output current values should not exceed the values in this table.

Note 1. This is the value when normal driving ability is set with a pin for which normal driving ability is selectable.

Note 2. This is the value when high driving ability is set with a pin for which normal driving ability is selectable or the value of the pin to which high driving ability is fixed.

Note 3. This is the value when high-speed interface high-driving ability is set with a pin for which high-speed interface high-driving ability is selectable.

Table 56.9 Normal Output Characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V, T_a = 25°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output high level voltage	Normal drive (P00 to P02, P12 to P14, P27, P36, P40 to P47, P50 to P52, P54 to P56, P72, P74 to P77, P80 to P83, P90 to P93, PA0 to PA7, PB0 to PB7, PC0 to PC7, PD0 to PD7, PE0 to PE7, PH1, PH2)	V_{OH}	—	3.26	—	V	I_{OH} = –0.5 mA
			—	3.22	—		I_{OH} = –1.0 mA
			—	3.13	—		I_{OH} = –2.0 mA
			—	2.94	—		I_{OH} = –4.0 mA
	High drive (P00 to P02, P03, P05, P07, P12 to P17, P20 to P27, P30 to P34, P37, P50 to P56, P60 to P67, P70 to P77, P80 to P83, P86, P87, P90 to P93, PA0 to PA7, PB0 to PB7, PC0 to PC7, PD0 to PD7, PE0 to PE7, PF5, PH1, PH2, PJ3, PJ5)		—	3.28	—		I_{OH} = –0.5 mA
			—	3.25	—		I_{OH} = –1.0 mA
			—	3.20	—		I_{OH} = –2.0 mA
			—	3.10	—		I_{OH} = –4.0 mA
	Highs-peed interface high-drive (P00 to P02, P12 to P14, P17, P20 to P23, P27, P30, P31, P50 to P56, P70, P72, P73 to P77, P80 to P83, P86, P87, P90 to P93, PA0 to PA7, PB0 to PB7, PC0 to PC7, PD0 to PD7, PE0 to PE7, PH1, PH2)		—	3.29	—		I_{OH} = –0.5 mA
			—	3.28	—		I_{OH} = –1.0 mA
			—	3.25	—		I_{OH} = –2.0 mA
			—	3.20	—		I_{OH} = –4.0 mA
			—	3.18	—		I_{OH} = –5.0 mA
Output low level voltage	Normal drive (P00 to P02, P12 to P14, P27, P36, P40 to P47, P50 to P52, P54 to P56, P72, P74 to P77, P80 to P83, P90 to P93, PA0 to PA7, PB0 to PB7, PC0 to PC7, PD0 to PD7, PE0 to PE7, PH1, PH2)	V_{OL}	—	0.04	—		I_{OL} = 0.5 mA
			—	0.09	—		I_{OL} = 1.0 mA
			—	0.18	—		I_{OL} = 2.0 mA
			—	0.39	—		I_{OL} = 4.0 mA
	High drive (P00 to P02, P03, P05, P07, P12 to P17, P20 to P27, P30 to P34, P37, P50 to P56, P60 to P67, P70 to P77, P80 to P83, P86, P87, P90 to P93, PA0 to PA7, PB0 to PB7, PC0 to PC7, PD0 to PD7, PE0 to PE7, PF5, PH1, PH2, PJ3, PJ5)		—	0.02	—		I_{OL} = 0.5 mA
			—	0.04	—		I_{OL} = 1.0 mA
			—	0.09	—		I_{OL} = 2.0 mA
			—	0.18	—		I_{OL} = 4.0 mA
	Highs-peed interface high-drive (P00 to P02, P12 to P14, P17, P20 to P23, P27, P30, P31, P50 to P56, P70, P72, P73 to P77, P80 to P83, P86, P87, P90 to P93, PA0 to PA7, PB0 to PB7, PC0 to PC7, PD0 to PD7, PE0 to PE7, PH1, PH2)		—	0.01	—		I_{OL} = 0.5 mA
			—	0.02	—		I_{OL} = 1.0 mA
			—	0.04	—		I_{OL} = 2.0 mA
			—	0.09	—		I_{OL} = 4.0 mA
			—	0.11	—		I_{OL} = 5.0 mA
			—	0.23	—		I_{OL} = 10.0 mA
			—	0.36	—		I_{OL} = 15.0 mA

Table 56.10 Thermal Resistance Value (Reference)

Item	Package	Symbol	Max.	Unit	Test Conditions
Thermal resistance	144-pin LFQFP (PLQP0144KA-B)	θ_{ja}	48.4	°C/W	JESD51-2 and JESD51-7 compliant
	100-pin LFQFP (PLQP0100KB-B)		51.7		
	64-pin LFQFP (PLQP0064KB-C)		51.2		
	48-pin HWQFN (PWQN0048KC-A)		19.1*1		
	145-pin TFLGA (PTLG0145JC-A)		30.9		JESD51-2 and JESD51-9 compliant
	145-pin TFLGA (PTLG0145KB-A)		30.6		
	100-pin TFLGA (PTLG0100JB-A)		30.9		
	64-pin TFBGA (PTBG0064KB-A)		32.0		
	144-pin LFQFP (PLQP0144KA-B)	Ψ_{jt}	1.2	°C/W	JESD51-2 and JESD51-7 compliant
	100-pin LFQFP (PLQP0100KB-B)		1.2		
	64-pin LFQFP (PLQP0064KB-C)		1.2		
	48-pin HWQFN (PWQN0048KC-A)		0.1*1		
	145-pin TFLGA (PTLG0145JC-A)		0.4		JESD51-2 and JESD51-9 compliant
	145-pin TFLGA (PTLG0145KB-A)		0.4		
	100-pin TFLGA (PTLG0100JB-A)		0.4		
	64-pin TFBGA (PTBG0064KB-A)		0.4		

Note: The values are reference values when the 4-layer board is used. Thermal resistance depends on the number of layers or size of the board. For details, refer to the JEDEC standards.

Note 1. This value applies when the exposed die pad for this purpose is connected to VSS.

56.4 AC Characteristics

Table 56.11 Operating Frequency (High-Speed Operating Mode)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
$T_a = T_{opr}$

Item			Symbol	Min.	Typ.	Max.	Unit
Operating frequency	System clock (ICLK)		f	—	—	120	MHz
	Peripheral module clock (PCLKA)			—	—	120	
	Peripheral module clock (PCLKB)			—	—	60	
	Peripheral module clock (PCLKC)			—	—	60	
	Peripheral module clock (PCLKD)			—	—	60	
	Flash-IF clock (FCLK)			—*1	—	60	
	External bus clock (BCLK)	Package of 144 pins or more		—	—	120	
		100-pin package		—	—	60	
	BCLK pin output	Package of 144 pins or more		—	—	60	
		100-pin package		—	—	30	
	SDRAM clock (SDCLK)	Package of 144 pins or more		—	—	60	
	SDCLK pin output	Package of 144 pins or more		—	—	60	

Note 1. The FCLK must run at a frequency of at least 4 MHz when changing the flash memory contents.

Table 56.12 Operating Frequency (Low-Speed Operating Mode 1)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
$T_a = T_{opr}$

Item			Symbol	Min.	Typ.	Max.	Unit
Operating frequency	System clock (ICLK)		f	—	—	1	MHz
	Peripheral module clock (PCLKA)			—	—	1	
	Peripheral module clock (PCLKB)			—	—	1	
	Peripheral module clock (PCLKC)*1			—	—	1	
	Peripheral module clock (PCLKD)*1			—	—	1	
	Flash-IF clock (FCLK)			—	—	1	
	External bus clock (BCLK)	Package of 144 pins or more		—	—	1	
		100-pin package		—	—	1	
	BCLK pin output	Package of 144 pins or more		—	—	1	
		100-pin package		—	—	1	
	SDRAM clock (SDCLK)	Package of 144 pins or more		—	—	1	
	SDCLK pin output	Package of 144 pins or more		—	—	1	

Note 1. When the 12-bit A/D converter is used, the frequency must be set to at least 1 MHz.

Table 56.13 Operating Frequency (Low-Speed Operating Mode 2)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
$T_a = T_{opr}$

Item			Symbol	Min.	Typ.	Max.	Unit
Operating frequency	System clock (ICLK)		f	32	—	264	kHz
	Peripheral module clock (PCLKA)			—	—	264	
	Peripheral module clock (PCLKB)			—	—	264	
	Peripheral module clock (PCLKC)*1			—	—	264	
	Peripheral module clock (PCLKD)*1			—	—	264	
	Flash-IF clock (FCLK)			32	—	264	
	External bus clock (BCLK)	Package of 144 pins or more		—	—	264	
		100-pin package		—	—	264	
	BCLK pin output	Package of 144 pins or more		—	—	264	
		100-pin package		—	—	264	
	SDRAM clock (SDCLK)	Package of 144 pins or more		—	—	264	
	SDCLK pin output	Package of 144 pins or more		—	—	264	

Note 1. The 12-bit A/D converter cannot be used.

56.4.1 Reset Timing

Table 56.14 Reset Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
$T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
RES# pulse width	Power-on	t_{RESWP}	1	—	—	ms	Figure 56.1
	Deep software standby mode	t_{RESWD}	0.6	—	—	ms	Figure 56.2
	Software standby mode, low-speed operating mode 2	t_{RESWS}	0.3	—	—	ms	
	Programming or erasure of the code flash memory, or programming, erasure or blank checking of the data flash memory	t_{RESWF}	200	—	—	µs	
	Other than above	t_{RESW}	200	—	—	µs	
Waiting time after release from the RES# pin reset		t_{RESWT}	54	—	55	t_{Lcyc}	Figure 56.1
Internal reset time (independent watchdog timer reset, watchdog timer reset, software reset)		t_{RESW2}	100	—	108	t_{Lcyc}	

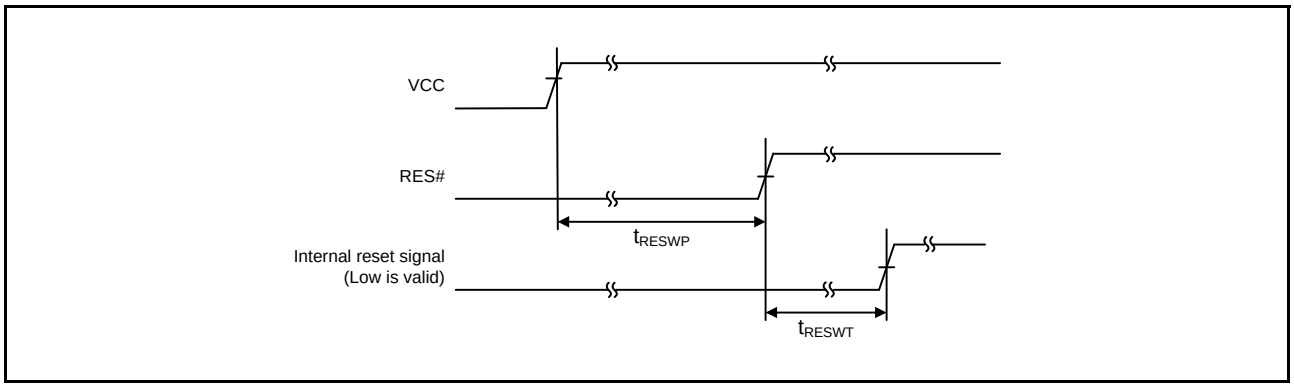


Figure 56.1 Reset Input Timing at Power-On

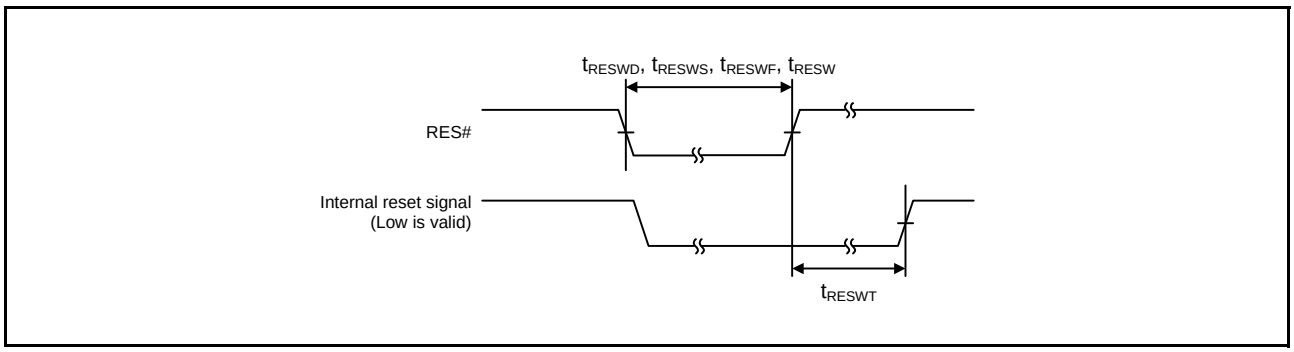


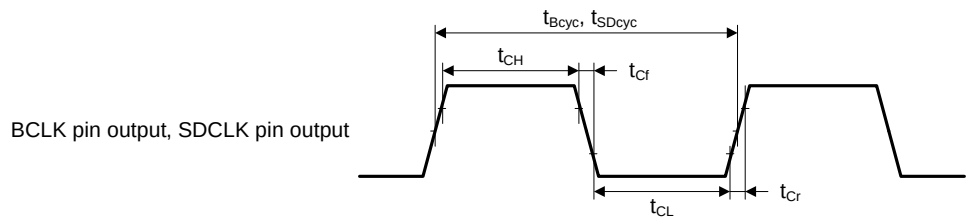
Figure 56.2 Reset Input Timing

56.4.2 Clock Timing

Table 56.15 BCLK Pin Output, SDCLK Pin Output Clock Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
$T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
BCLK pin output cycle time	Package of 144 pins or more	t_{Bcyc}	16.6	—	—	ns	Figure 56.3
	100-pin package		33.2	—	—		
BCLK pin output high pulse width		t_{CH}	3.3	—	—	ns	
BCLK pin output low pulse width		t_{CL}	3.3	—	—	ns	
BCLK pin output rising time		t_{Cr}	—	—	5	ns	
BCLK pin output falling time		t_{Cf}	—	—	5	ns	
SDCLK pin output cycle time	Package of 144 pins or more	t_{Bcyc}	16.6	—	—	ns	
SDCLK pin output high pulse width		t_{CH}	3.3	—	—	ns	
SDCLK pin output low pulse width		t_{CL}	3.3	—	—	ns	
SDCLK pin output rising time		t_{Cr}	—	—	5	ns	
SDCLK pin output falling time		t_{Cf}	—	—	5	ns	



Test conditions: V_{OH} = 0.7 × VCC, V_{OL} = 0.3 × VCC, C = 30 pF

Figure 56.3 BCLK Pin and SDCLK Pin Output Timing

Table 56.16 EXTAL Clock Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
$T_a = T_{opr}$

Item	Symbol	f_{EXMAIN} ≤ 24MHz			f_{EXMAIN} > 24MHz			Unit	Test Conditions
		Min.	Typ.	Max.	Min.	Typ.	Max.		
EXTAL external clock input cycle time	t_{EXcyc}	41.66	—	—	33.33	—	—	ns	Figure 56.4
EXTAL external clock input frequency	f_{EXMAIN}	—	—	24	—	—	30	MHz	
EXTAL external clock input high pulse width	t_{EXH}	15.83	—	—	13.33	—	—	ns	
EXTAL external clock input low pulse width	t_{EXL}	15.83	—	—	13.33	—	—	ns	
EXTAL external clock rising time	t_{EXr}	—	—	5	—	—	5	ns	
EXTAL external clock falling time	t_{EXf}	—	—	5	—	—	5	ns	

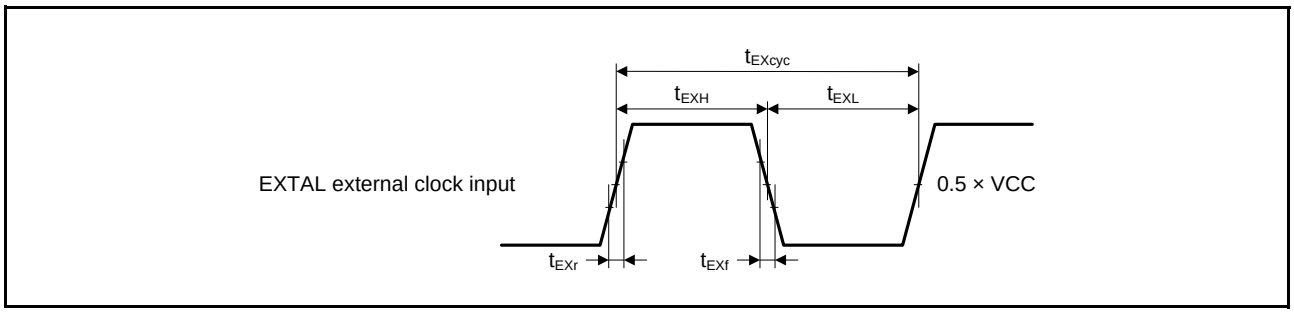


Figure 56.4 EXTAL External Clock Input Timing

Table 56.17 Main Clock Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
$T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Main clock oscillation frequency	f_{MAIN}	8	—	24	MHz	
Main clock oscillator stabilization time (crystal)	$t_{MAINOSC}$	—	—	—*1	ms	Figure 56.5
Main clock oscillation stabilization waiting time (crystal)	$t_{MAINOSCWT}$	—	—	—*2	ms	

Note 1. When using a main clock, ask the manufacturer of the oscillator to evaluate its oscillation. Refer to the results of evaluation provided by the manufacturer for the oscillation stabilization time.

Note 2. The number of cycles selected by the value of the MOSCWTCR.MSTS[7:0] bits determines the main clock oscillation stabilization waiting time in accord with the formula below.
$t_{MAINOSCWT}$ = [(MSTS[7:0] bits × 32) + 10] / f_{LOCO}

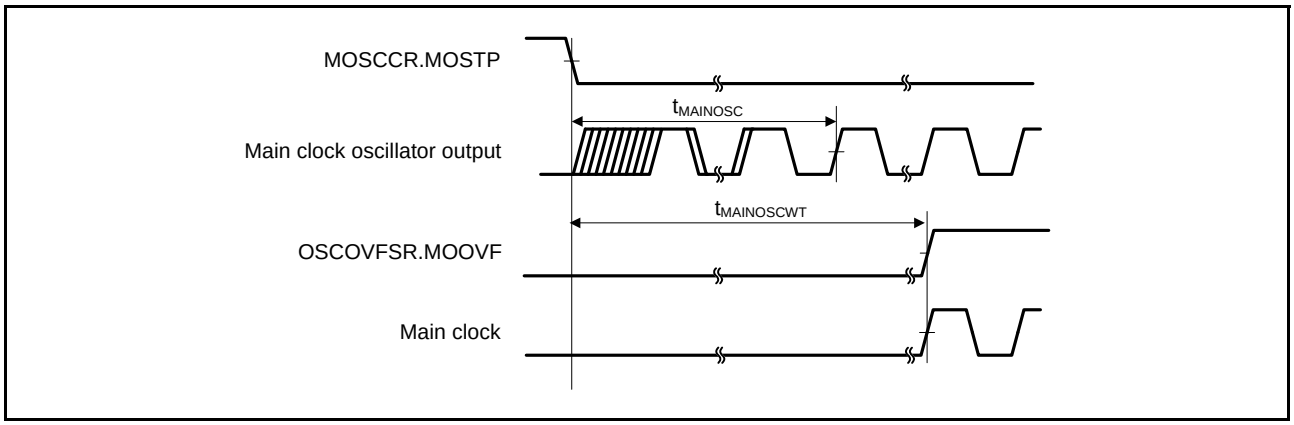


Figure 56.5 Main Clock Oscillation Start Timing

Table 56.18 LOCO and IWDT-Dedicated Low-Speed Clock Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
$T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
LOCO clock cycle time	t_{Lcyc}	3.78	4.16	4.63	µs	
LOCO clock oscillation frequency	f_{LOCO}	216 (–10%)	240	264 (+10%)	kHz	
LOCO clock oscillation stabilization waiting time	t_{LOCOWT}	—	—	44	µs	Figure 56.6
IWDT-dedicated low-speed clock cycle time	t_{ILcyc}	7.57	8.33	9.26	µs	
IWDT-dedicated low-speed clock oscillation frequency	f_{ILOCO}	108 (–10%)	120	132 (+10%)	kHz	
IWDT-dedicated low-speed clock oscillation stabilization waiting time	$t_{ILOCOWT}$	—	142	190	µs	Figure 56.7

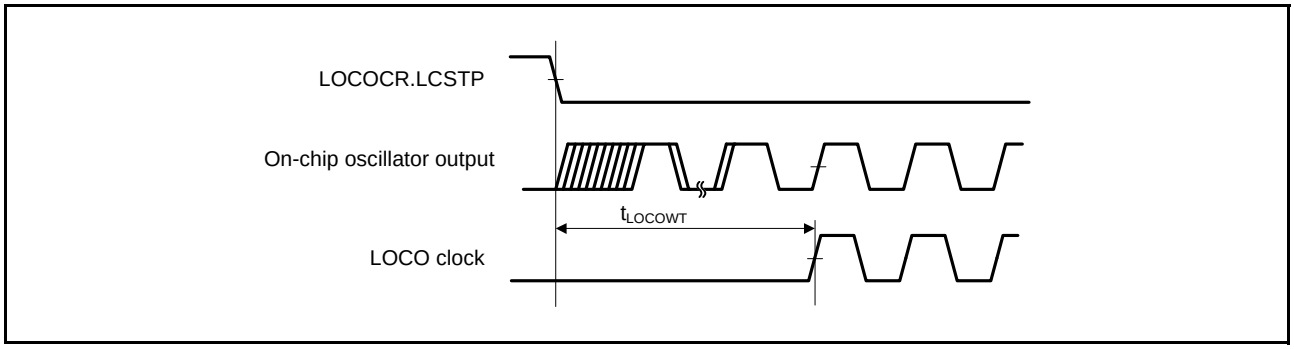


Figure 56.6 LOCO Clock Oscillation Start Timing

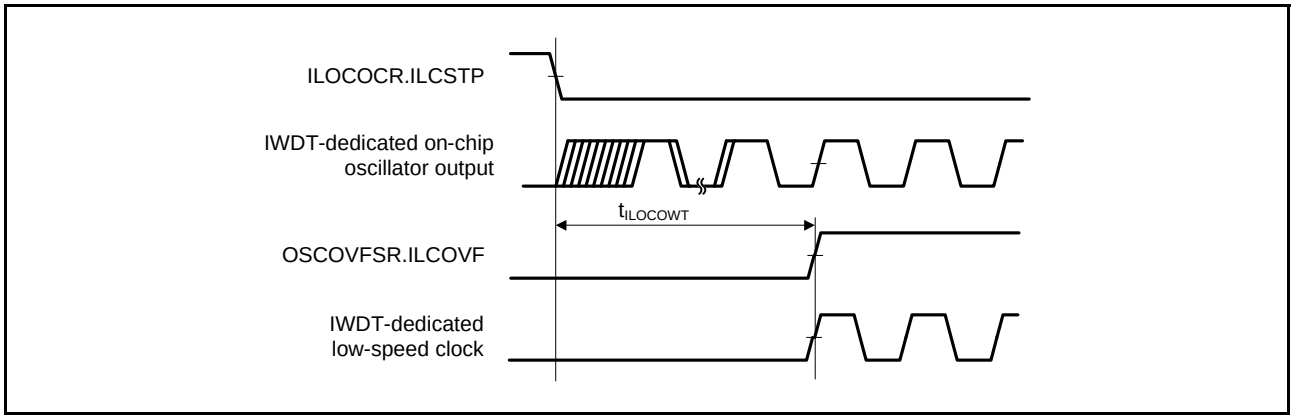


Figure 56.7 IWDT-dedicated Low-Speed Clock Oscillation Start Timing

Table 56.19 HOCO Clock Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
$T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
HOCO clock oscillation frequency	FLL not in use	f_{HOCO}	15.616 (–2.40%)	16	16.384 (+2.40%)	MHz	–20°C ≤ T_a
			17.568 (–2.40%)	18	18.432 (+2.40%)		
			19.520 (–2.40%)	20	20.480 (+2.40%)		
			15.520 (–3.00%)	16	16.480 (+3.00%)		T_a < –20°C
			17.460 (–3.00%)	18	18.540 (+3.00%)		
			19.400 (–3.00%)	20	20.600 (+3.00%)		
	FLL in use	f_{HOCO}	15.960 (–0.25%)	16	16.040 (+0.25%)	MHz	Sub-clock frequency precision: ±50 ppm
			17.955 (–0.25%)	18	18.045 (+0.25%)		
			19.950 (–0.25%)	20	20.050 (+0.25%)		
HOCO clock oscillation stabilization waiting time		t_{HOCOWT}	—	105	149	µs	Figure 56.8
HOCO clock power supply stabilization time		t_{HOCOP}	—	—	150	µs	Figure 56.9
FLL stabilization waiting time		t_{FLLWT}	—	—	1.8	ms	

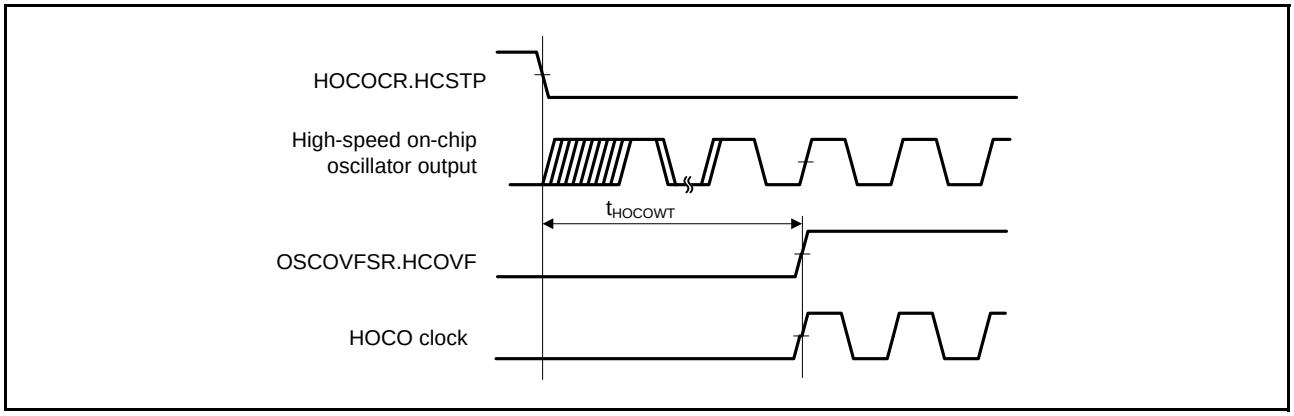


Figure 56.8 HOCO Clock Oscillation Start Timing (Oscillation is Started by Setting the HOCOCR.HCSTP Bit)

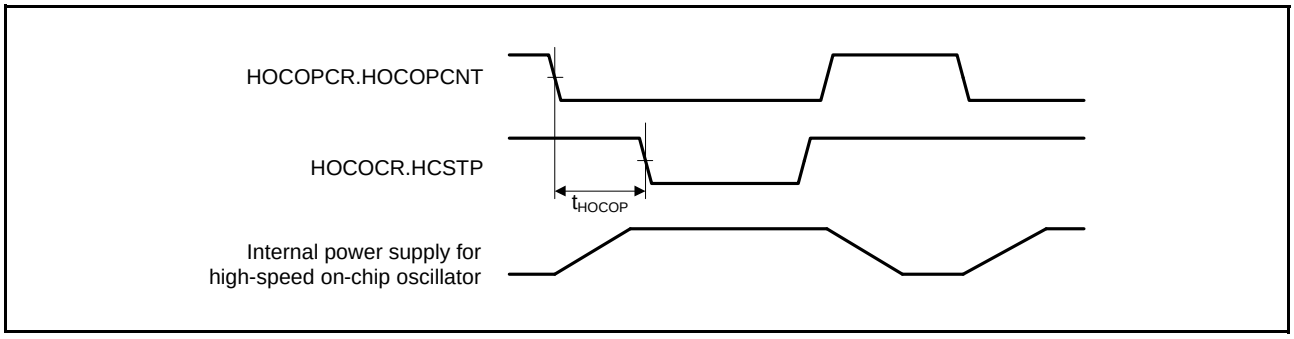


Figure 56.9 High-Speed On-Chip Oscillator Power Supply Control Timing

Table 56.20 PLL Clock Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
$T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
PLL clock oscillation frequency	f_{PLL}	120	—	240	MHz	
PLL clock oscillation stabilization waiting time	t_{PLLWT}	—	259	320	µs	Figure 56.10

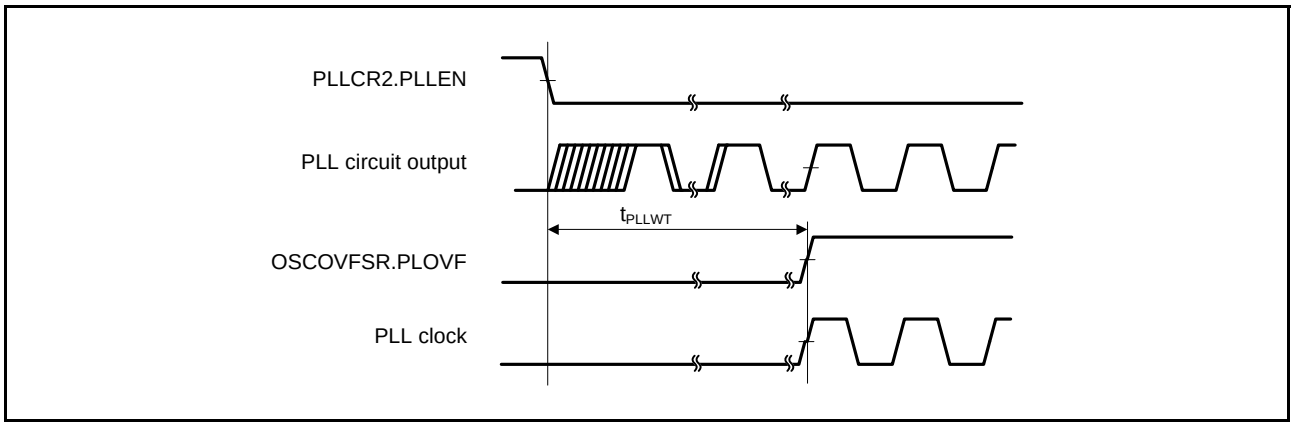


Figure 56.10 PLL Clock Oscillation Start Timing

Table 56.21 Sub-Clock Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
When a low C_L crystal resonator is in use: V_{BATT} = 2.0 to 3.6 V,
When a standard C_L crystal resonator is in use: V_{BATT} = 1.62 to 3.6 V, $T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Sub-clock oscillation frequency	f_{SUB}	—	32.768	—	kHz	
Sub-clock oscillation stabilization time	t_{SUBOSC}	—	—	*1	s	Figure 56.11
Sub-clock oscillation stabilization waiting time	$t_{SUBOSCWT}$	—	—	*2	s	

Note 1. When using a sub-clock, ask the manufacturer of the oscillator to evaluate its oscillation. Refer to the results of evaluation provided by the manufacturer for the oscillation stabilization time.

Note 2. The number of cycles selected by the value of the SOSCWTCR.SSTS[7:0] bits determines the sub-clock oscillation stabilization waiting time in accord with the formula below.
$t_{SUBOSCWT}$ = [(SSTS[7:0] bits × 16384) + 10] / f_{LOCO}

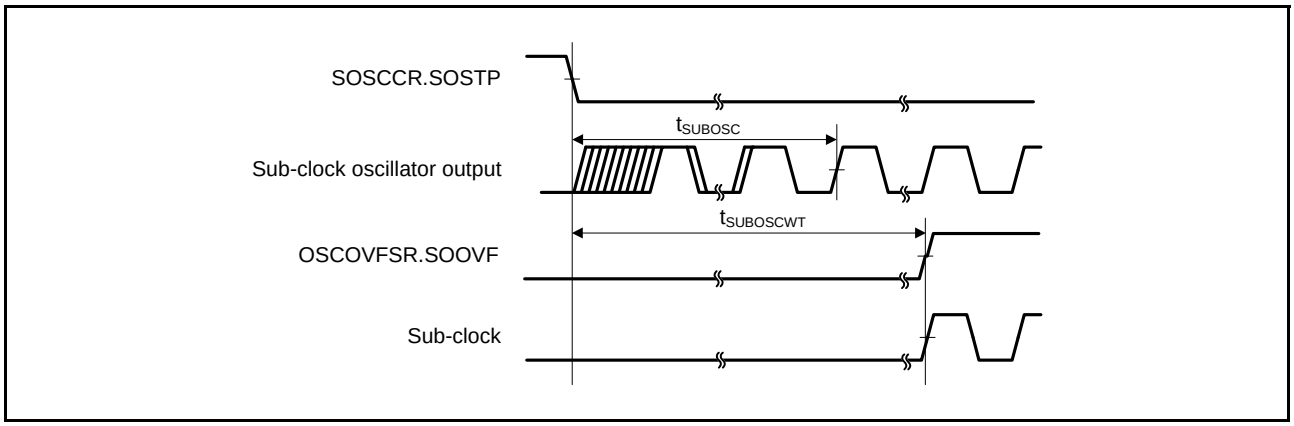


Figure 56.11 Sub-Clock Oscillation Start Timing

Table 56.22 CLKOUT Pin Output Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
$T_a = T_{opr}$,
High-drive output is selected by the drive capacity control register

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
CLKOUT pin output cycle time	t_{Ccyc}	25	—	—	ns	Figure 56.12 t_{Ccyc} = 25 ns
CLKOUT pin output high pulse width*1	t_{CH}	5	—	—	ns	
CLKOUT pin output low pulse width*1	t_{CL}	5	—	—	ns	
CLKOUT pin output rising time	t_{Cr}	—	—	5	ns	
CLKOUT pin output falling time	t_{Cf}	—	—	5	ns	

Note 1. If the main clock oscillator is selected by the CLKOUT output source select bit (CKOCR.CKOSEL[2:0]) and the external clock input is selected by the main clock oscillator switching bit (MOFCR.MOSEL), the pulse width depends on the input clock wave form.

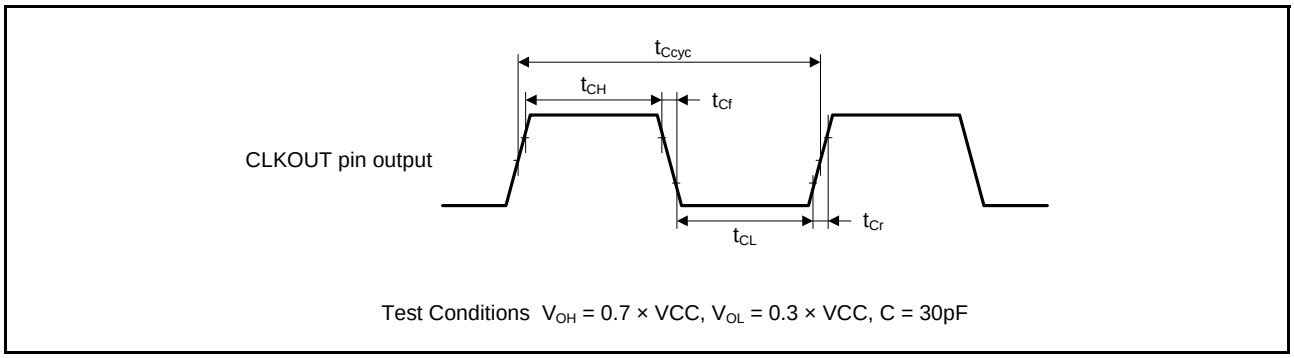


Figure 56.12 CLKOUT Pin Output Timing

56.4.3 Timing of Recovery from Low Power Consumption Modes

Table 56.23 Timing of Recovery from Low Power Consumption Modes (1)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
$T_a = T_{opr}$

Item			Symbol	Min.	Typ.	Max.		Unit	Test Conditions
						$t_{SBYOSCWT}$*2	t_{SBYSEQ}*3		
Recovery time from software standby mode *1	Crystal resonator connected to main clock oscillator	Main clock oscillator operating	t_{SBYMC}	—	—	{(MSTS[7:0] bit × 32) + 76} / 0.216	100 + 7 / f_{ICLK} + 2n / f_{MAIN}	µs	Figure 56.13
		Main clock oscillator and PLL circuit operating	t_{SBYPC}			{(MSTS[7:0] bit × 32) + 138} / 0.216	100 + 7 / f_{ICLK} + 2n / f_{PLL}		
	External clock input to main clock oscillator	Main clock oscillator operating	t_{SBYEX}			352	100 + 7 / f_{ICLK} + 2n / f_{EXMAIN}		
		Main clock oscillator and PLL circuit operating	t_{SBYPE}			639	100 + 7 / f_{ICLK} + 2n / f_{PLL}		
	Sub-clock oscillator operating		t_{SBYSC}			{(SSTS[7:0] bit × 16384) + 13} / 0.216 + 10 / f_{FCLK}	100 + 4 / f_{ICLK} + 2n / f_{SUE}		
	High-speed on-chip oscillator operating	High-speed on-chip oscillator operating	t_{SBYHO}			454	100 + 7 / f_{ICLK} + 2n / f_{HOCO}		
		High-speed on-chip oscillator operating and PLL circuit operating	t_{SBYPH}			741	100 + 7 / f_{ICLK} + 2n / f_{PLL}		
	Low-speed on-chip oscillator operating*4		t_{SBYLO}			338	100 + 7 / f_{ICLK} + 2n / f_{LOCO}		

Note 1. The time for recovery from software standby mode is determined by the value obtained by adding the oscillation stabilization waiting time ($t_{SBYOSCWT}$) and the time required for operations by the software standby release sequencer (t_{SBYSEQ}).
Note 2. When several oscillators were running before the transition to software standby, the greatest value of the oscillation stabilization waiting time $t_{SBYOSCWT}$ is selected.
Note 3. For n, the greatest value is selected from among the internal clock division settings.
Note 4. This condition applies when f_{ICLK}:f_{FCLK} = 1:1, 2:1, or 4:1.

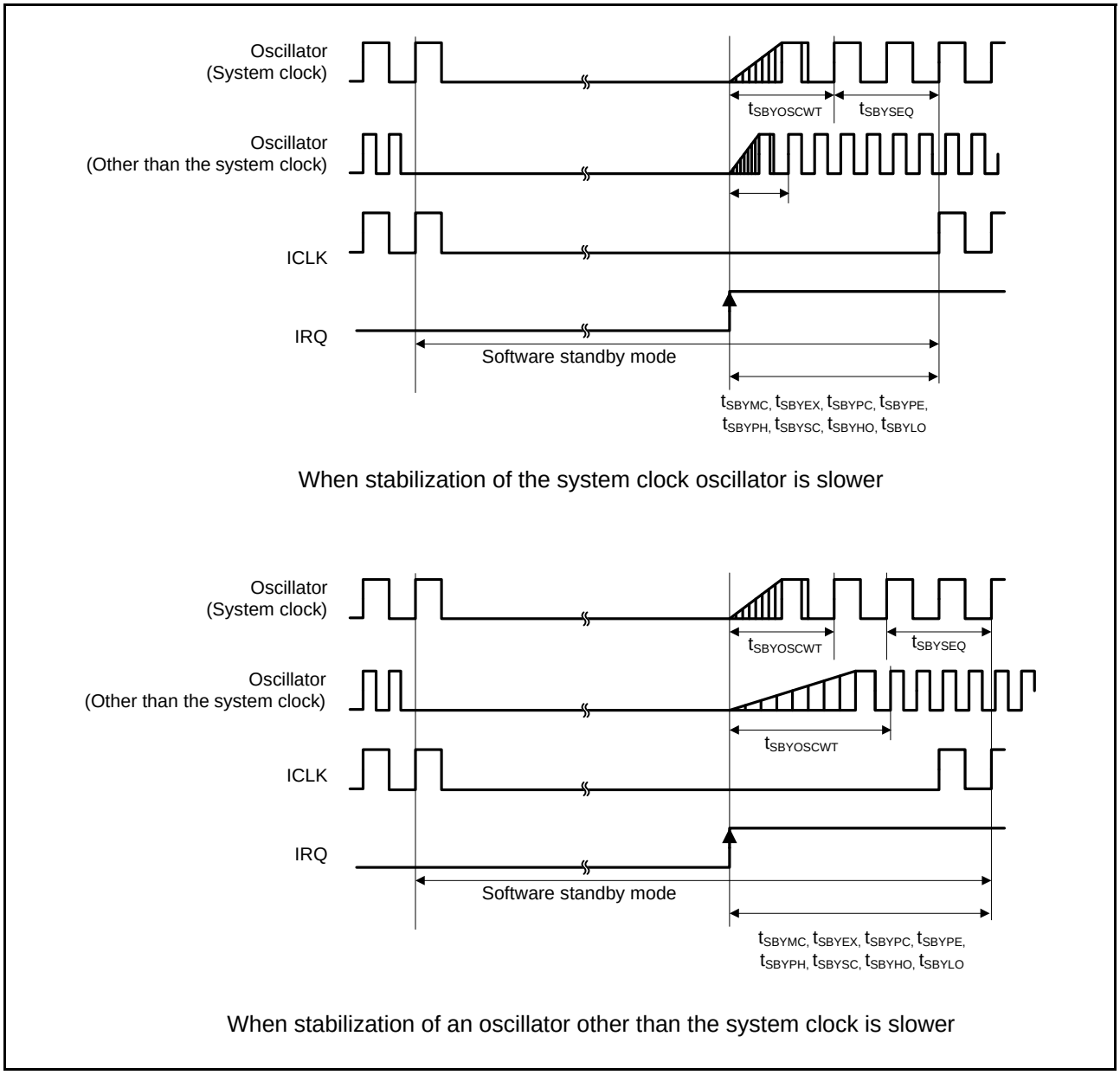


Figure 56.13 Software Standby Mode Recovery Timing

Table 56.24 Timing of Recovery from Low Power Consumption Modes (2)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
$T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Recovery time from deep software standby mode	t_{DSBY}	—	—	0.9	ms	Figure 56.14
Waiting time after recovery from deep software standby mode	t_{DSBYWT}	23	—	24	t_{Lcyc}	

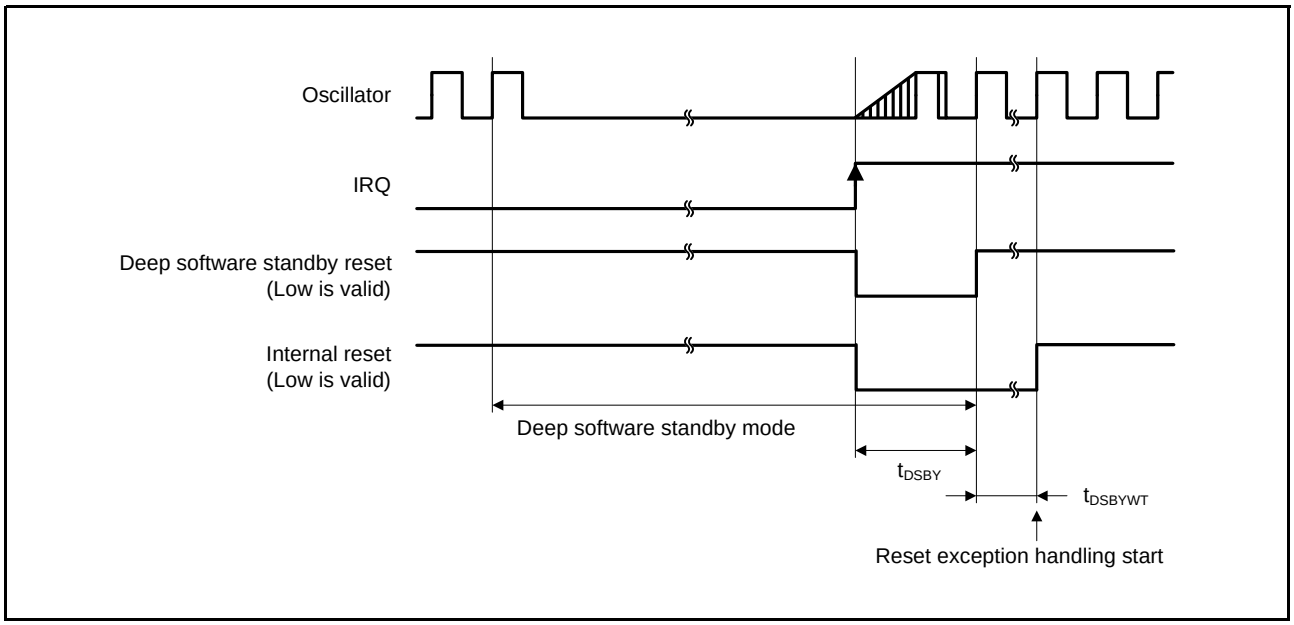


Figure 56.14 Deep Software Standby Mode Recovery Timing

56.4.4 Control Signal Timing

Table 56.25 Control Signal Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKB = 8 to 60 MHz, $T_a = T_{opr}$

Item	Symbol	Min.*1	Typ.	Max.	Unit	Test Conditions*1
NMI pulse width	t_{NMIW}	200	—	—	ns	t_{PBcyc} × 2 ≤ 200 ns, Figure 56.15
		t_{PBcyc} × 2	—	—		t_{PBcyc} × 2 > 200 ns, Figure 56.15
IRQ pulse width	t_{IRQW}	200	—	—	ns	t_{PBcyc} × 2 ≤ 200 ns, Figure 56.16
		t_{PBcyc} × 2	—	—		t_{PBcyc} × 2 > 200 ns, Figure 56.16

Note 1. t_{PBcyc}: PCLKB cycle

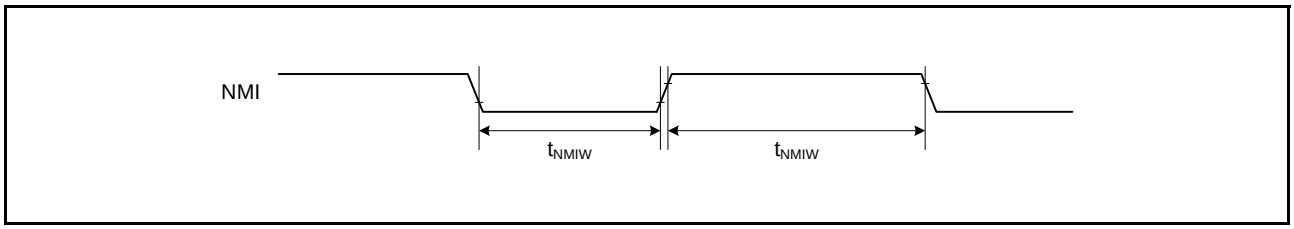


Figure 56.15 NMI Interrupt Input Timing

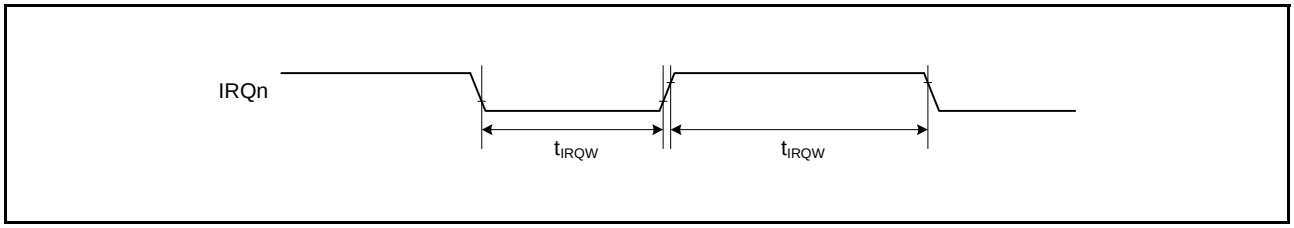


Figure 56.16 IRQ Interrupt Input Timing

56.4.5 Bus Timing

Table 56.26 Bus Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
ICLK = PCLKA = 8 to 120 MHz, PCLKB = BCLK = SDCLK = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the drive capacity control register.

Item	Symbol	Min.	Max.	Unit	Test Conditions
Address delay time	t_{AD}	—	12.5	ns	Figure 56.17 to Figure 56.22
Byte control delay time	t_{BCD}	—	12.5	ns	
CS# delay time	t_{CSD}	—	12.5	ns	
ALE delay time	t_{ALED}	—	12.5	ns	
RD# delay time	t_{RSD}	—	12.5	ns	
Read data setup time	t_{RDS}	12.5	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
WR# delay time	t_{WRD}	—	12.5	ns	
Write data delay time	t_{WDD}	—	12.5	ns	
Write data hold time	t_{WDH}	0	—	ns	
WAIT# setup time	t_{WTS}	12.5	—	ns	Figure 56.23
WAIT# hold time	t_{WTH}	0	—	ns	
Address delay time 2 (SDRAM)	t_{AD2}	1	12.5	ns	Figure 56.24 to Figure 56.30
CS# delay time 2 (SDRAM)	t_{CSD2}	1	12.5	ns	
DQM delay time (SDRAM)	t_{DQMD}	1	12.5	ns	
CKE delay time (SDRAM)	t_{CKED}	1	12.5	ns	
Read data setup time 2 (SDRAM)	t_{RDS2}	10	—	ns	
Read data hold time 2 (SDRAM)	t_{RDH2}	0	—	ns	
Write data delay time 2 (SDRAM)	t_{WDD2}	—	12.5	ns	
Write data hold time 2 (SDRAM)	t_{WDH2}	1	—	ns	
WE# delay time (SDRAM)	t_{WED}	1	12.5	ns	
RAS# delay time (SDRAM)	t_{RASD}	1	12.5	ns	
CAS# delay time (SDRAM)	t_{CASD}	1	12.5	ns	

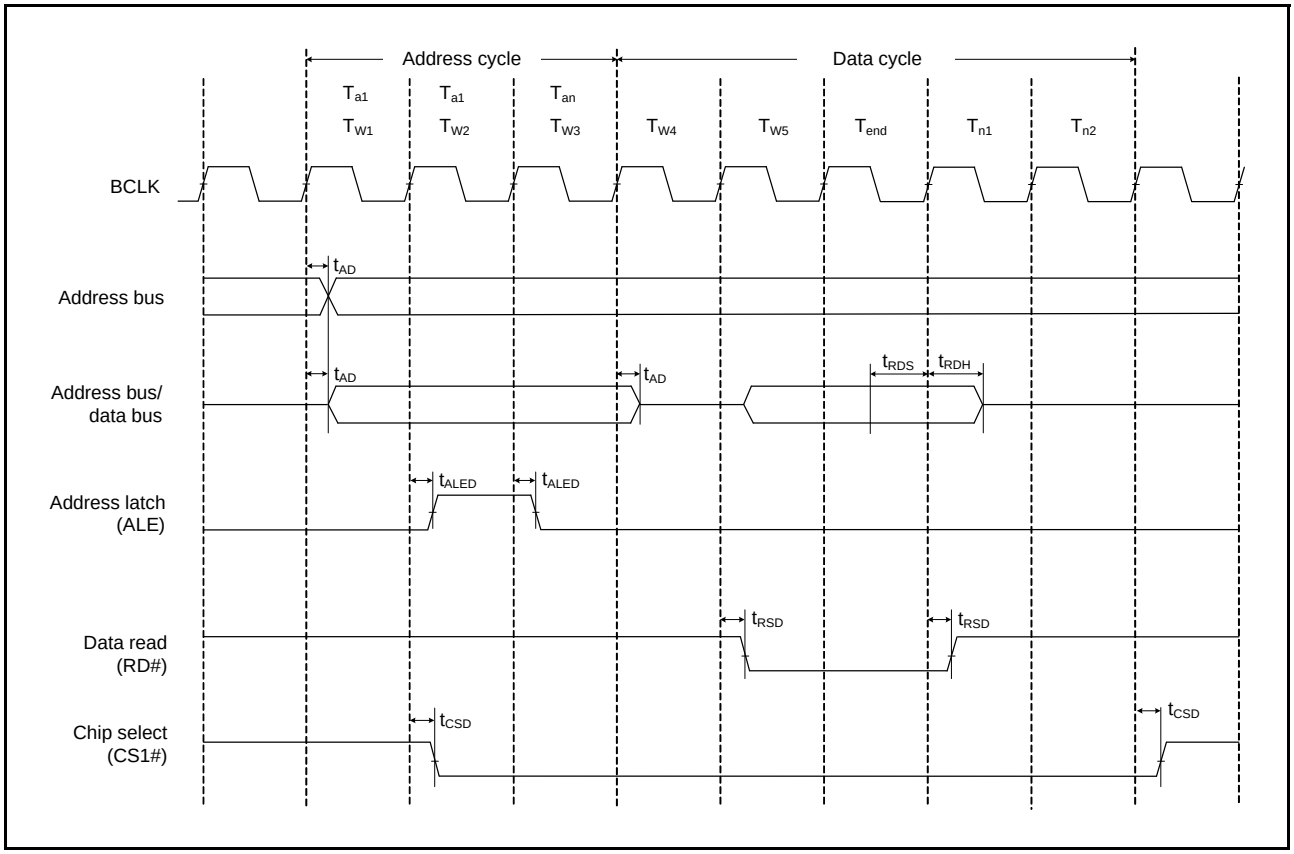


Figure 56.17 Address/Data Multiplexed Bus Read Access Timing

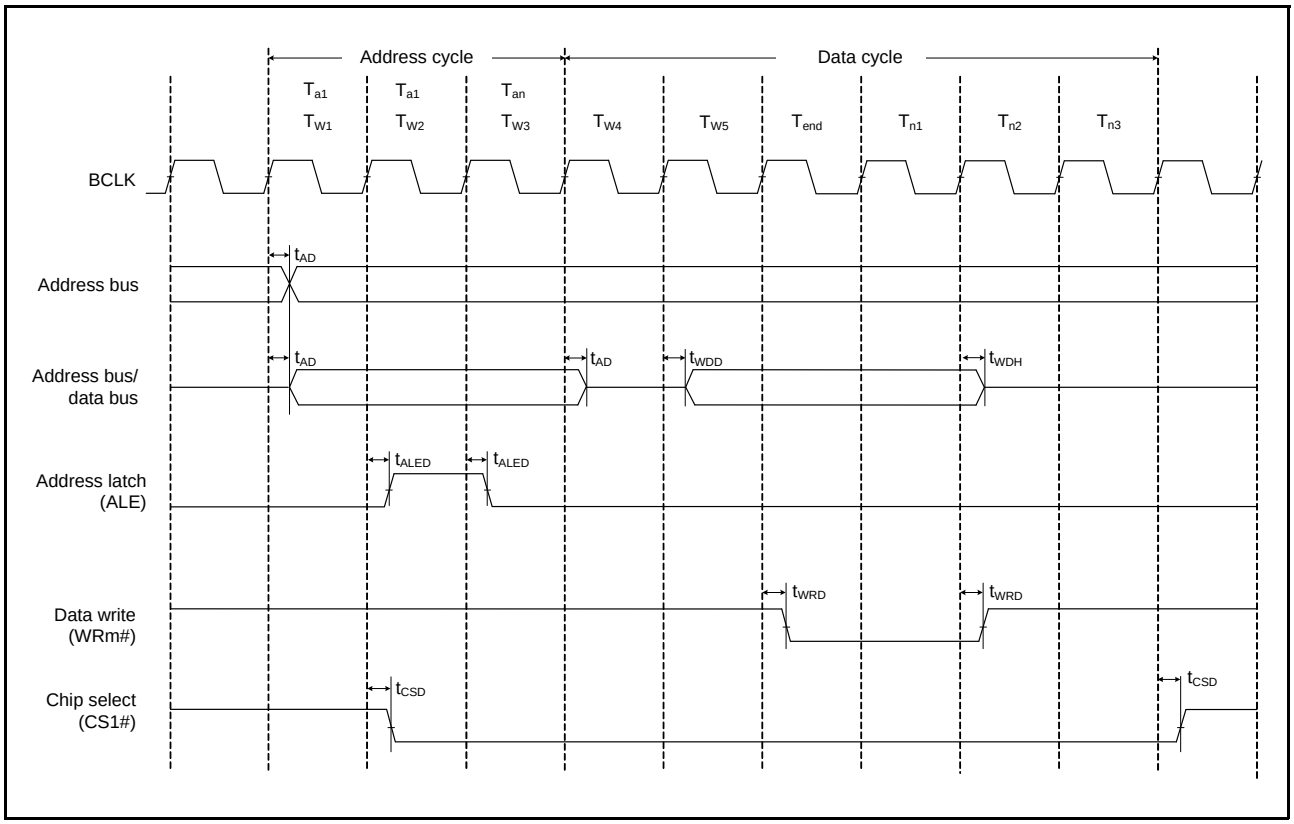


Figure 56.18 Address/Data Multiplexed Bus Write Access Timing

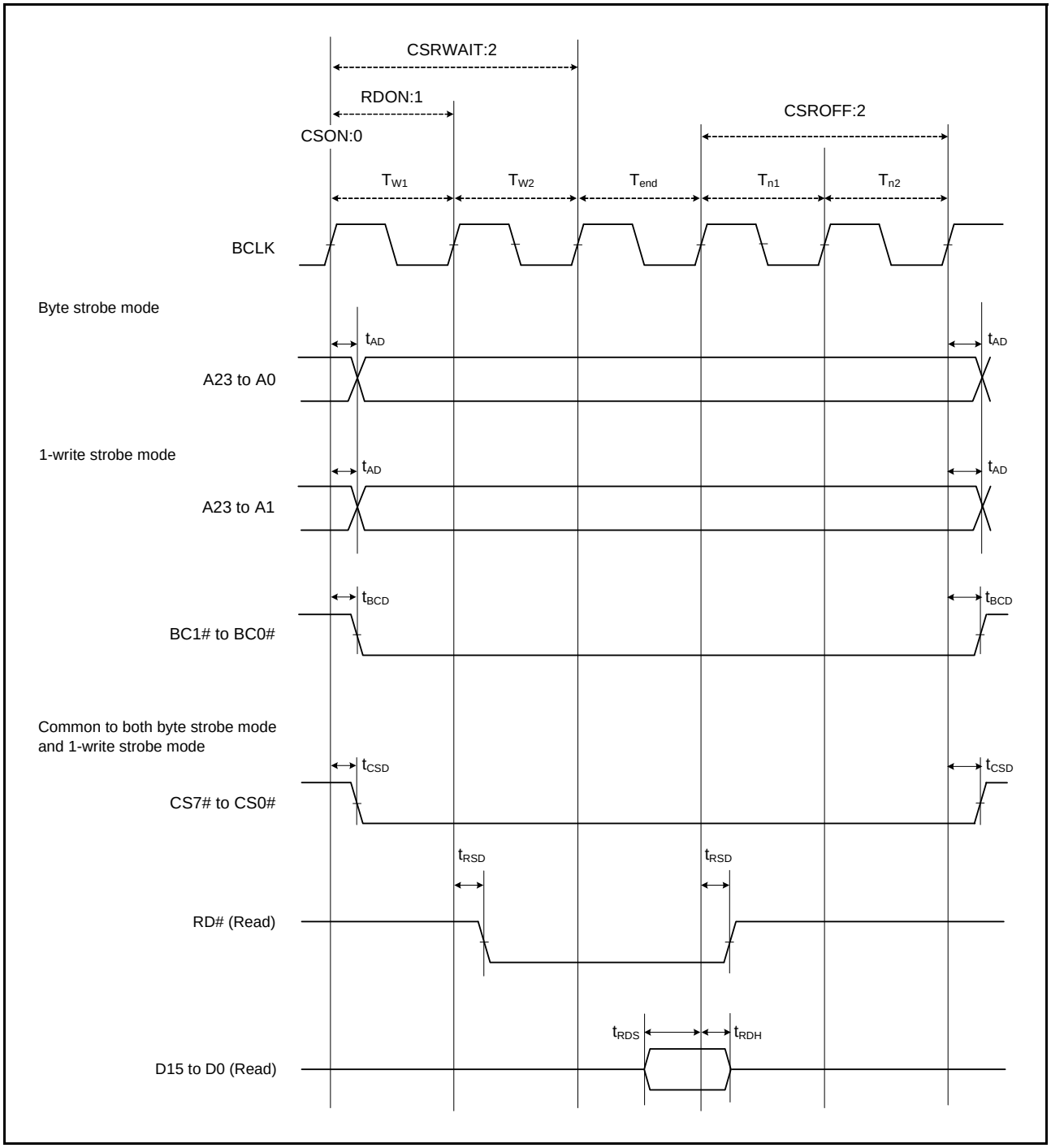
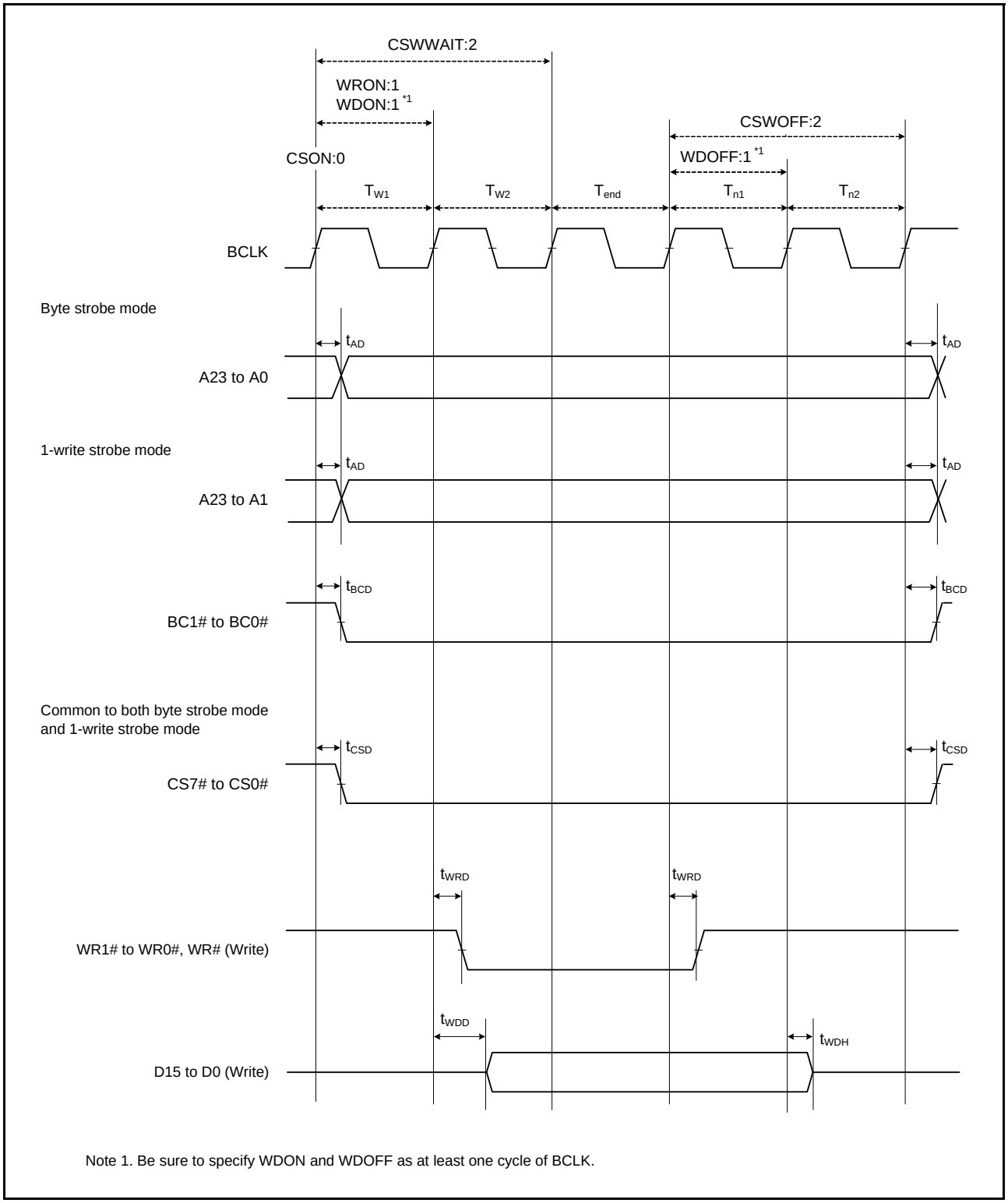


Figure 56.19 External Bus Timing/Normal Read Cycle (Bus Clock Synchronized)



Note 1. Be sure to specify WDON and WDOFF as at least one cycle of BCLK.

Figure 56.20 External Bus Timing/Normal Write Cycle (Bus Clock Synchronized)

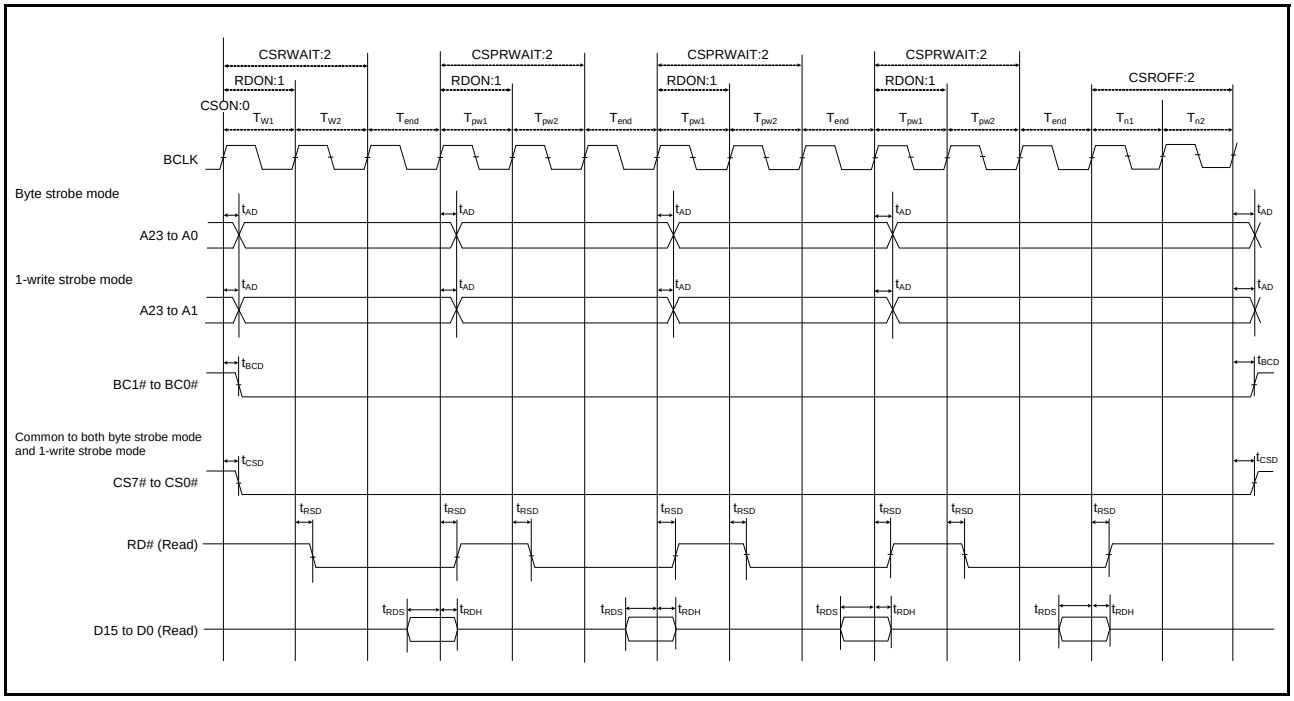


Figure 56.21 External Bus Timing/Page Read Cycle (Bus Clock Synchronized)

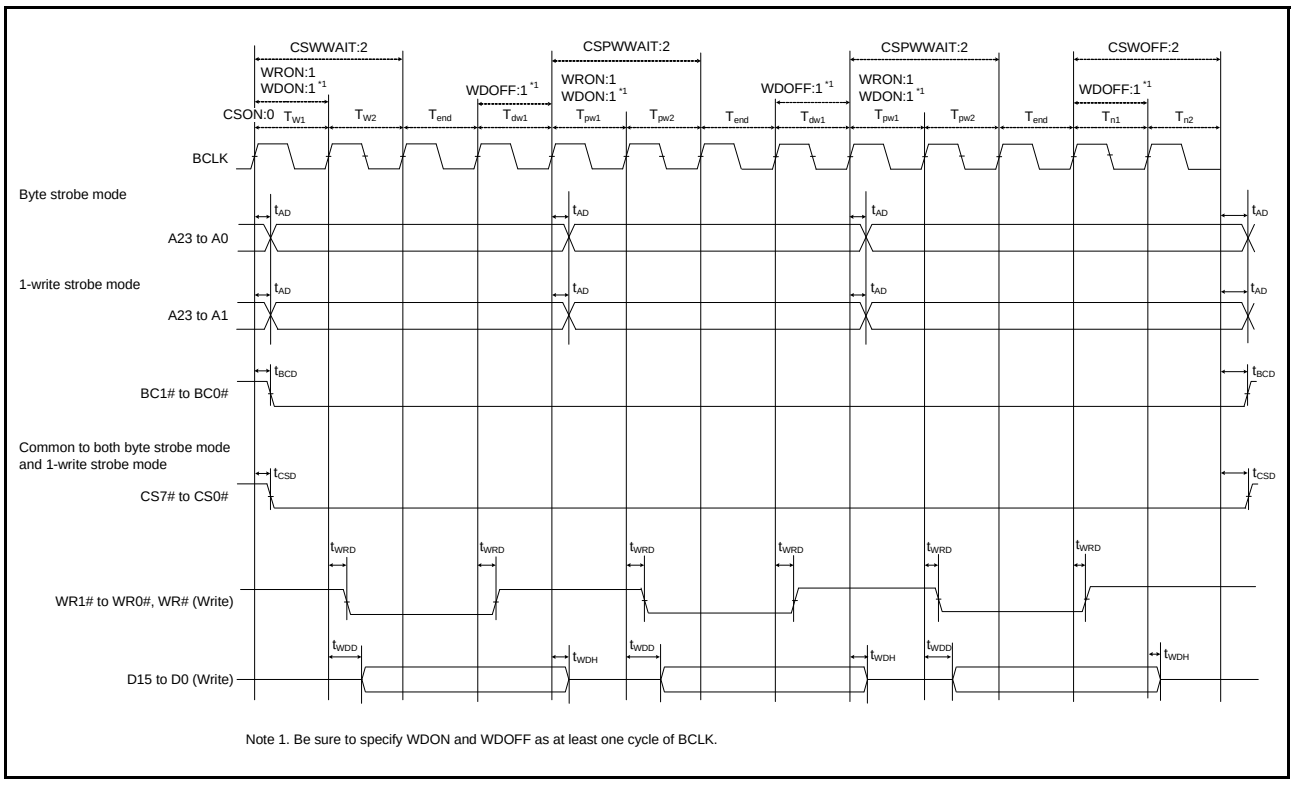


Figure 56.22 External Bus Timing/Page Write Cycle (Bus Clock Synchronized)

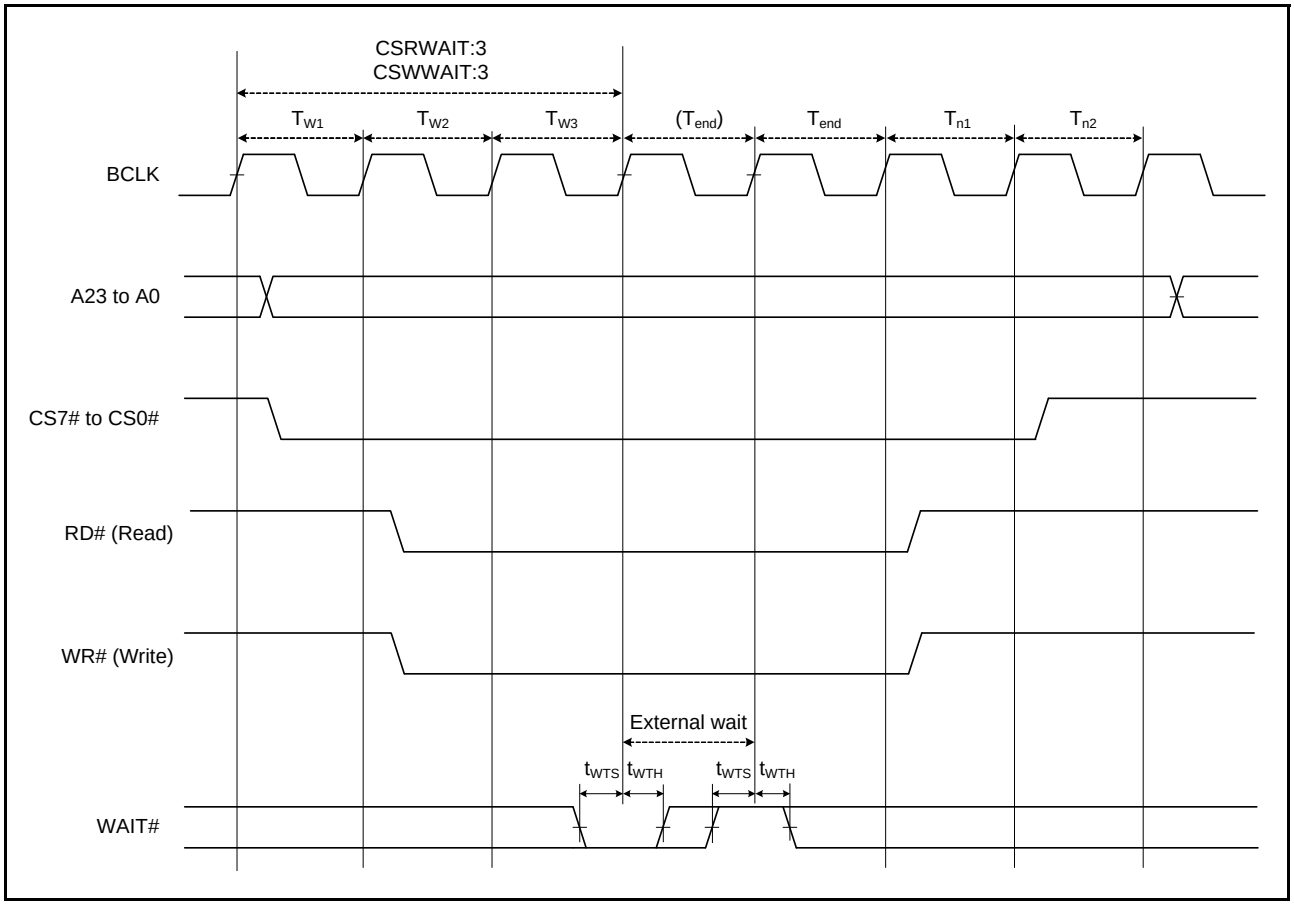
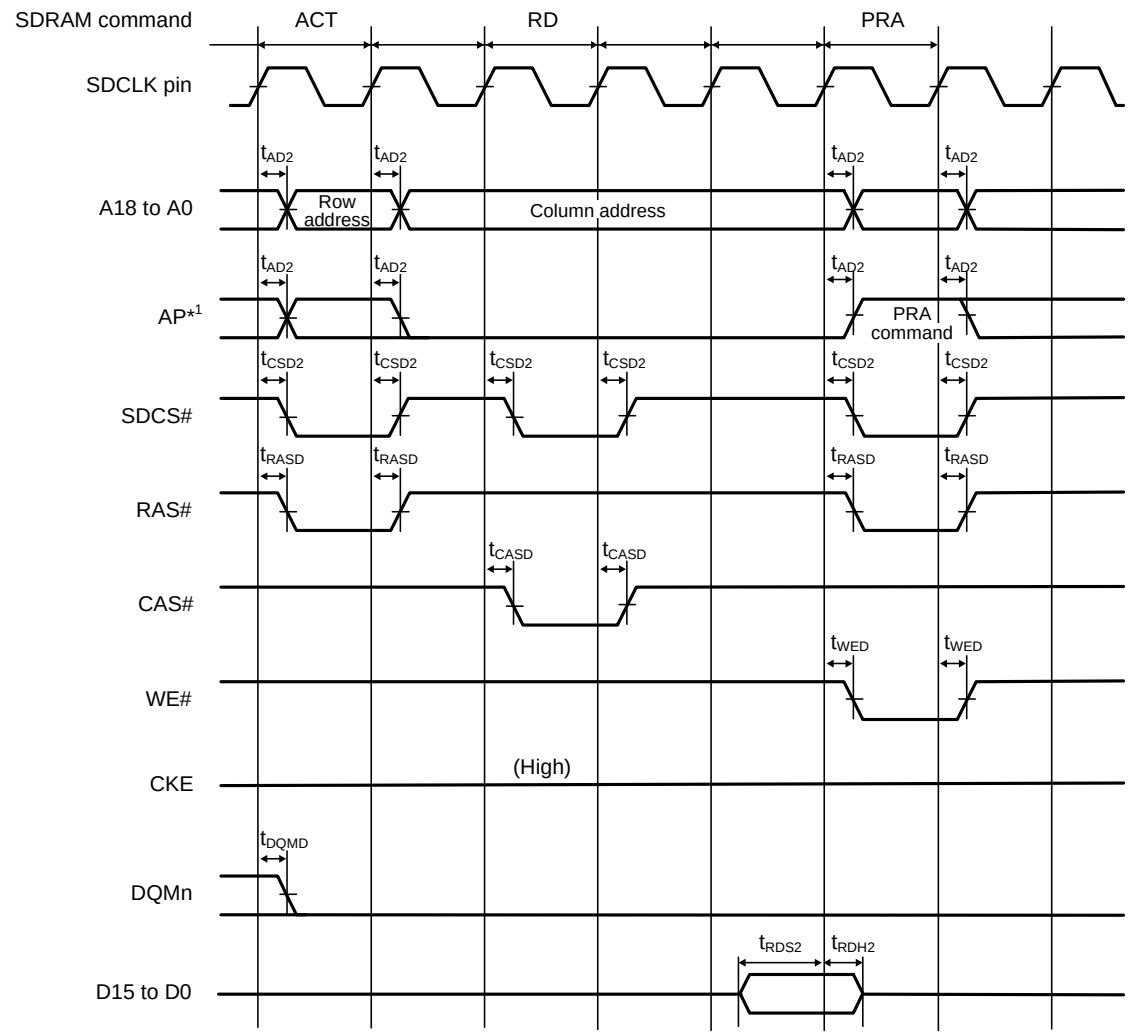
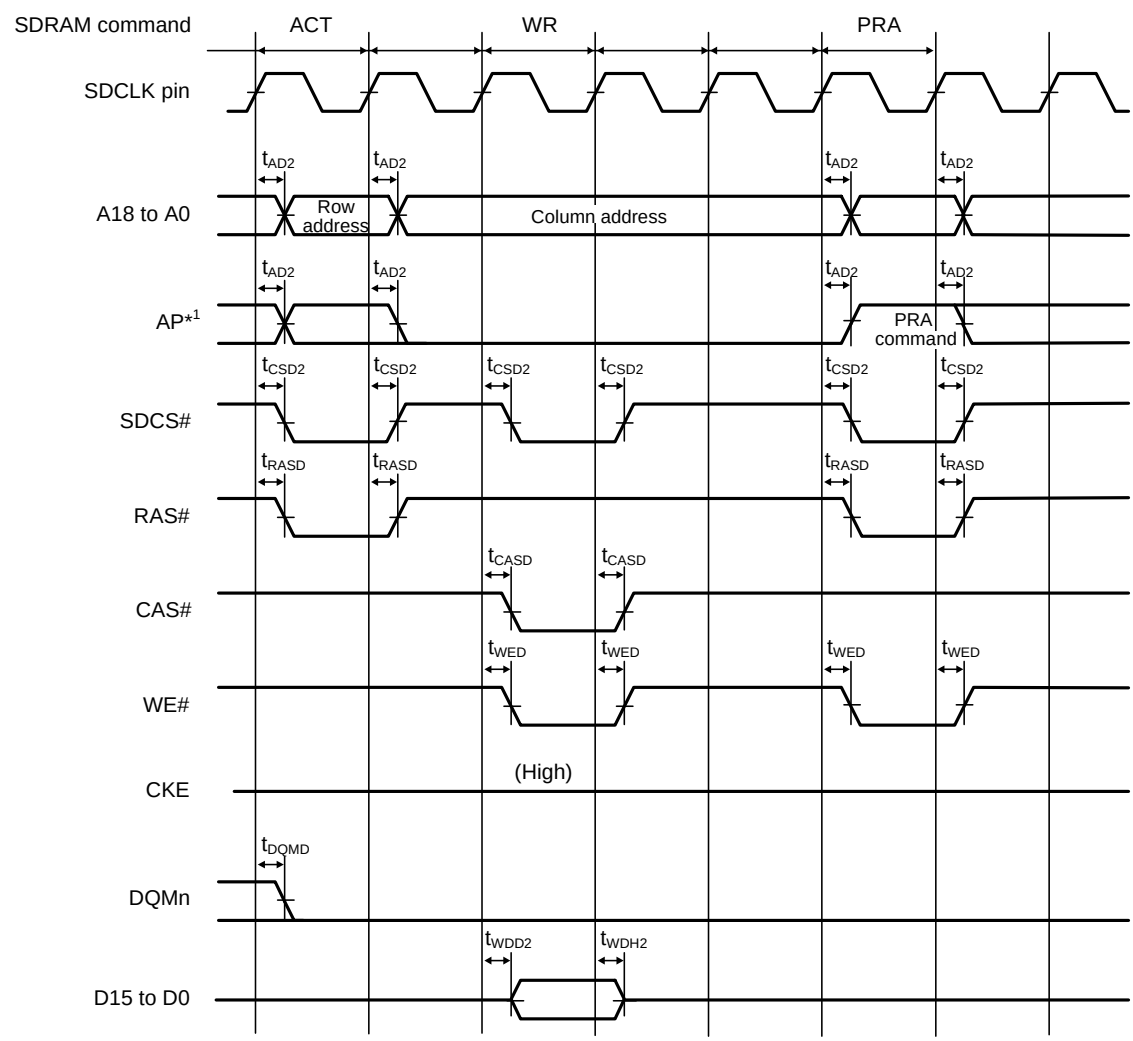


Figure 56.23 External Bus Timing/External Wait Control



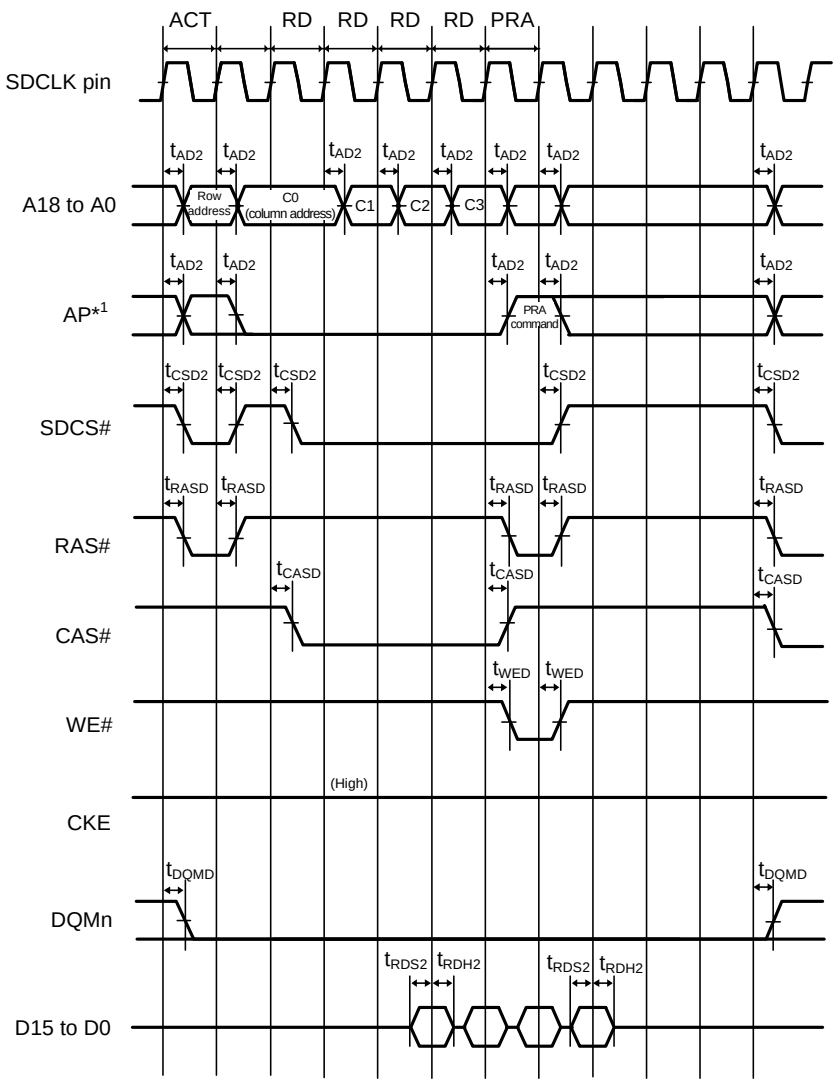
Note 1. Address pins for output of the precharge-setting command (Precharge-sel) for SDRAM.

Figure 56.24 SDRAM Space Single Read Bus Timing



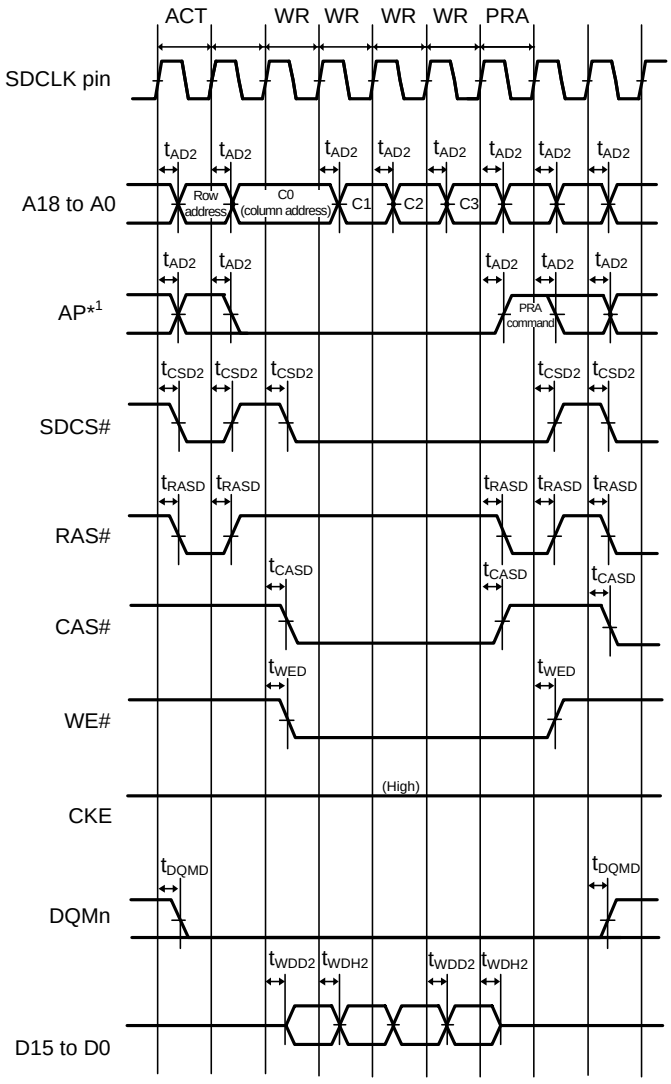
Note 1. Address pins for output of the precharge-setting command (Precharge-sel) for SDRAM.

Figure 56.25 SDRAM Space Single Write Bus Timing



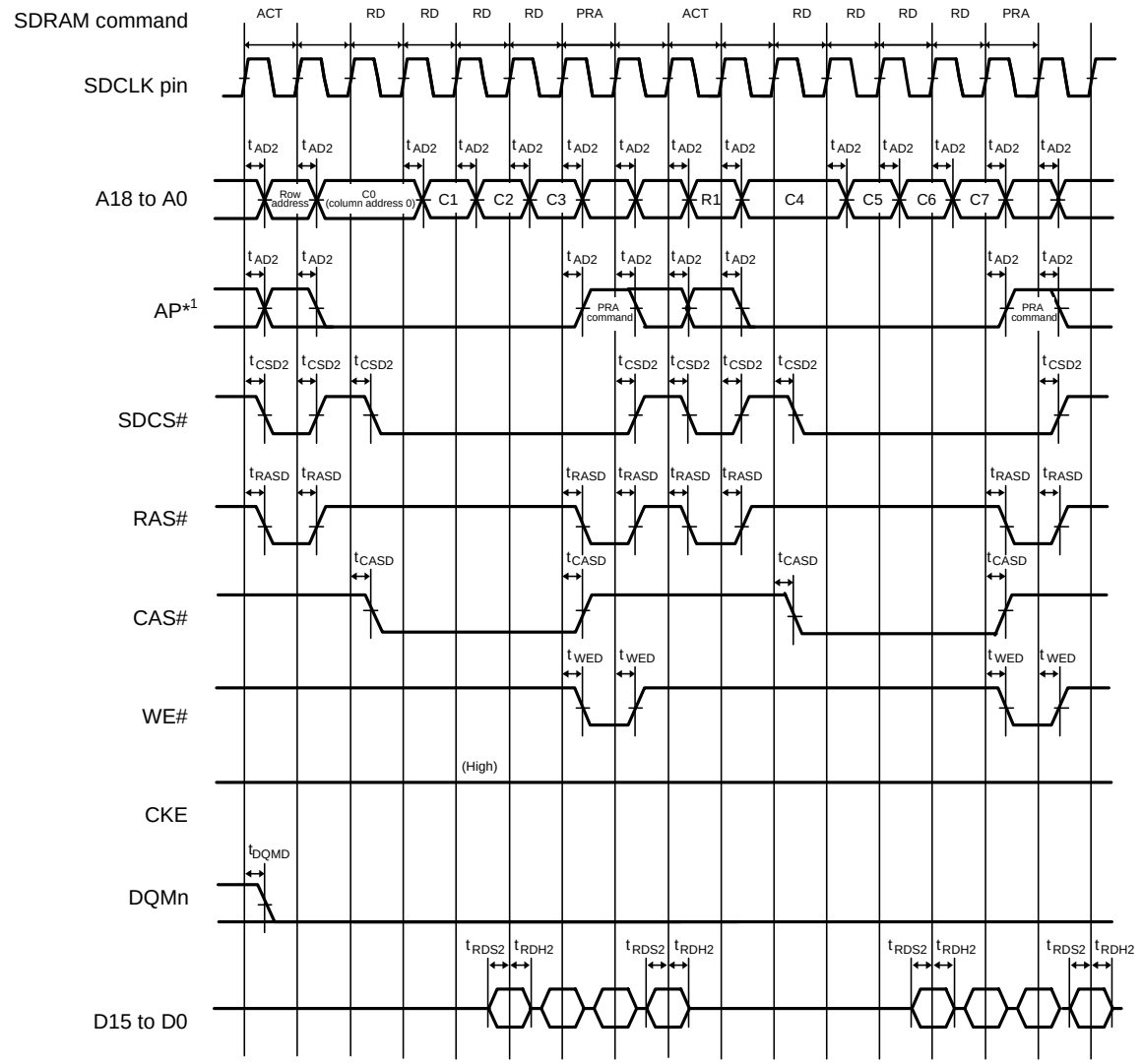
Note 1. Address pins for output of the precharge-setting command (Precharge-sel) for SDRAM.

Figure 56.26 SDRAM Space Multiple Read Bus Timing



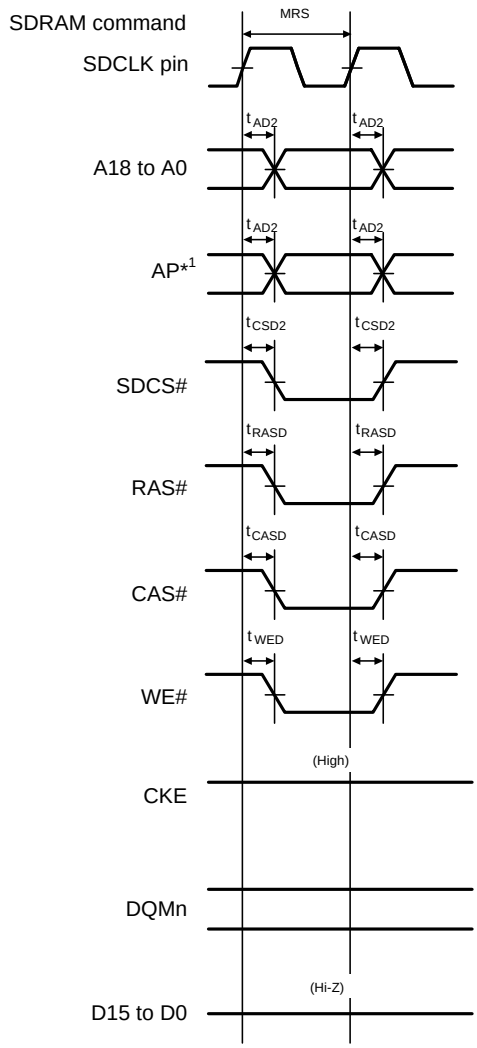
Note 1. Address pins for output of the precharge-setting command (Precharge-sel) for SDRAM.

Figure 56.27 SDRAM Space Multiple Write Bus Timing



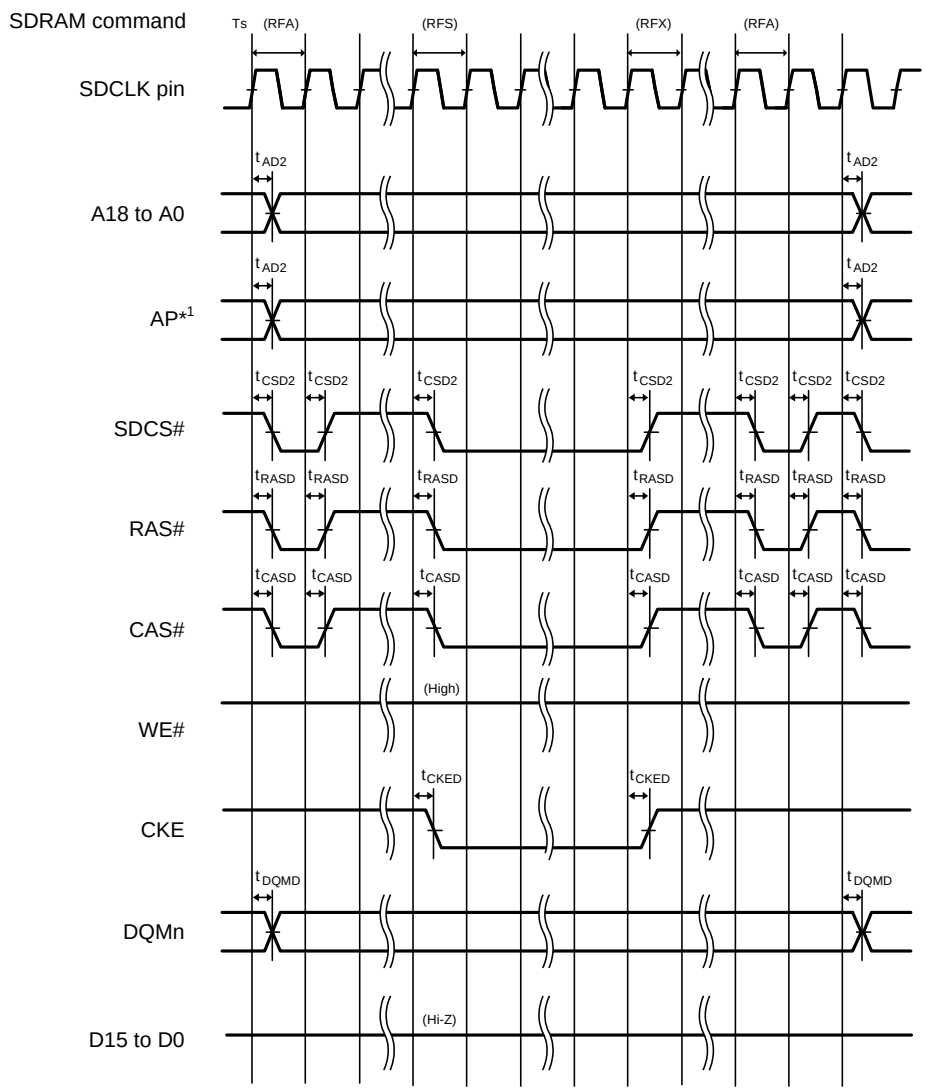
Note 1. Address pins for output of the precharge-setting command (Precharge-sel) for SDRAM.

Figure 56.28 SDRAM Space Multiple Read Line Stride Bus Timing



Note 1. Address pins for output of the precharge-setting command (Precharge-sel) for SDRAM.

Figure 56.29 SDRAM Space Mode Register Set Bus Timing



Note 1. Address pins for output of the precharge-setting command (Precharge-sel) for SDRAM.

Figure 56.30 SDRAM Space Self-Refresh Bus Timing

56.4.6 EXDMAC Timing

Table 56.27 EXDMAC Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
ICLK = PCLKA = 8 to 120 MHz, PCLKB = BCLK = SDCLK = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the drive capacity control register.

Item		Symbol	Min.	Max.	Unit	Test Conditions
EXDMAC	EDREQ setup time	t_{EDRQS}	13	—	ns	Figure 56.31
	EDREQ hold time	t_{EDRQH}	2	—	ns	
	EDACK delay time	t_{EDACD}	—	13	ns	Figure 56.32, Figure 56.33

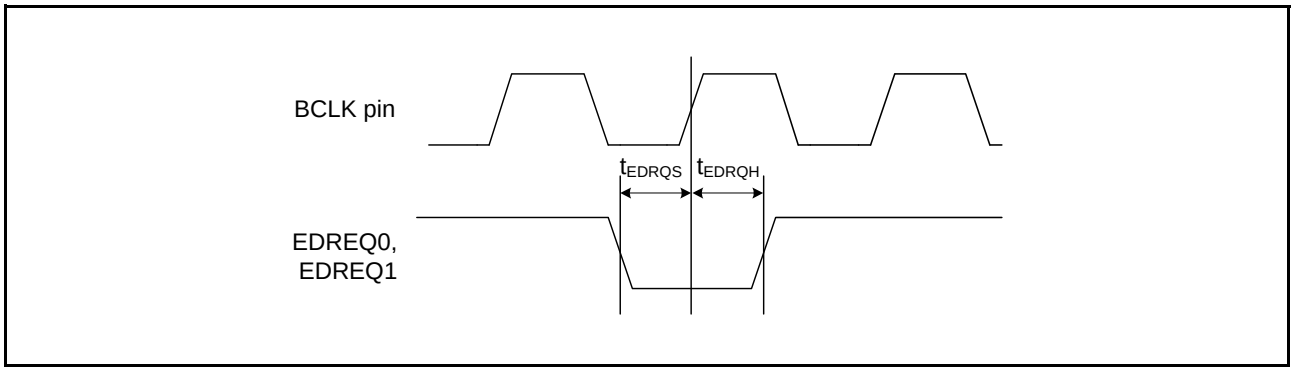


Figure 56.31 EDREQ0 and EDREQ1 Input Timing

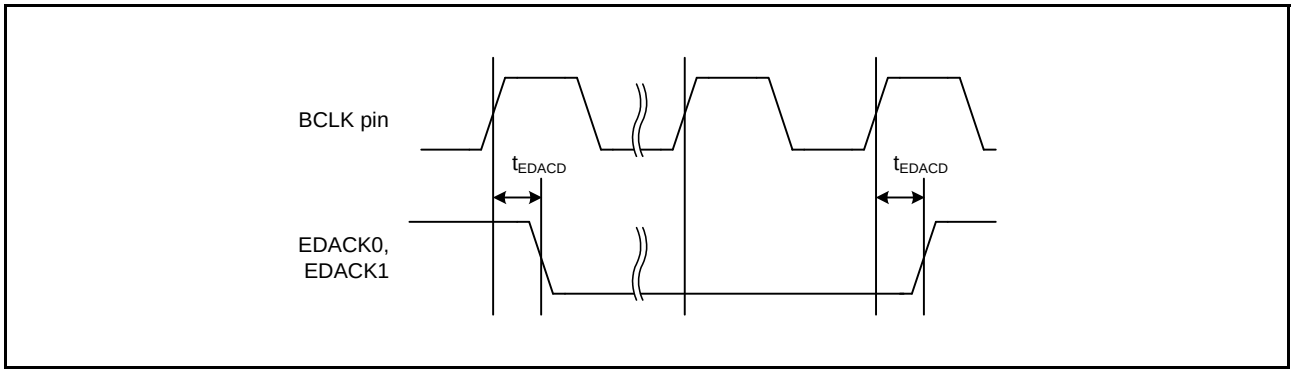


Figure 56.32 EDACK0 and EDACK1 Single-Address Transfer Timing (for a CS Area)

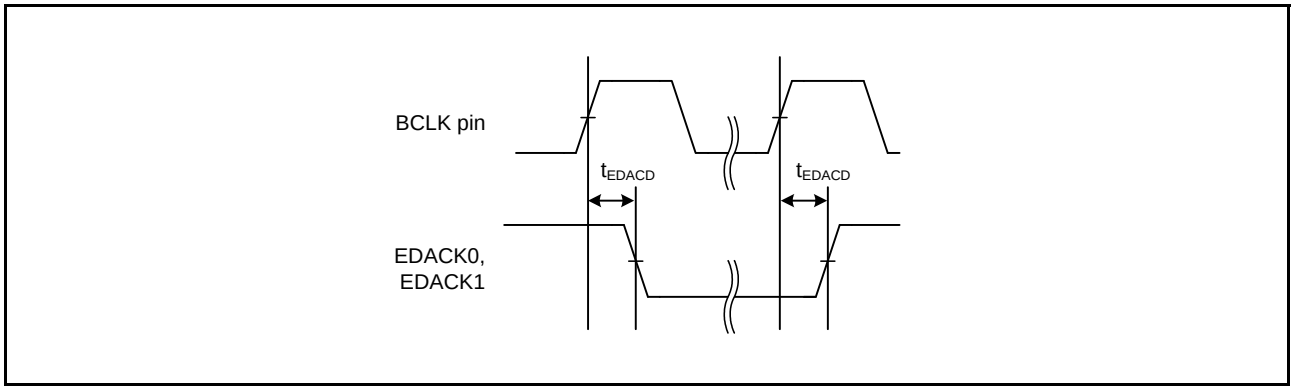


Figure 56.33 EDACK0 and EDACK1 Single-Address Transfer Timing (for SDRAM)

56.4.7 Timing of On-Chip Peripheral Modules

56.4.7.1 I/O Port

Table 56.28 I/O Port Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the drive capacity control register.

Item		Symbol	Min.	Max.	Unit*1	Test Conditions
I/O ports	Input data pulse width	t_{PRW}	1.5	—	t_{PBcyc}	Figure 56.34

Note 1. t_{PBcyc}: PCLKB cycle

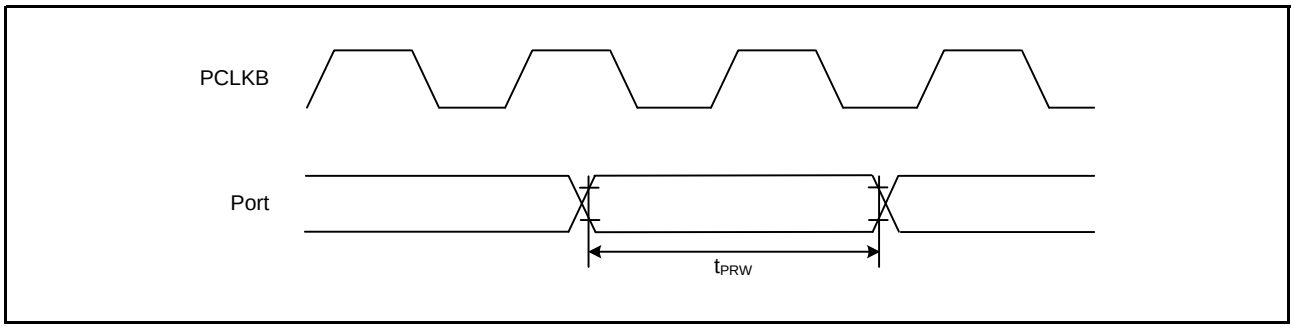


Figure 56.34 I/O Port Input Timing

56.4.7.2 TPU

Table 56.29 TPU Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the drive capacity control register.

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
TPU	Input capture input pulse width	Single-edge setting	t_{TICW}	1.5	—	t_{PBcyc}	Figure 56.35
		Both-edge setting		2.5	—		
	Timer clock pulse width	Single-edge setting	t_{TCKWH}, t_{TCKWL}	1.5	—	t_{PBcyc}	Figure 56.36
		Both-edge setting		2.5	—		
		Phase counting mode		2.5	—		

Note 1. t_{PBcyc}: PCLKB cycle

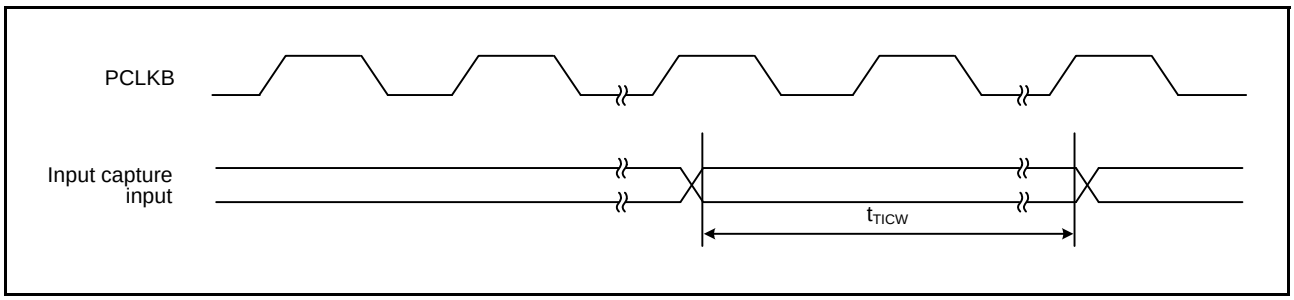


Figure 56.35 TPU Input Capture Input Timing

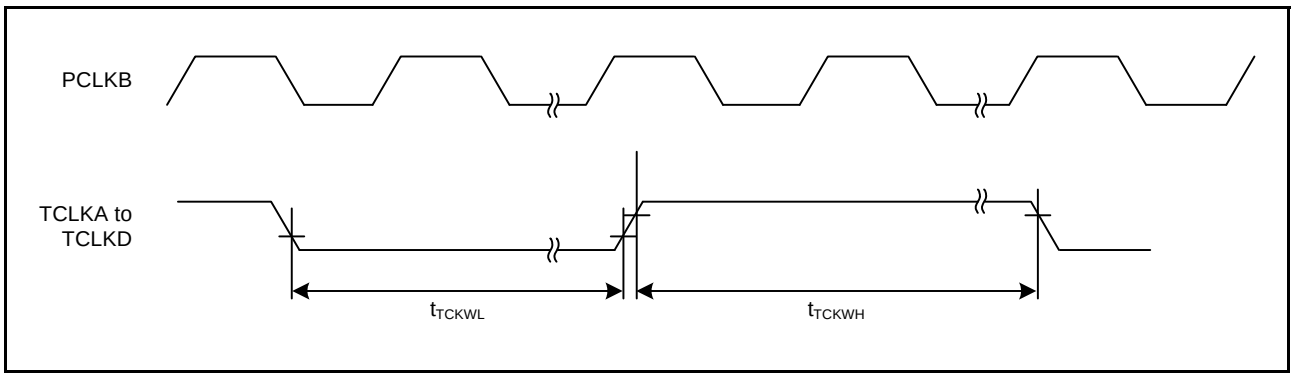


Figure 56.36 TPU Clock Input Timing

56.4.7.3 TMR

Table 56.30 TMR Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the drive capacity control register.

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
TMR	Timer clock pulse width	Single-edge setting	t_{TMCWH}, t_{TMCWL}	1.5	—	t_{PBcyc}	Figure 56.37
		Both-edge setting		2.5	—		

Note 1. t_{PBcyc}: PCLKB cycle

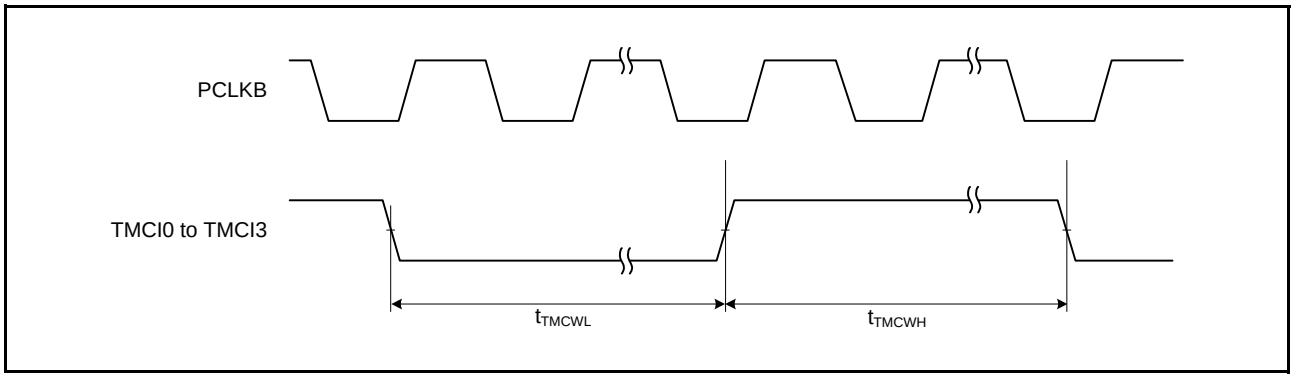


Figure 56.37 TMR Clock Input Timing

56.4.7.4 CMTW

Table 56.31 CMTW Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the drive capacity control register.

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
CMTW	Input capture input pulse width	Single-edge setting	$t_{CMTWTICW}$	1.5	—	t_{PBcyc}	Figure 56.38
		Both-edge setting		2.5	—		

Note 1. t_{PBcyc}: PCLKB cycle

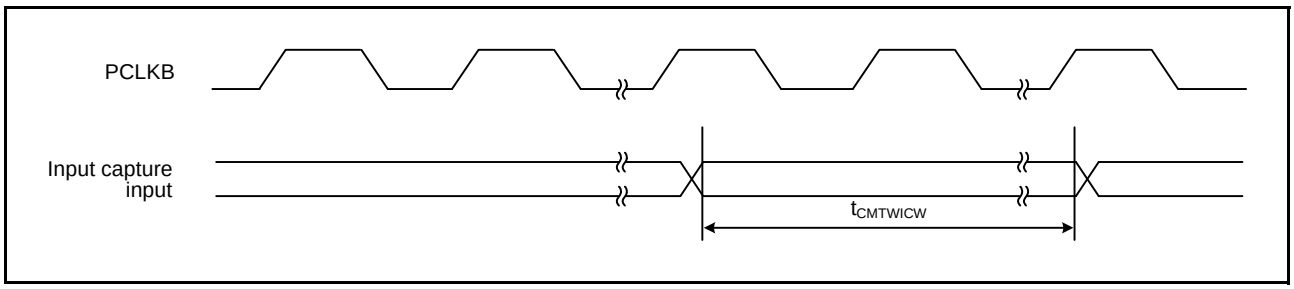


Figure 56.38 CMTW Input Capture Input Timing

56.4.7.5 MTU

Table 56.32 MTU Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the drive capacity control register.

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
MTU	Input capture input pulse width	Single-edge setting	t_{MTICW}	1.5	—	t_{PAcyc}	Figure 56.39
		Both-edge setting		2.5	—		
	Timer clock pulse width	Single-edge setting	t_{MTCKWH}, t_{MTCKWL}	1.5	—	t_{PAcyc}	Figure 56.40
		Both-edge setting		2.5	—		
		Phase counting mode		2.5	—		

Note 1. t_{PAcyc}: PCLKA cycle

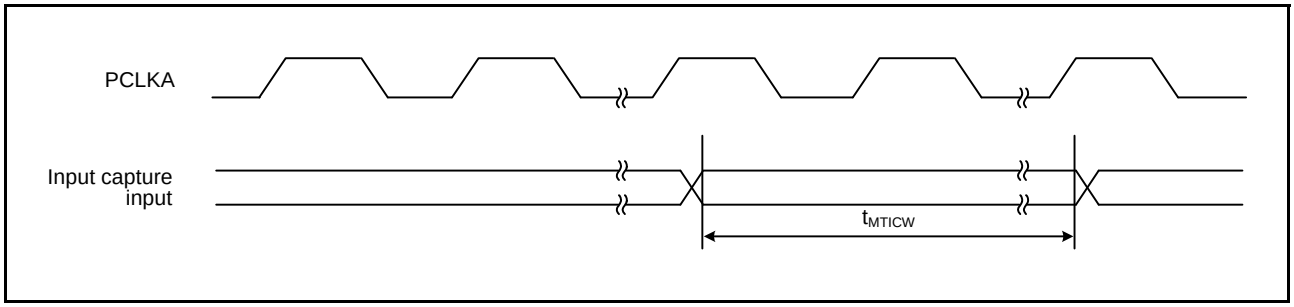


Figure 56.39 MTU Input Capture Input Timing

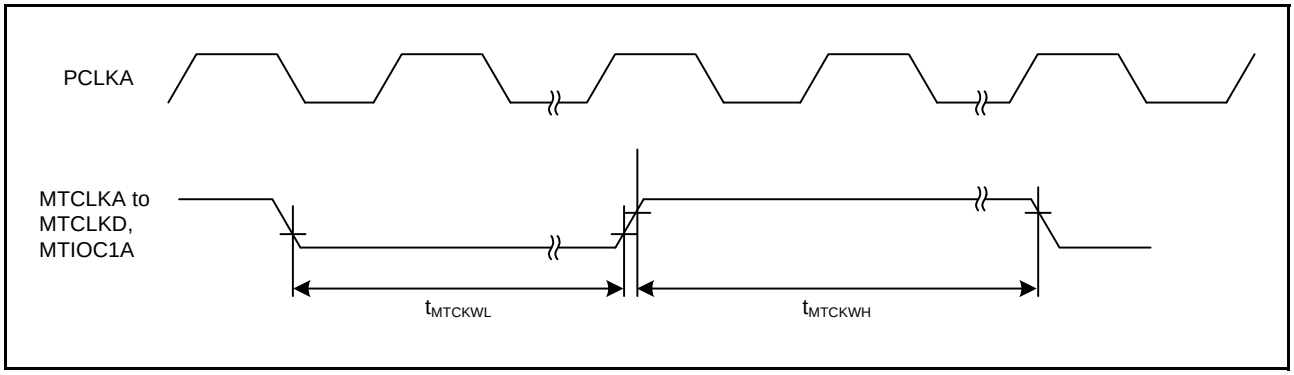


Figure 56.40 MTU Clock Input Timing

56.4.7.6 POE3

Table 56.33 POE3 Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the drive capacity control register.

Item			Symbol	Min.	Typ.	Max.	Unit*1	Test Conditions
POE	POEn# input pulse width (n = 0, 4, 8, 10, 11)		t_{POEW}	1.5	—	—	t_{PBcyc}	Figure 56.41
	Output disable time	Transition of the POEn# signal level	t_{POEDI}	—	—	5 PCLKB + 0.24	µs	Figure 56.42 When detecting falling edges (ICSRm.POEnM[3:0] = 0000b (m = 1 to 5; n = 0, 4, 8, 10, 11))
		Simultaneous conduction of output pins	t_{POEDO}	—	—	3 PCLKB + 0.2	µs	Figure 56.43
		Register setting	t_{POEDS}	—	—	1 PCLKB + 0.2	µs	Figure 56.44 Time for access to the register is not included.
		Oscillation stop detection	t_{POEDOS}	—	—	21	µs	Figure 56.45

Note 1. t_{PBcyc}: PCLKB cycle

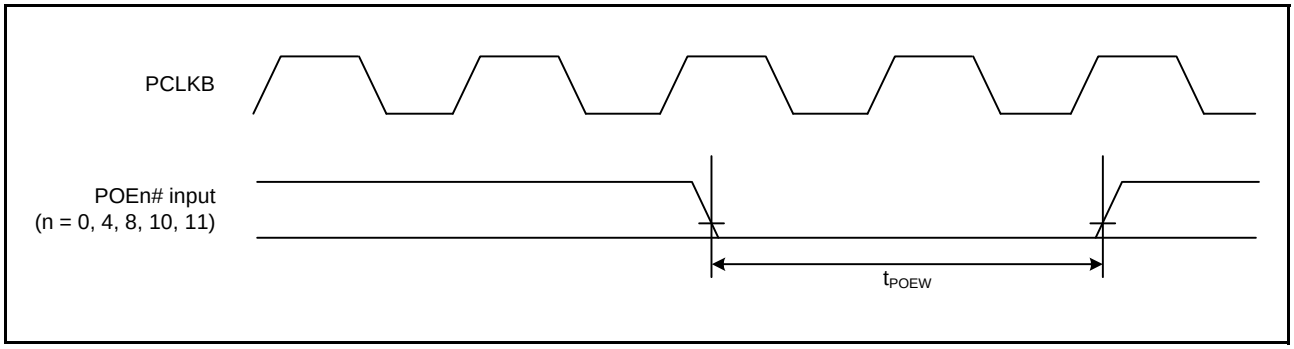


Figure 56.41 POE# Pin Input Timing

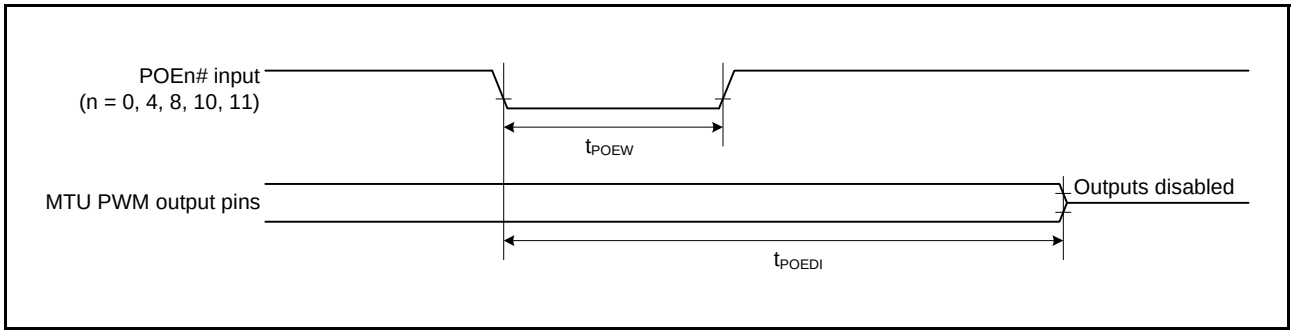
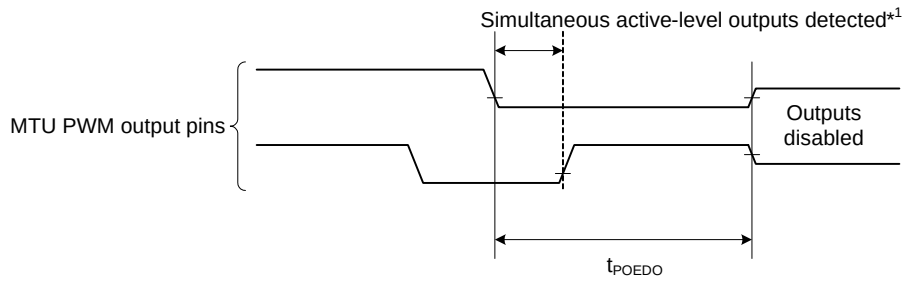


Figure 56.42 Output Disable Time for POE in Response to Transition of the POEn# Signal Level



Note 1. When the active level is set to low.

Figure 56.43 Output Disable Time for POE in Response to the Simultaneous Conduction of Output Pins

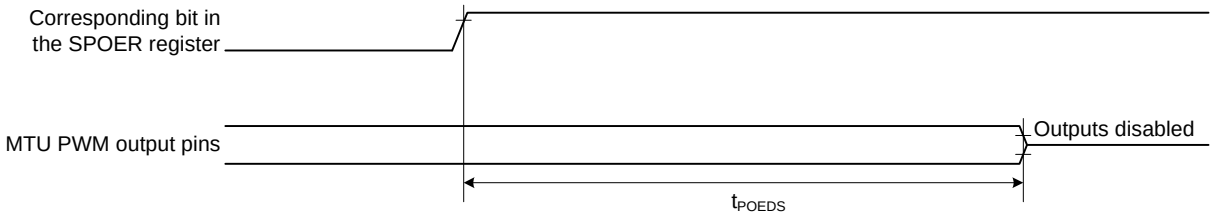


Figure 56.44 Output Disable Time for POE in Response to the Register Setting

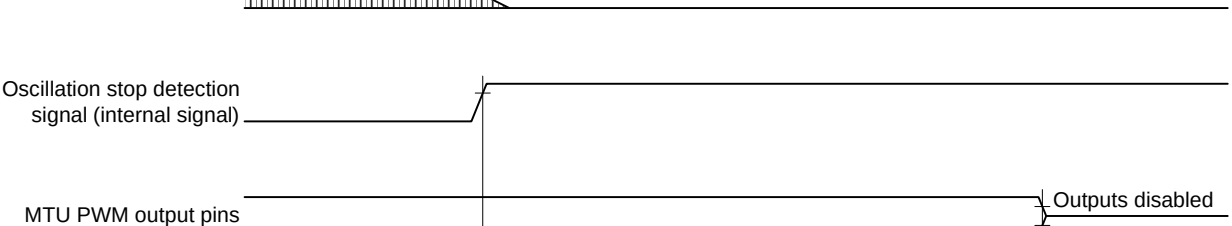


Figure 56.45 Output Disable Time for POE in Response to the Oscillation Stop Detection

56.4.7.7 A/D Converter Trigger

Table 56.34 A/D Converter Trigger Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the drive capacity control register.

Item		Symbol	Min.	Max.	Unit*1	Test Conditions
A/D converter	A/D converter trigger input pulse width	t_{TRGW}	1.5	—	t_{PBcyc}	Figure 56.46

Note 1. t_{PBcyc}: PCLKB cycle

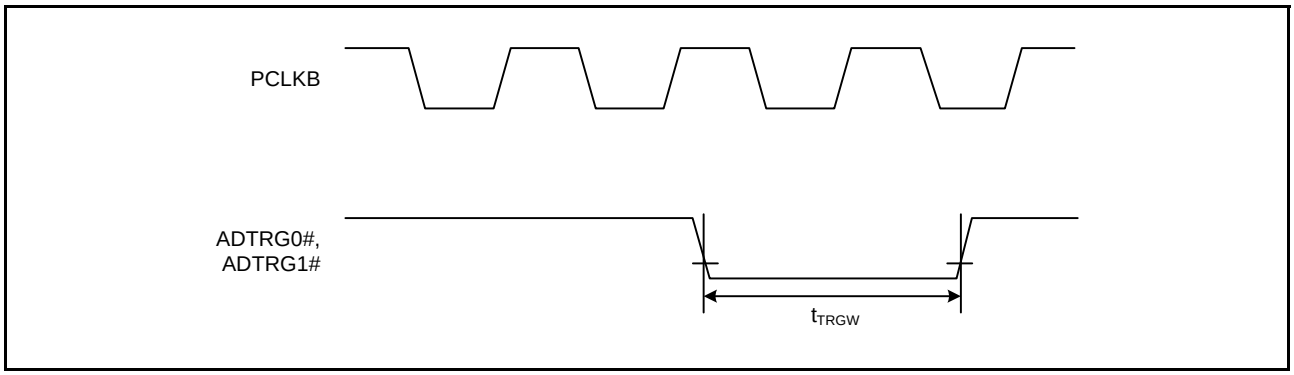


Figure 56.46 A/D Converter Trigger Input Timing

56.4.7.8 CAC

Table 56.35 CAC Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the drive capacity control register.

Item*1, *2			Symbol	Min.*1, *2	Max.	Unit	Test Conditions
CAC	CACREF input pulse width	$t_{PBcyc} \le t_{CAC}$	t_{CACREF}	4.5 t_{CAC} + 3 t_{PBcyc}	—	ns	
		$t_{PBcyc} > t_{CAC}$		5 t_{CAC} + 6.5 t_{PBcyc}	—		

Note 1. t_{PBcyc}: PCLKB cycle
Note 2. t_{CAC}: CAC count clock source cycle

56.4.7.9 SCI

Table 56.36 SCIk, SCIh, and SCIm Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the drive capacity control register.

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
SCIk, SCIh	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{PBcyc}	Figure 56.47
		Clock synchronous		6	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time		t_{SCKr}	—	5	ns	
	Input clock fall time		t_{SCKf}	—	5	ns	
	Output clock cycle	Asynchronous (SCIk)	t_{Scyc}	6	—	t_{PBcyc}	
		Asynchronous (SCIh)		8	—		
		Clock synchronous		4	—		
	Output clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Output clock rise time		t_{SCKr}	—	5	ns	
	Output clock fall time		t_{SCKf}	—	5	ns	
	Transmit data delay time	Clock synchronous	t_{TXD}	—	28	ns	Figure 56.48
	Receive data setup time	Clock synchronous	t_{RXS}	15	—	ns	
	Receive data hold time	Clock synchronous	t_{RXH}	5	—	ns	
SCIm	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{PAcyc}	Figure 56.47
		Clock synchronous		6	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time		t_{SCKr}	—	5	ns	
	Input clock fall time		t_{SCKf}	—	5	ns	
	Output clock cycle	Asynchronous	t_{Scyc}	6	—	t_{PAcyc}	
		Clock synchronous		4	—		
	Output clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Output clock rise time		t_{SCKr}	—	5	ns	
	Output clock fall time		t_{SCKf}	—	5	ns	
	Transmit data delay time	Master	t_{TXD}	—	15	ns	Figure 56.48
		Slave		—	28		
	Receive data setup time	Clock synchronous	t_{RXS}	20	—	ns	
	Receive data hold time	Clock synchronous	t_{RXH}	5	—	ns	

Note 1. t_{PBcyc}: PCLKB cycle; t_{PAcyc}: PCLKA cycle

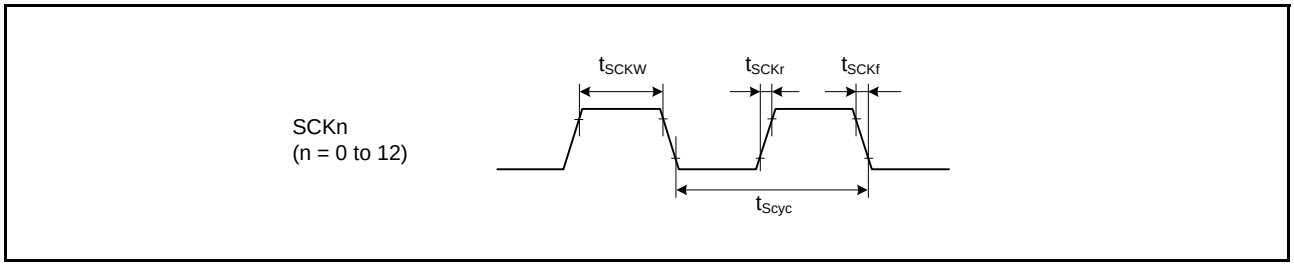


Figure 56.47 SCK Clock Input Timing

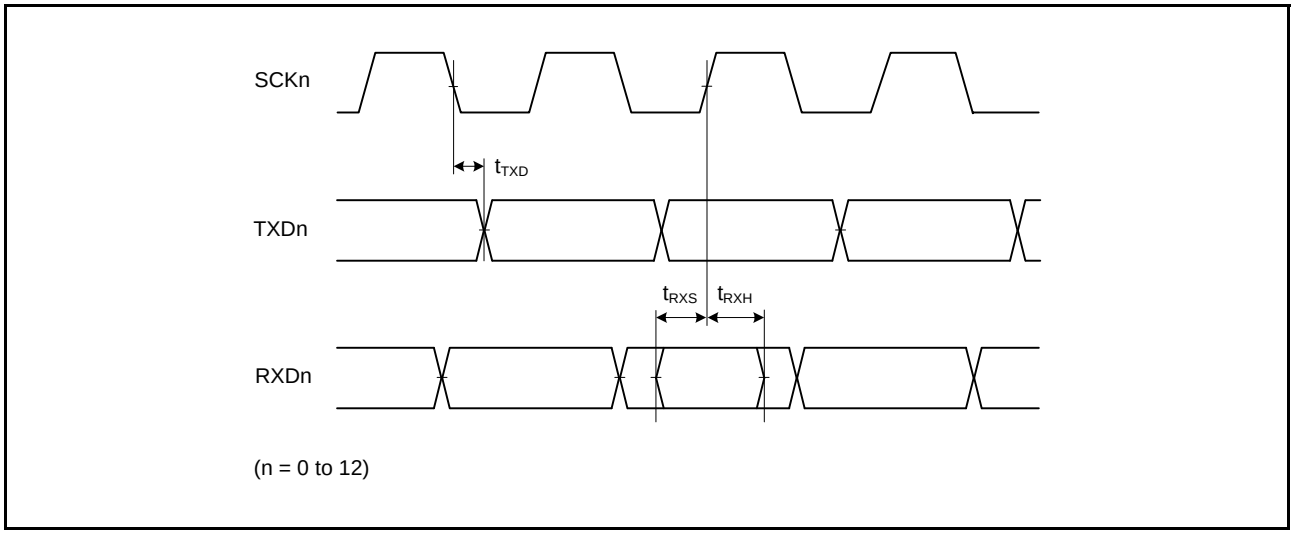


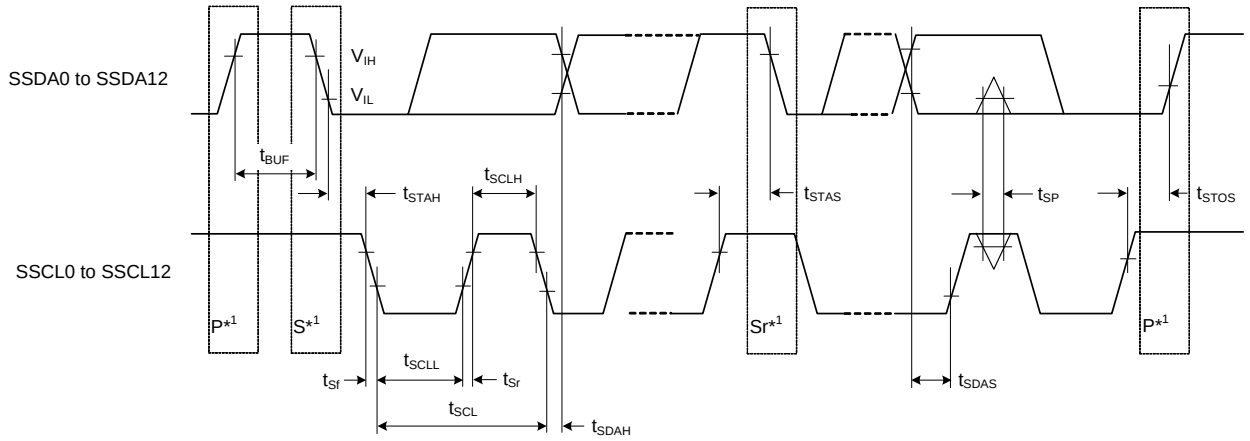
Figure 56.48 SCI Input/Output Timing: Clock Synchronous Mode

Table 56.37 Simple IIC Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
High-drive output is selected by the drive capacity control register.

Item		Symbol	Min.	Max.	Unit	Test Conditions
Simple IIC (Standard-mode)	SSCL, SSDA input rise time	t_{Sr}	—	1000	ns	Figure 56.49
	SSCL, SSDA input fall time	t_{Sf}	—	300	ns	
	SSCL, SSDA input spike pulse removal time	t_{SP}	0	4 × t_{Pcyc}	ns	
	Data input setup time	t_{SDAS}	250	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SSCL, SSDA capacitive load	C_b*1	—	400	pF	
Simple IIC (Fast-mode)	SSCL, SSDA input rise time	t_{Sr}	—	300	ns	
	SSCL, SSDA input fall time	t_{Sf}	—	300	ns	
	SSCL, SSDA input spike pulse removal time	t_{SP}	0	4 × t_{Pcyc}	ns	
	Data input setup time	t_{SDAS}	100	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SSCL, SSDA capacitive load	C_b*1	—	400	pF	

Note: t_{Pcyc} refers to the period of PCLKA in SCI10 and SCI11, and of PCLKB in SCI0 to SCI9, and SCI12.
Note 1. Cb is the total capacitance of the bus lines.



Note 1. S, P, and Sr indicate the following conditions.
S: Start condition
P: Stop condition
Sr: Restart condition

Test conditions
V_{IH} = 0.7 × VCC, V_{IL} = 0.3 × VCC
V_{OL} = 0.6 V, I_{OL} = 6 mA

Figure 56.49 Simple IIC Bus Interface Input/Output Timing

Table 56.38 Simple SPI Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the drive capacity control register.

Item		Symbol	Min.	Max.	Unit	Test Conditions
Simple SPI	SCK clock cycle output (master)	t_{SPcyc}	4	—	t_{Pcyc}	Figure 56.50
	SCK clock cycle input (slave)		6	—		
	SCK clock high pulse width	t_{SPCKWH}	0.4	0.6	t_{SPcyc}	
	SCK clock low pulse width	t_{SPCKWL}	0.4	0.6	t_{SPcyc}	
	SCK clock rise/fall time	t_{SPCKr}, t_{SPCKf}	—	20	ns	
	Data input setup time	t_{SU}	33.3	—	ns	Figure 56.51 to Figure 56.54
	Data input hold time	t_H	33.3	—	ns	
	SS input setup time	t_{LEAD}	1	—	t_{SPcyc}	
	SS input hold time	t_{LAG}	1	—	t_{SPcyc}	
	Data output delay time	t_{OD}	—	33.3	ns	
	Data output hold time	t_{OH}	–10	—	ns	
	Data rise/fall time	t_{Dr}, t_{Df}	—	16.6	ns	
	SS input rise/fall time	t_{SSLr}, t_{SSLf}	—	16.6	ns	
	Slave access time	t_{SA}	—	5	t_{Pcyc}	Figure 56.53, Figure 56.54
	Slave output release time	t_{REL}	—	5	t_{Pcyc}	

Note: t_{Pcyc} refers to the period of PCLKA in SCI10 and SCI11, and of PCLKB in SCI0 to SCI9, and SCI12.

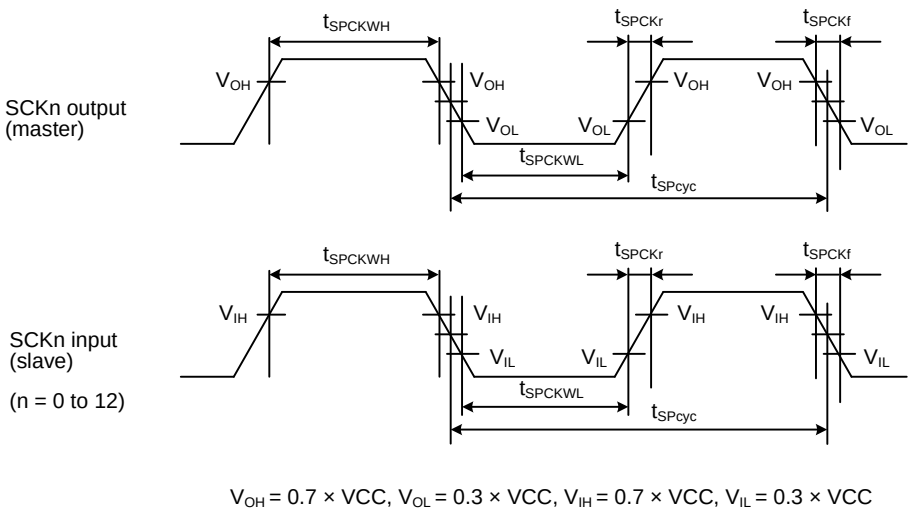


Figure 56.50 Simple SPI Clock Timing

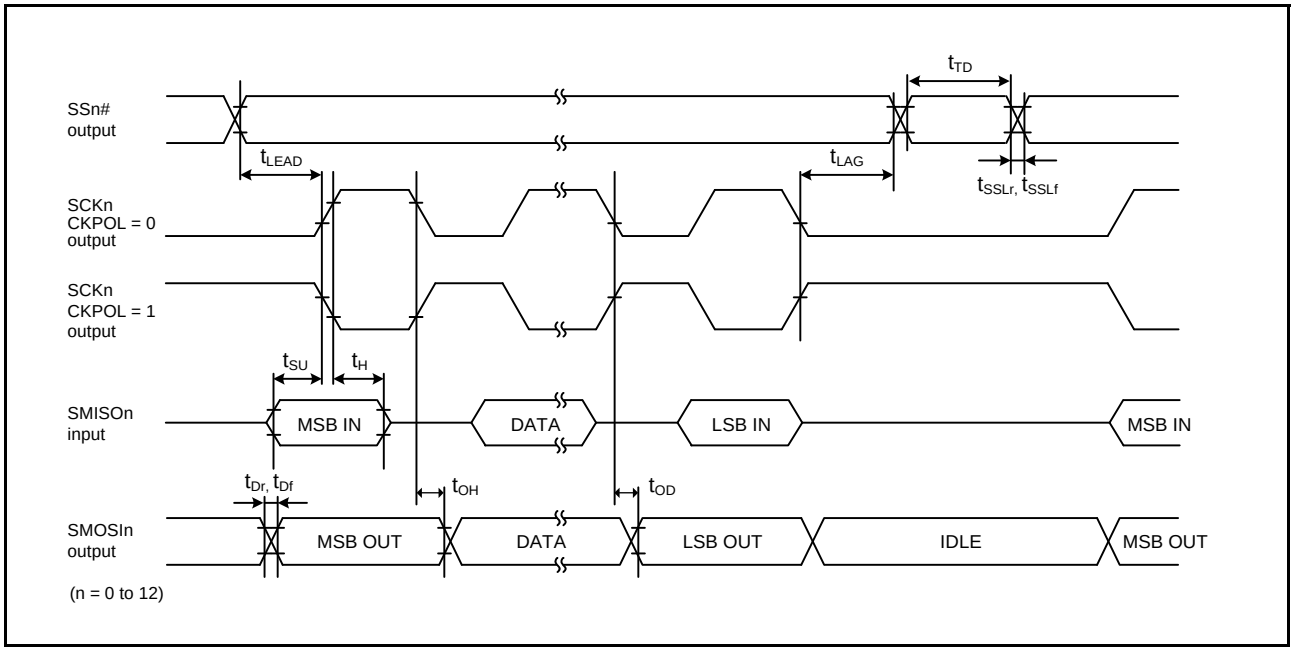


Figure 56.51 Simple SPI Timing (Master, CKPH = 1)

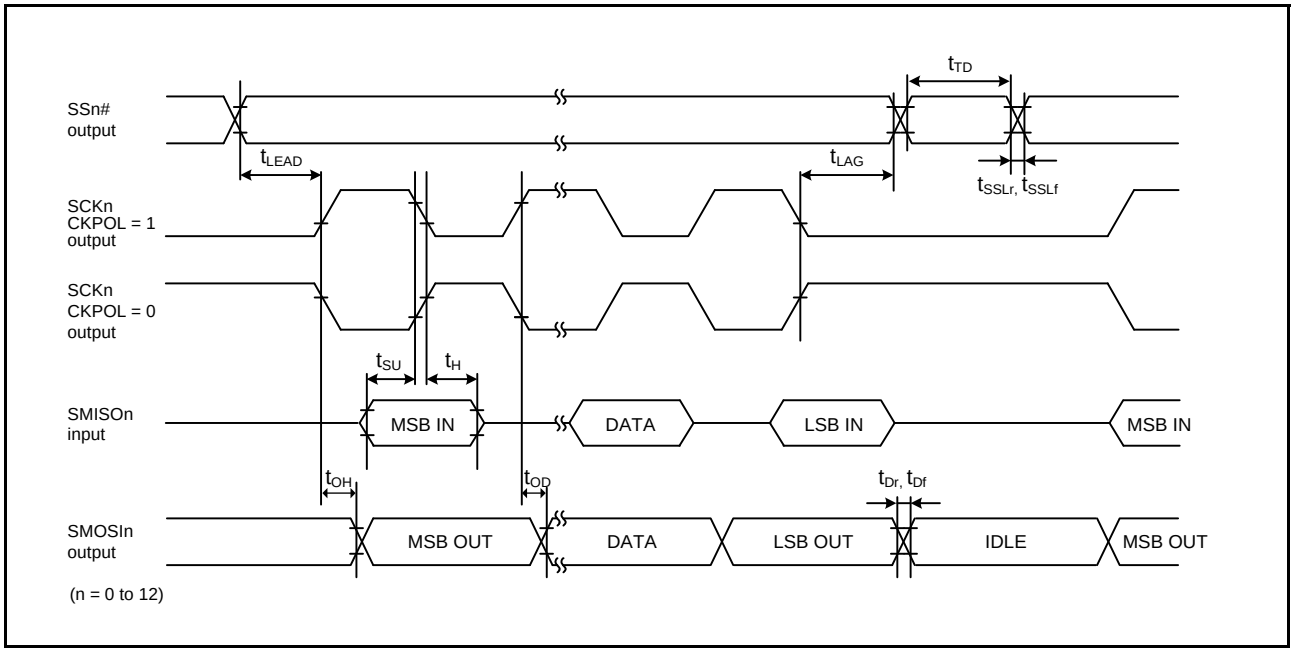


Figure 56.52 Simple SPI Timing (Master, CKPH = 0)

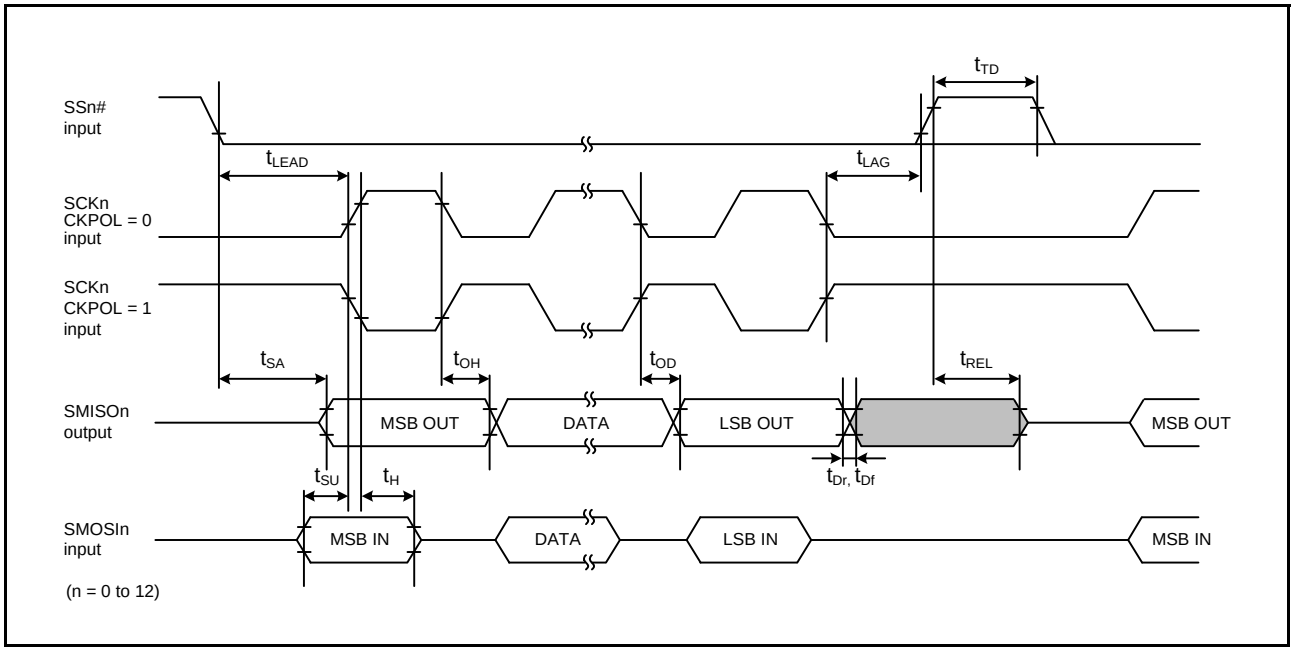


Figure 56.53 Simple SPI Timing (Slave, CKPH = 1)

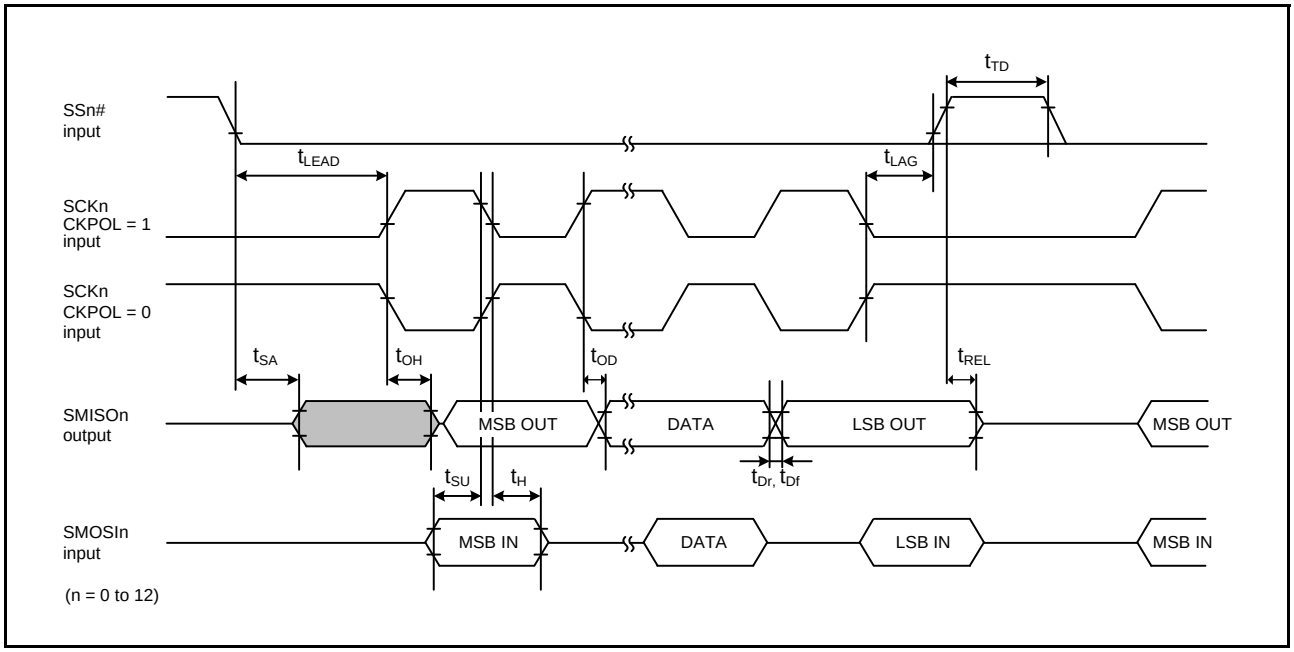


Figure 56.54 Simple SPI Timing (Slave, CKPH = 0)

56.4.7.10 RSCI

Table 56.39 RSCI Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-speed interface high-drive output is selected by the drive capacity control register.

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
RSCI	Input clock cycle	Asynchronous	t_{Scyc}	4	—	t_{PAcyc}	Figure 56.55
		Clock synchronous		2	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time		t_{SCKr}	—	5	ns	
	Input clock fall time		t_{SCKf}	—	5	ns	
	Output clock cycle	Asynchronous	t_{Scyc}	6	—	t_{PAcyc}	
		Clock synchronous		2	—		
	Output clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Output clock rise time		t_{SCKr}	—	5	ns	
	Output clock fall time		t_{SCKf}	—	5	ns	
	Receive data setup time	Master	t_{RXS}	0.5	—	ns	Figure 56.56
		Slave		2.5	—		
	Receive data hold time	Master	t_{RXH}	11	—	ns	
		Slave		2.5	—		
	Transmit data delay time	Master	t_{TXD}	—	4	ns	
		Slave		—	15		

Note 1. t_{PAcyc}: PCLKA cycle; t_{Scyc}: SCK cycle

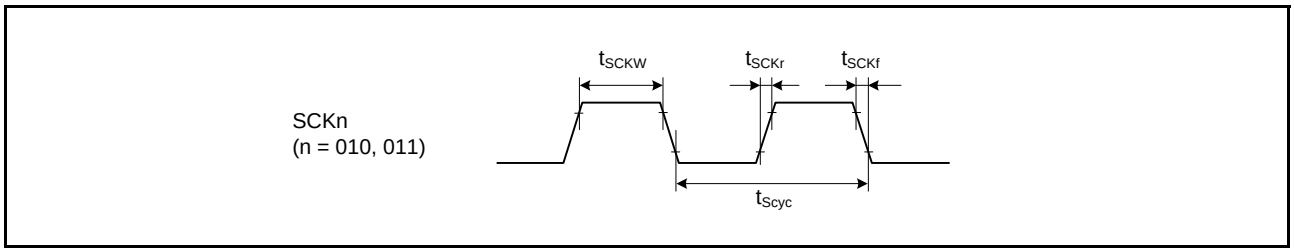


Figure 56.55 SCK Clock Input Timing

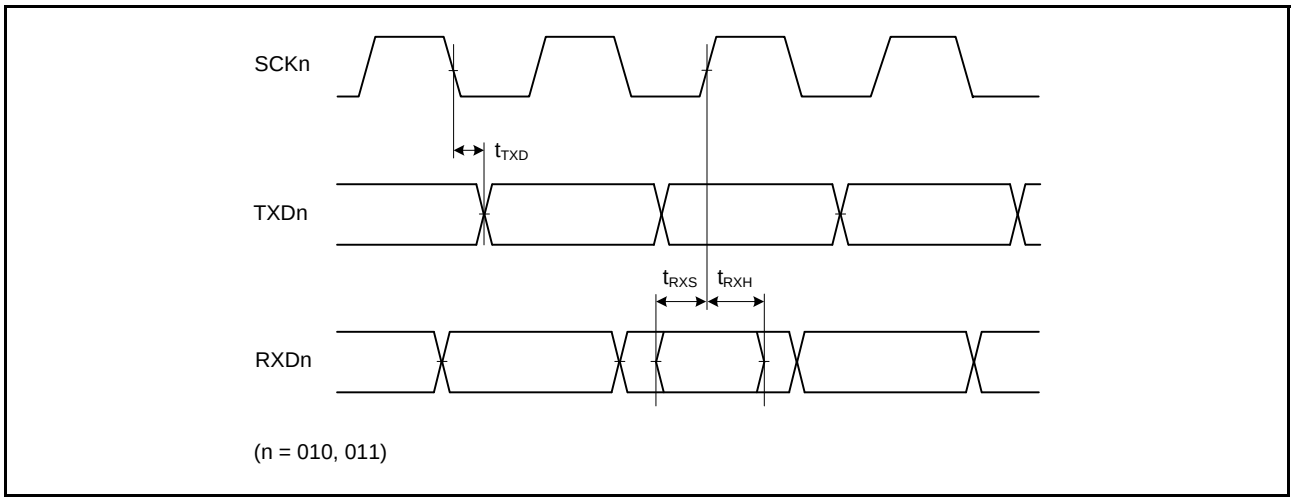


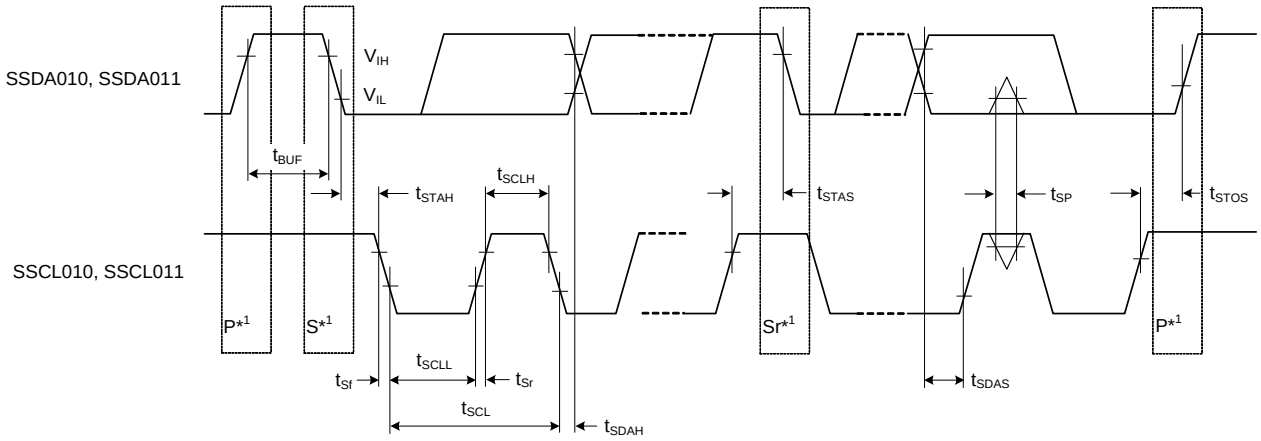
Figure 56.56 RSCI Input/Output Timing: Clock Synchronous Mode

Table 56.40 Simple IIC Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
High-speed interface high-drive output is selected by the drive capacity control register.

Item		Symbol	Min.	Max.	Unit	Test Conditions
Simple IIC (Standard-mode)	SSCL, SSDA input rise time	t_{Sr}	—	1000	ns	Figure 56.57
	SSCL, SSDA input fall time	t_{Sf}	—	300	ns	
	SSCL, SSDA input spike pulse removal time	t_{SP}	0	4 × t_{PAcyc}	ns	
	Data input setup time	t_{SDAS}	250	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SSCL, SSDA capacitive load	C_b*1	—	400	pF	
Simple IIC (Fast-mode)	SSCL, SSDA input rise time	t_{Sr}	—	300	ns	
	SSCL, SSDA input fall time	t_{Sf}	—	300	ns	
	SSCL, SSDA input spike pulse removal time	t_{SP}	0	4 × t_{PAcyc}	ns	
	Data input setup time	t_{SDAS}	100	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SSCL, SSDA capacitive load	C_b*1	—	400	pF	

Note: t_{PAcyc}: PCLKA cycle
Note 1. Cb is the total capacitance of the bus lines.



Note 1. S, P, and Sr indicate the following conditions.
S: Start condition
P: Stop condition
Sr: Restart condition

Test conditions
V_{IH} = 0.7 × VCC, V_{IL} = 0.3 × VCC
V_{OL} = 0.6 V, I_{OL} = 6 mA

Figure 56.57 Simple IIC Bus Interface Input/Output Timing

Table 56.41 Simple SPI Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-speed interface high-drive output is selected by the drive capacity control register.

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
Simple SPI	SCK clock cycle output (master)		t_{SPcyc}	2	—	t_{PAcyc}	Figure 56.58
	SCK clock cycle input (slave)			2	—		
	SCK clock high pulse width		t_{SPCKWH}	0.4	0.6	t_{SPcyc}	
	SCK clock low pulse width		t_{SPCKWL}	0.4	0.6	t_{SPcyc}	
	SCK clock rise/fall time	Output	t_{SPCKr}, t_{SPCKf}	—	5	ns	
		Input		—	1	µs	
	Data input setup time	Master	t_{SU}	0.5	—	ns	Figure 56.59 to Figure 56.62
		Slave		2.5	—		
	Data input hold time	Master	t_H	11	—	ns	
		Slave		2.5	—		
	Data output delay time	Master	t_{OD}	—	4	ns	
		Slave		—	15		
	Data output hold time	Master	t_{OH}	0	—	ns	
		Slave		0	—		
	Data rise/fall time	Output	t_{Dr}, t_{Df}	—	5	ns	
		Input		—	1	—	
	Slave access time		t_{SA}	—	5	t_{PAcyc}	Figure 56.61, Figure 56.62
	Slave output release time		t_{REL}	—	5	t_{PAcyc}	
	SS input setup time		t_{LEAD}	1	—	t_{SPcyc}	Figure 56.59 to Figure 56.62
	SS input hold time		t_{LAG}	1	—	t_{SPcyc}	
	SS input rise/fall time		t_{SSLr}, t_{SSLf}	—	1	µs	

Note 1. t_{PAcyc}: PCLKA cycle

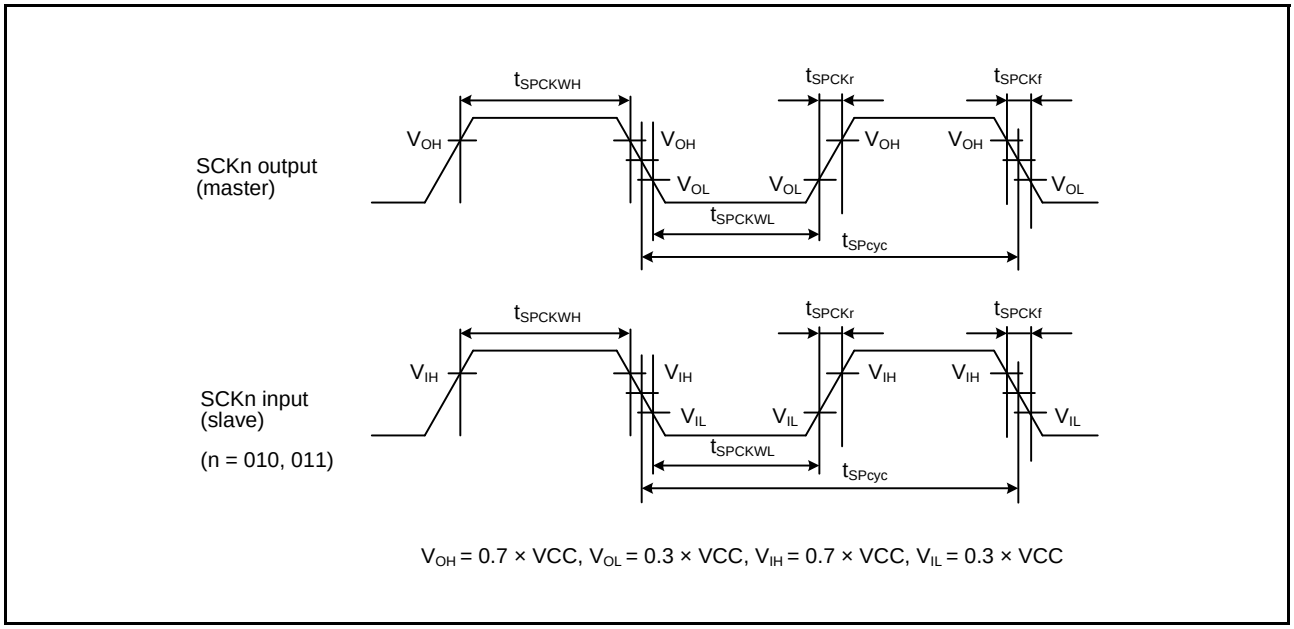


Figure 56.58 Simple SPI Clock Timing

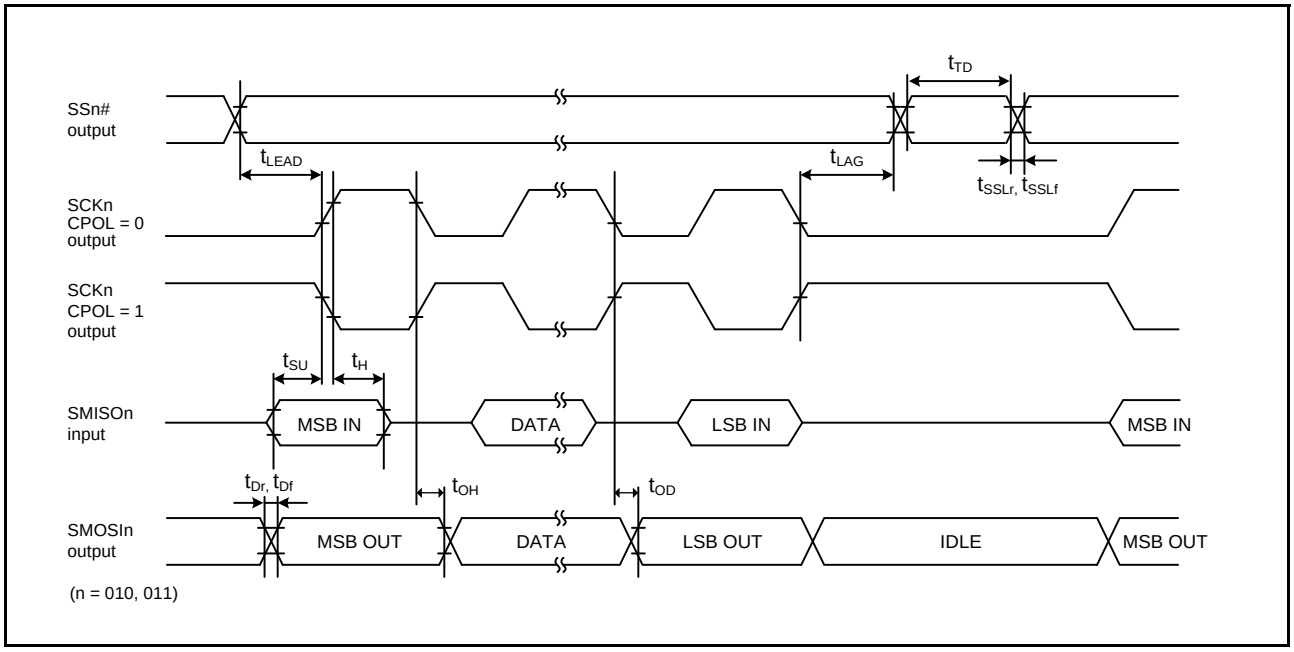


Figure 56.59 Simple SPI Timing (Master, CPHA = 0)

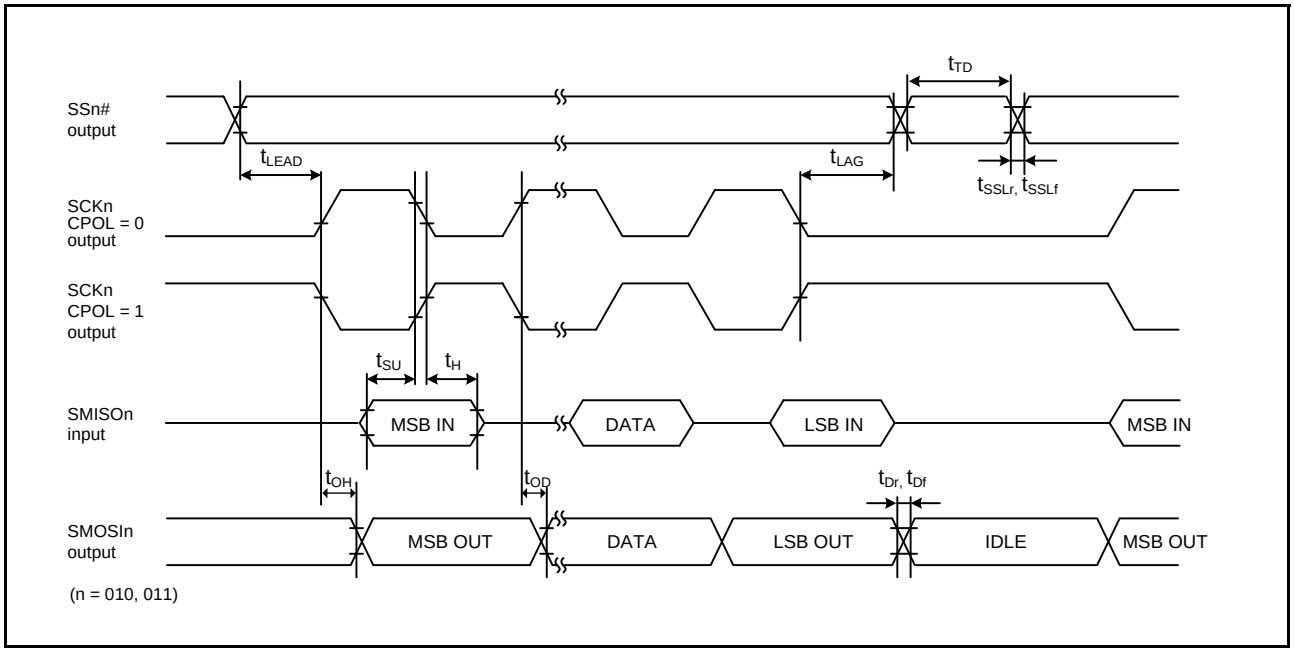


Figure 56.60 Simple SPI Timing (Master, CPHA = 1)

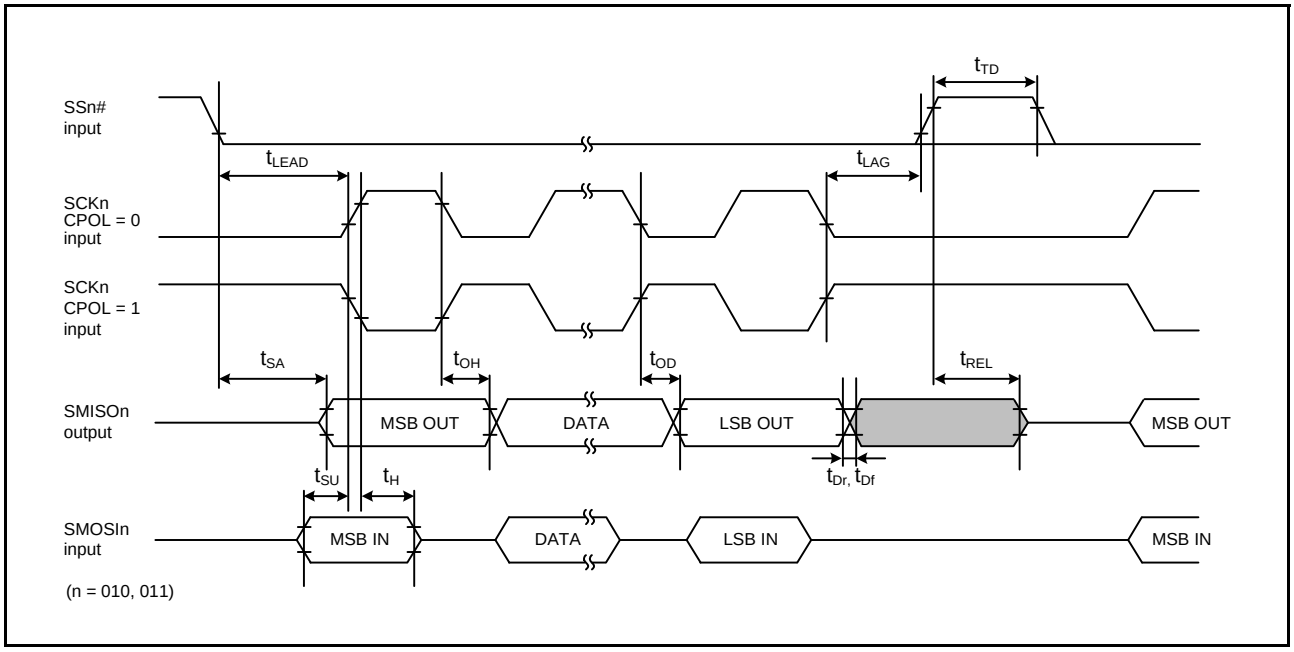


Figure 56.61 Simple SPI Timing (Slave, CPHA = 0)

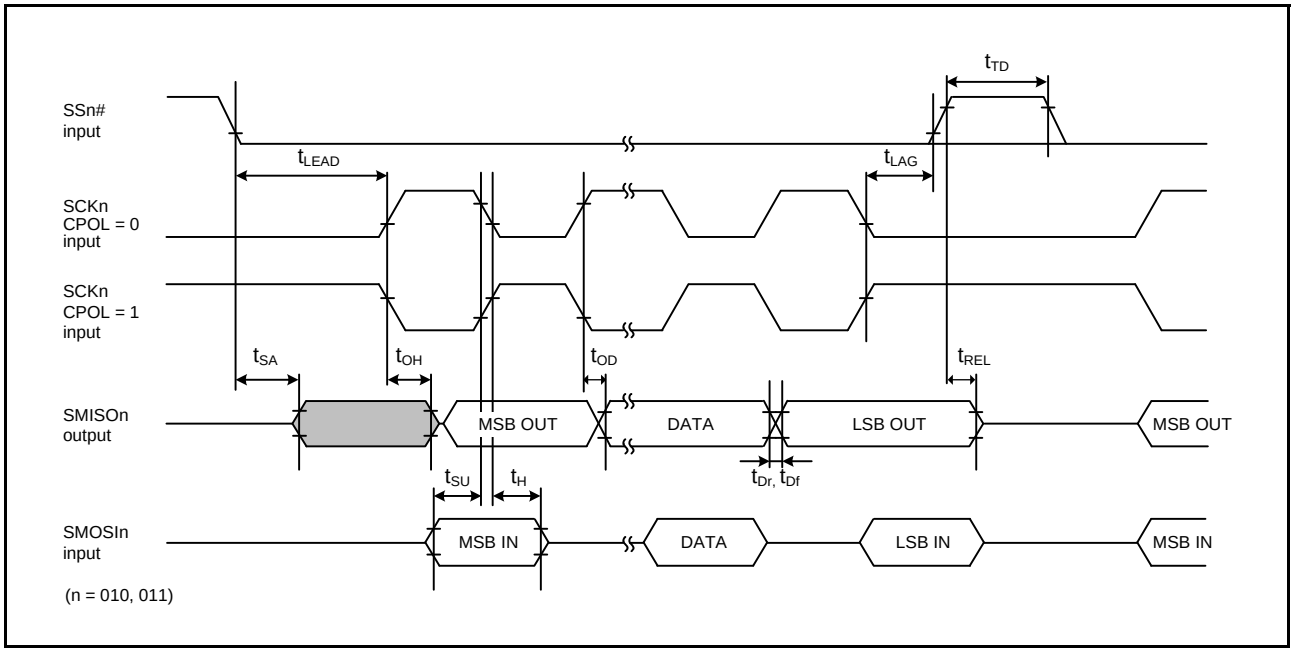


Figure 56.62 Simple SPI Timing (Slave, CPHA = 1)

56.4.7.11 SSIE

Table 56.42 Expansion Serial Sound Interface Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the drive capacity control register.

Item			Symbol	Min.	Max.	Unit	Test Conditions
AUDIO_CLK	Cycle		t_{EXcyc}	20	—	ns	Figure 56.63
	High/low level		t_{EXL}/t_{EXH}	0.4	0.6	t_{EXcyc}	
SSIBCK0	Cycle	Master	t_O	80	—	ns	Figure 56.64
		Slave	t_I	80	—	ns	
	Output clock high level	Master	t_{HC}	0.35	—	t_O	
	Output clock low level		t_{LC}	0.35	—	t_O	
	Input clock high level	Slave	t_{HC}	0.35	—	t_I	
	Input clock low level		t_{LC}	0.35	—	t_I	
	Output clock rise time	Master	t_{RC}	—	0.15	t_O	
	Output clock fall time		t_{FC}	—	0.15	t_O	
	Input clock rise time	Slave	t_{RC}	—	0.15	t_I	
	Input clock fall time		t_{FC}	—	0.15	t_I	
SSILRCK0,SSITXD0, SSIRXD0	Input setup time	Master	t_{SR}	12	—	ns	Figure 56.65, Figure 56.66
		Slave		12	—		
	Input hold time	Master	t_{HR}	8	—	ns	
		Slave		15	—		
	Output delay time	Master	t_{DTR}	–10	5	ns	
		Slave		0	20		
	Output delay time from when an SSILRCK0 signal is changed	Slave	t_{DTRW}	—	20	ns	Figure 56.67

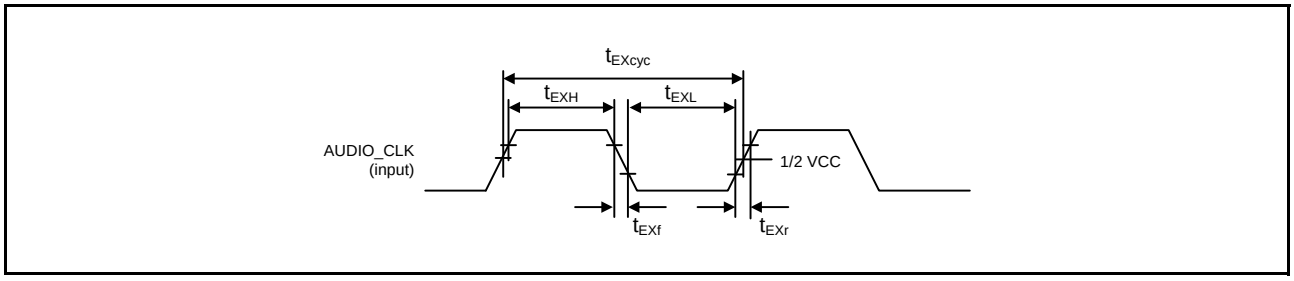


Figure 56.63 Clock Input Timing

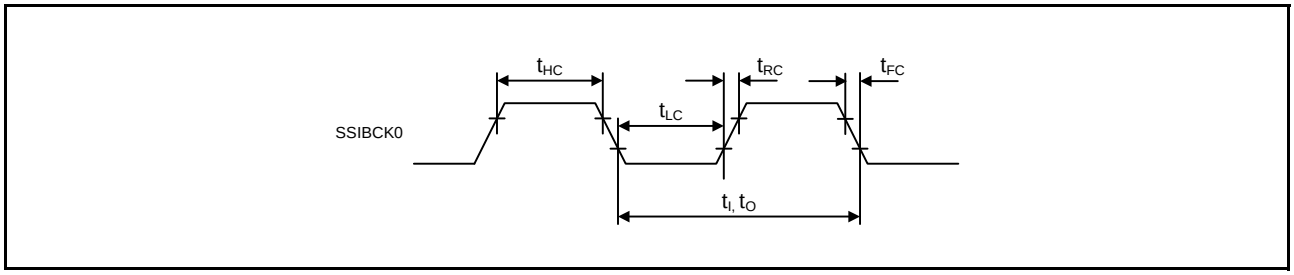


Figure 56.64 SSIE Clock Input/Output Timing

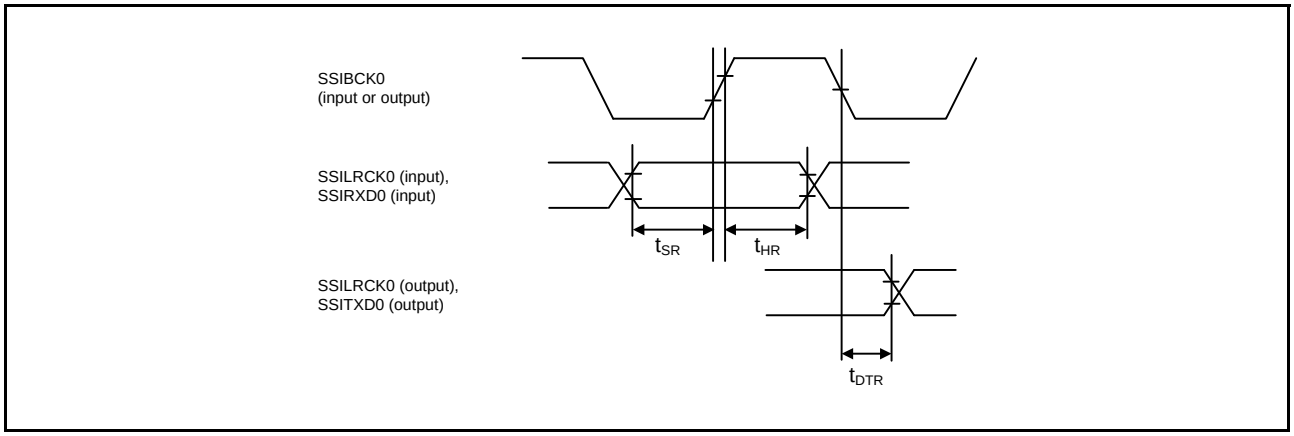


Figure 56.65 Transmission and Reception Timing for the SSIE Data When the SSICR.BCKP Bit is 0

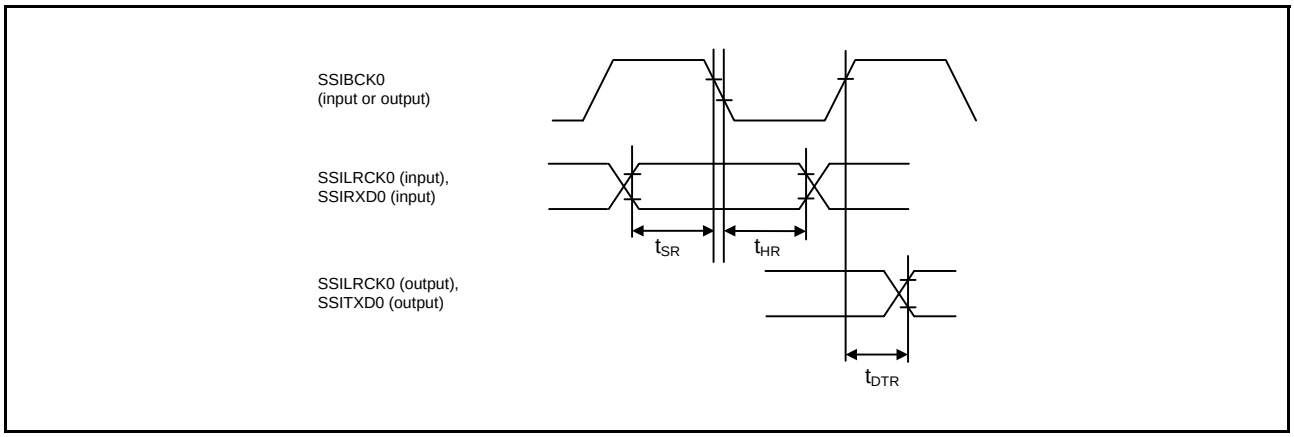


Figure 56.66 Transmission and Reception Timing for the SSIE Data When the SSICR.BCKP Bit is 1

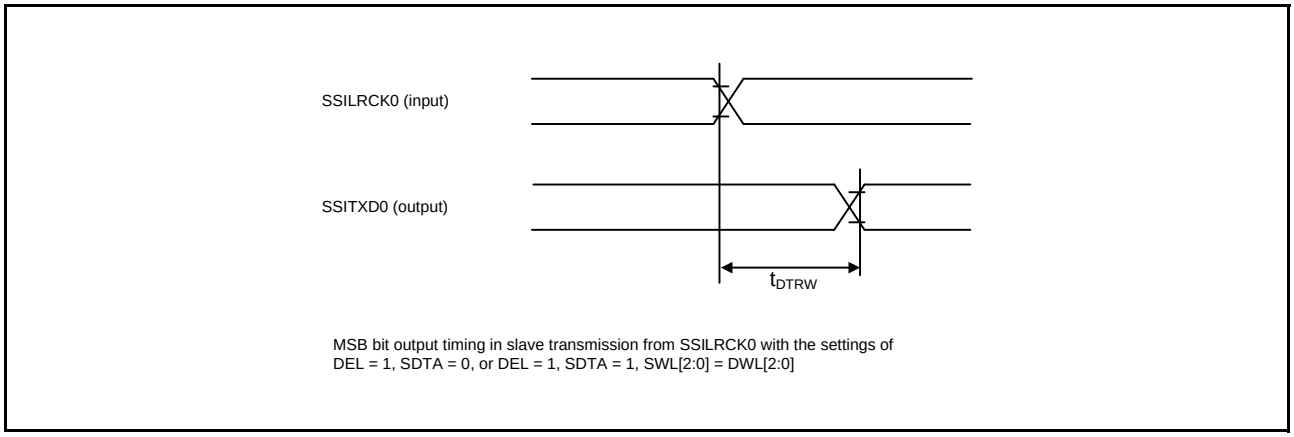


Figure 56.67 Output Delay of the SSIE Data from When an SSILRCK0 Signal is Changed

56.4.7.12 RSPI

Table 56.43 RSPI Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the drive capacity control register.

Item				Symbol	Min.*1	Max.*1	Unit*1	Test Conditions*2
RSPI	RSPCK clock cycle	Master		t_{SPcyc}	2	—	t_{PAcyc}	Figure 56.68
		Slave			4	—		
	RSPCK clock high pulse width	Master		t_{SPCKWH}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 3$	—	ns	
		Slave			0.4	0.6	t_{SPcyc}	
	RSPCK clock low pulse width	Master		t_{SPCKWL}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 3$	—	ns	
		Slave			0.4	0.6	t_{SPcyc}	
	RSPCK clock rise/fall time	Output		t_{SPCKr}, t_{SPCKf}	—	5	ns	
		Input			—	1	µs	
	Data input setup time	Master		t_{SU}	6	—	ns	Figure 56.69 to Figure 56.74
		Slave			8.3	—		
	Data input hold time	Master	PCLKA division ratio set to 1/2	t_{HF}	0	—	ns	
			PCLKA division ratio set to a value other than 1/2	t_H	t_{PAcyc}	—	ns	
		Slave			8.3	—		
	SSL setup time	Master		t_{LEAD}	1	8	t_{SPcyc}	
		Slave			4	—	t_{PAcyc}	
	SSL hold time	Master		t_{LAG}	1	8	t_{SPcyc}	
		Slave			4	—	t_{PAcyc}	
	Data output delay time	Master		t_{OD}	—	6.3	ns	
		Slave			—	28		
	Data output hold time	Master		t_{OH}	0	—	ns	
		Slave			0	—		
	Successive transmission delay time	Master		t_{TD}	$t_{SPcyc} + 2 \times t_{PAcyc}$	$8 \times t_{SPcyc} + 2 \times t_{PAcyc}$	ns	
		Slave			$4 \times t_{PAcyc}$	—		
	MOSI and MISO rise/fall time	Output		t_{Dr}, t_{Df}	—	5	ns	
		Input			—	1	µs	
	SSL rise/fall time	Output		t_{SSLr}, t_{SSLf}	—	5	ns	
		Input			—	1	µs	
	Slave access time			t_{SA}	—	28	ns	Figure 56.73, Figure 56.74
	Slave output release time			t_{REL}	—	28	ns	

Note 1. t_{PAcyc}: PCLKA cycle
Note 2. When a letter “-A”, “-B”, etc. to indicate group membership is appended to the pin name, each pin is recommended to use in combination with the pins in the same group. All RSPI AC timings are measured in combination with the pins in the same group.

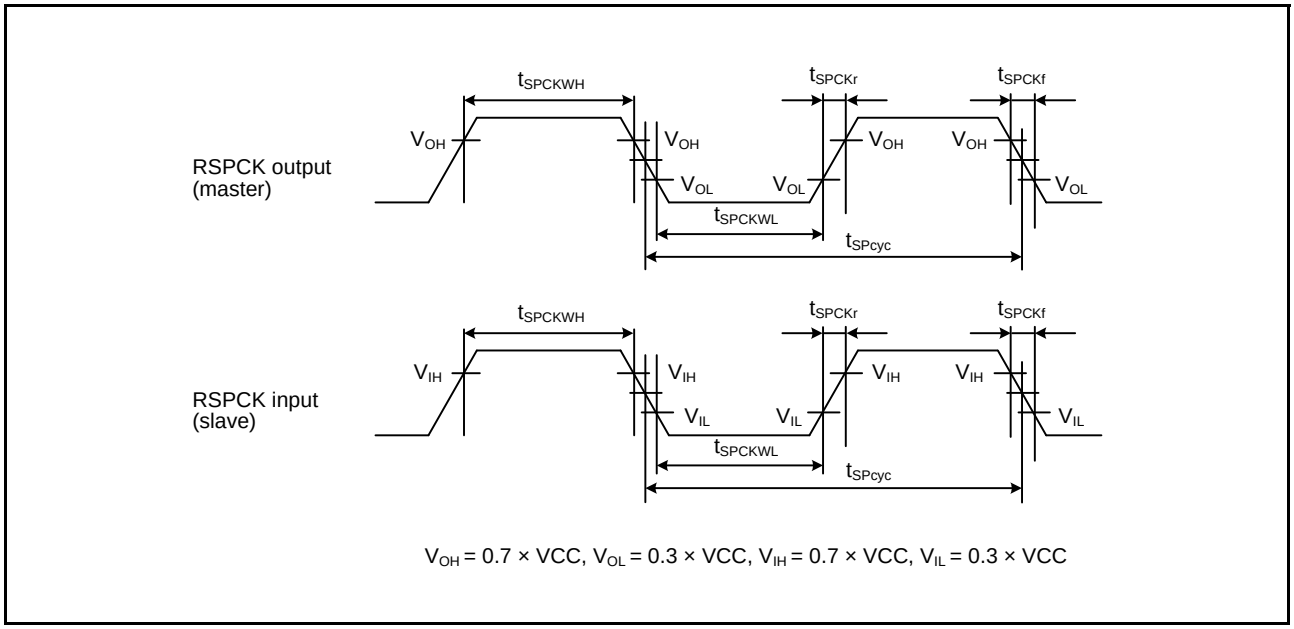


Figure 56.68 RSPI Clock Timing

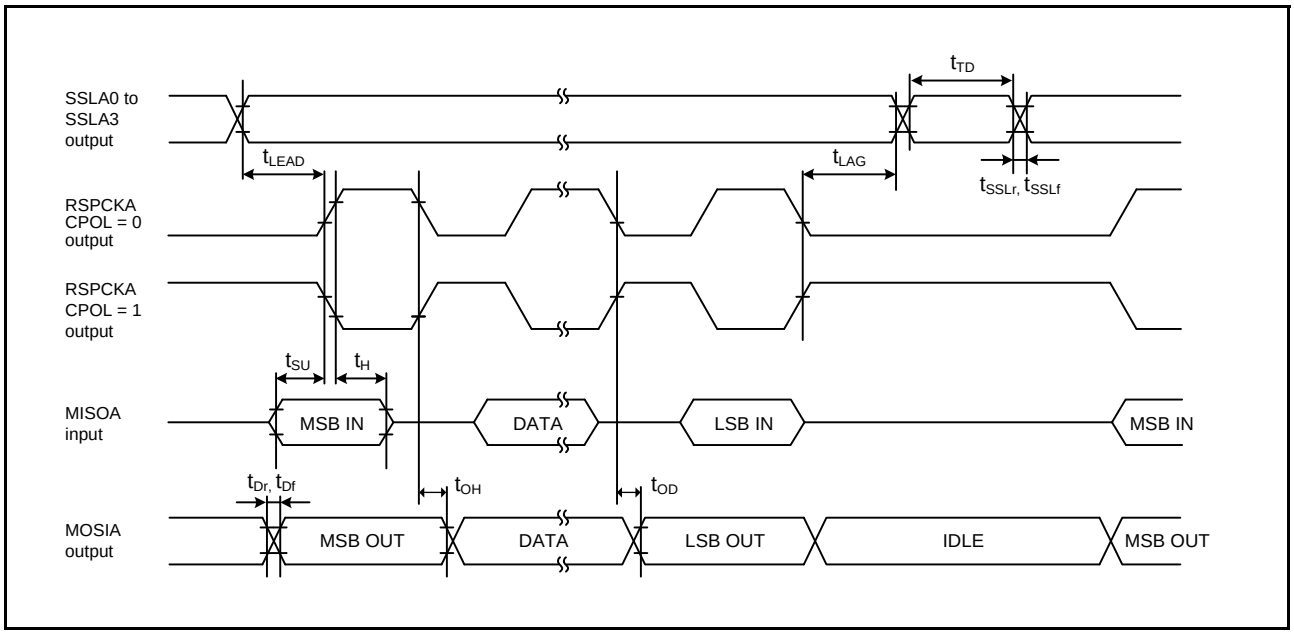


Figure 56.69 RSPI Timing (Master, CPHA = 0) (Bit Rate: PCLKA Division Ratio Set to a Value Other Than 1/2)

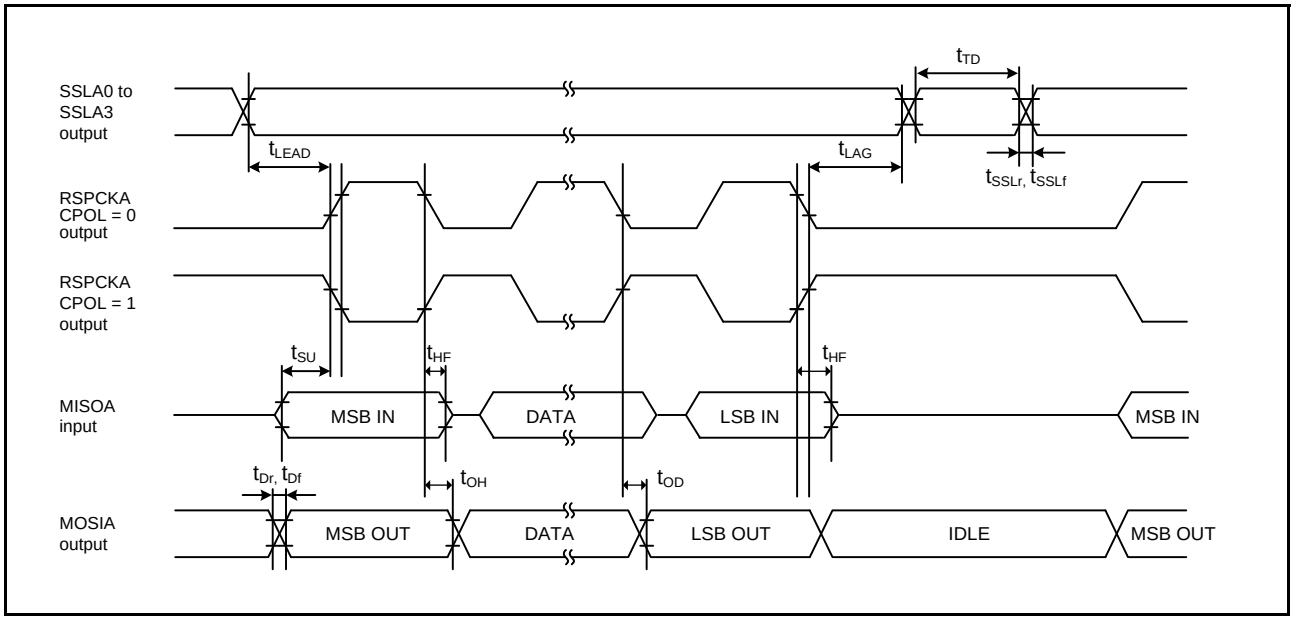


Figure 56.70 RSPI Timing (Master, CPHA = 0) (Bit Rate: PCLKA Division Ratio Set to 1/2)

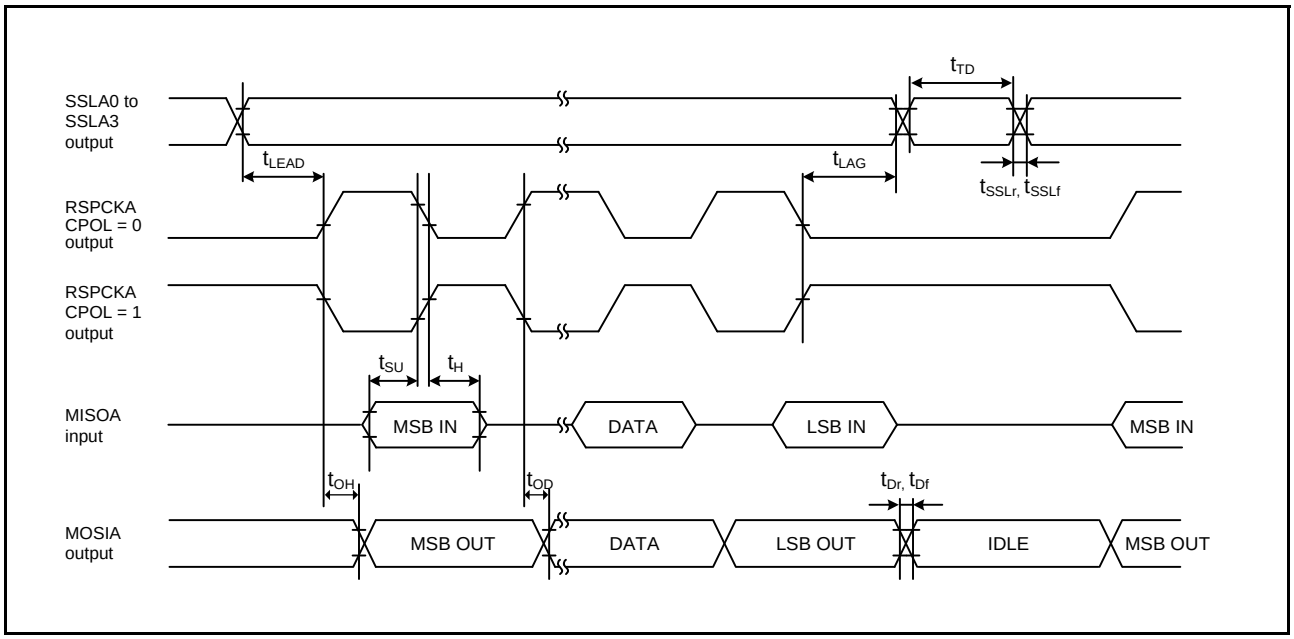


Figure 56.71 RSPI Timing (Master, CPHA = 1) (Bit Rate: PCLKA Division Ratio Set to a Value Other Than 1/2)

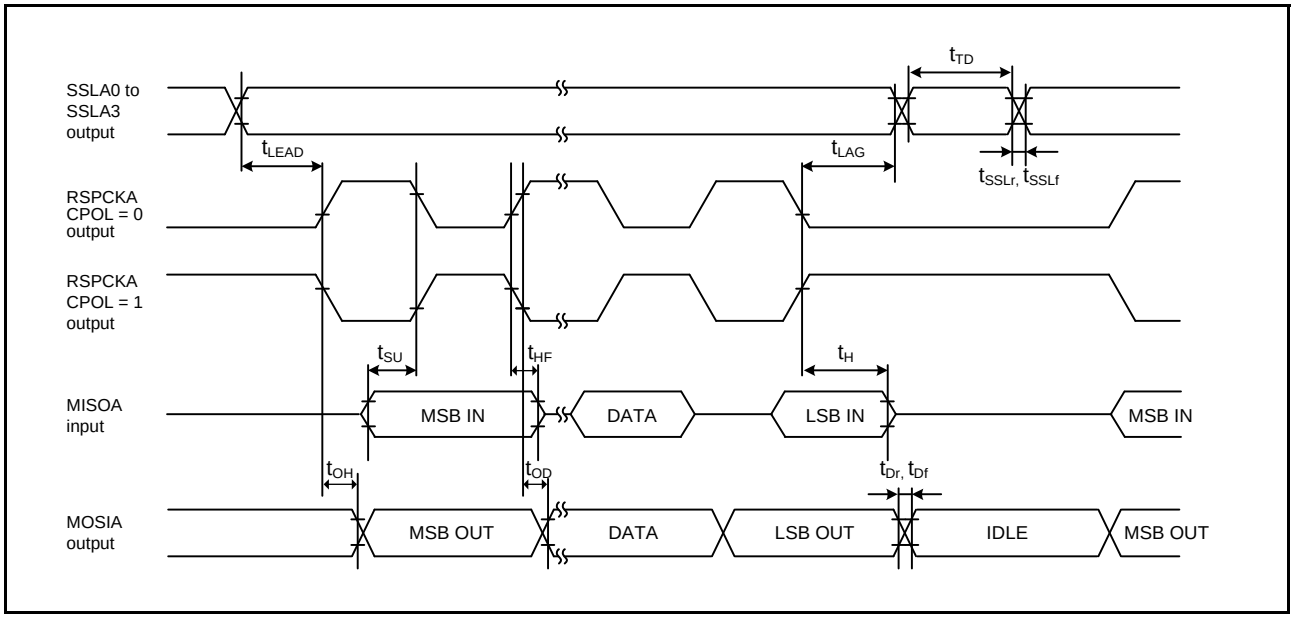


Figure 56.72 RSPI Timing (Master, CPHA = 1) (Bit Rate: PCLKA Division Ratio Set to 1/2)

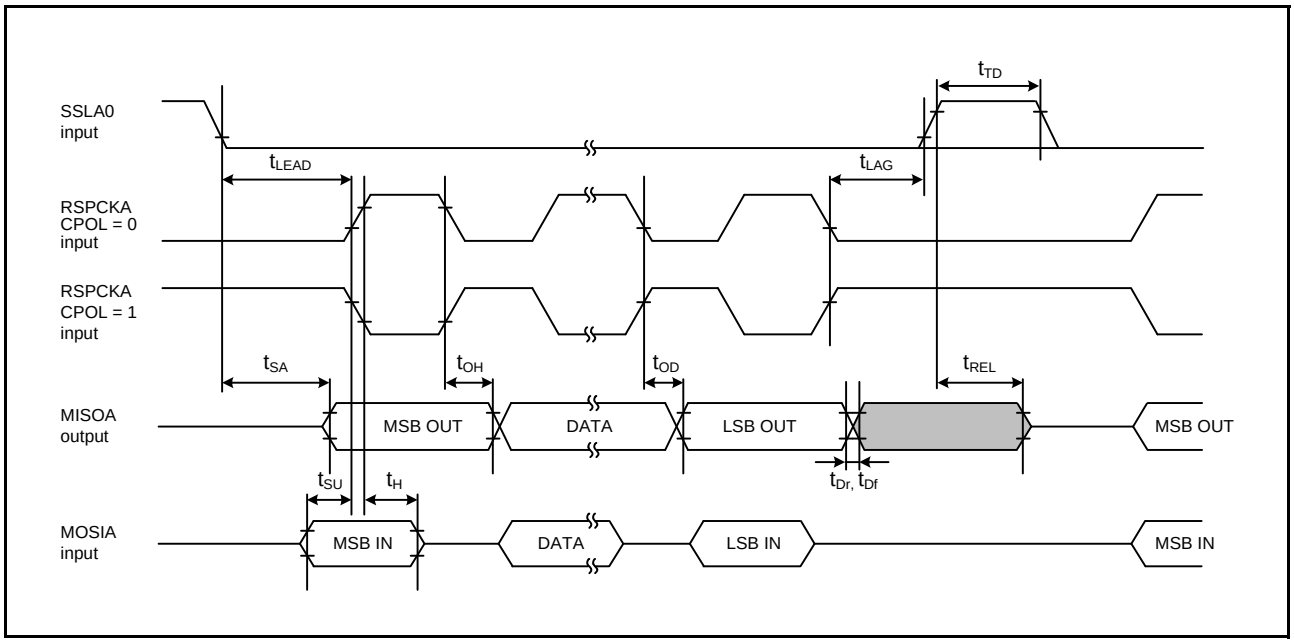


Figure 56.73 RSPI Timing (Slave, CPHA = 0)

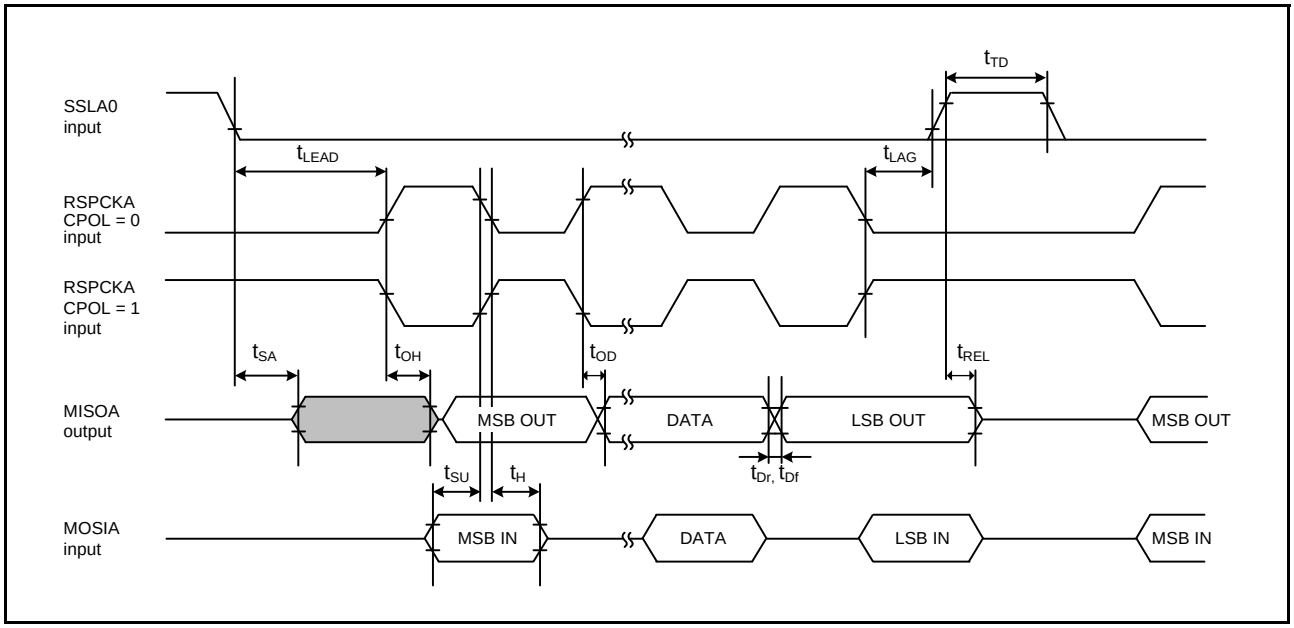


Figure 56.74 RSPI Timing (Slave, CPHA = 1)

56.4.7.13 RSPIA

Table 56.44 RSPIA Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-speed interface high-drive output is selected by the drive capacity control register.

Item			Symbol	Min.*1	Max.*1	Unit*1	Test Conditions*2
RSPI	RSPCK clock cycle	Master	t_{SPcyc}	2	—	t_{PAcyc}	Figure 56.75
		Slave		2	—		
	RSPCK clock high pulse width	Master	t_{SPCKWH}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 3$	—	ns	
		Slave		0.4	0.6	t_{SPcyc}	
	RSPCK clock low pulse width	Master	t_{SPCKWL}	$(t_{SPcyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 3$	—	ns	
		Slave		0.4	0.6	t_{SPcyc}	
	RSPCK clock rise/fall time	Output	t_{SPCKr}, t_{SPCKf}	—	5	ns	
		Input		—	1	µs	
	Data input setup time	Master	t_{SU}	0	—	ns	Figure 56.76 to Figure 56.82
		Slave		2.5	—		
	Data input hold time	Master	t_H	5.7	—	ns	
		Slave		2.5	—		
	SSL setup time	Master	t_{LEAD}	1	8	t_{SPcyc}	
		Slave		6	—	t_{PAcyc}	
	SSL hold time	Master	t_{LAG}	1	8	t_{SPcyc}	
		Slave		6	—	t_{PAcyc}	
	Data output delay time	Master	t_{OD}	—	4	ns	
		Slave		—	14		
	Data output hold time	Master	t_{OH}	0	—	ns	
		Slave		0	—		
	Successive transmission delay time	Master	t_{TD}	$t_{SPcyc} + 2 \times t_{PAcyc}$	$8 \times t_{SPcyc} + 2 \times t_{PAcyc}$	ns	
		Slave		t_{SPcyc}	—		
	MOSI and MISO rise/fall time	Output	t_{Dr}, t_{Df}	—	5	ns	
		Input		—	1	µs	
	SSL rise/fall time	Output	t_{SSLr}, t_{SSLf}	—	5	ns	
		Input		—	1	µs	
	Slave access time		t_{SA}	—	20	ns	Figure 56.79, Figure 56.80
	Slave output release time		t_{REL}	—	20	ns	
	TI SSP SS input setup time	Slave	t_{TISS}	4.5	—	ns	Figure 56.81, Figure 56.82
	TI SSP SS input hold time	Slave	t_{TISH}	2.5	—	ns	
	TI SSP next-access delay time	Slave	t_{TIND}	$2 \times t_{PAcyc}$ + SLNDL × t_{PAcyc}	—	ns	
	TI SSP SS output delay time	Master	t_{TISSOD}	—	7	ns	Figure 56.78

Note 1. t_{PAcyc}: PCLKA cycle
Note 2. When a letter “-A”, “-B”, etc. to indicate group membership is appended to the pin name, each pin is recommended to use in combination with the pins in the same group. All RSPIA AC timings are measured in combination with the pins in the same group.

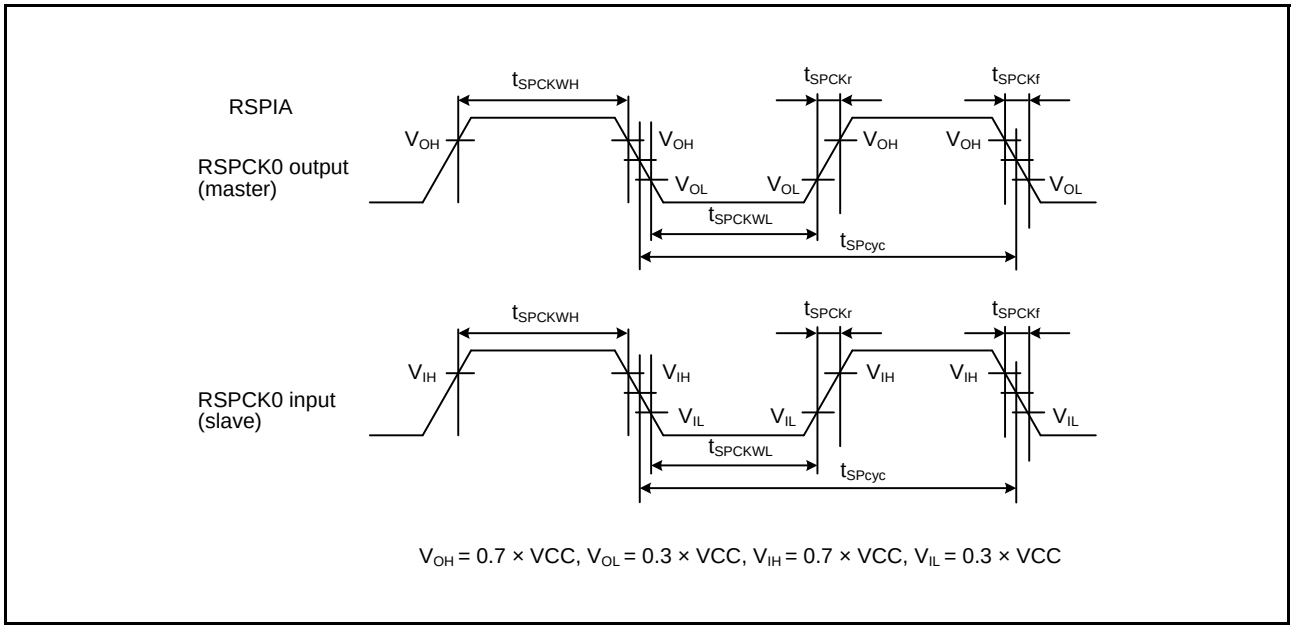


Figure 56.75 RSPIA Clock Timing

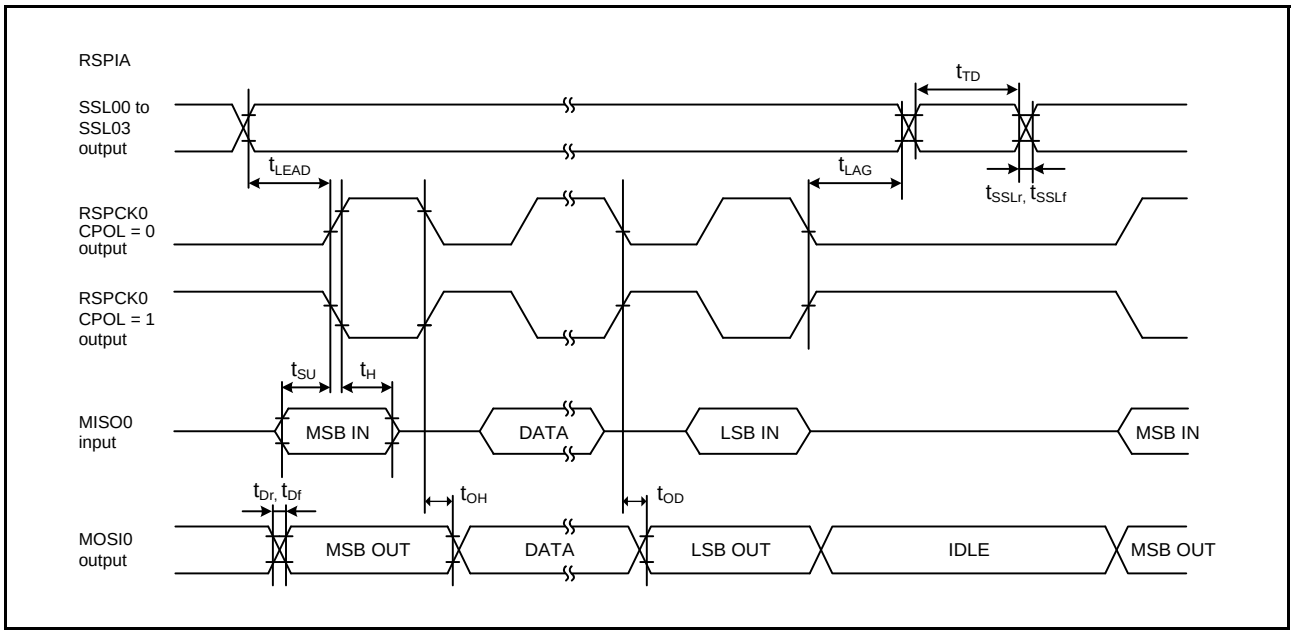


Figure 56.76 RSPIA Timing (Master, Motorola SPI, CPHA = 0)

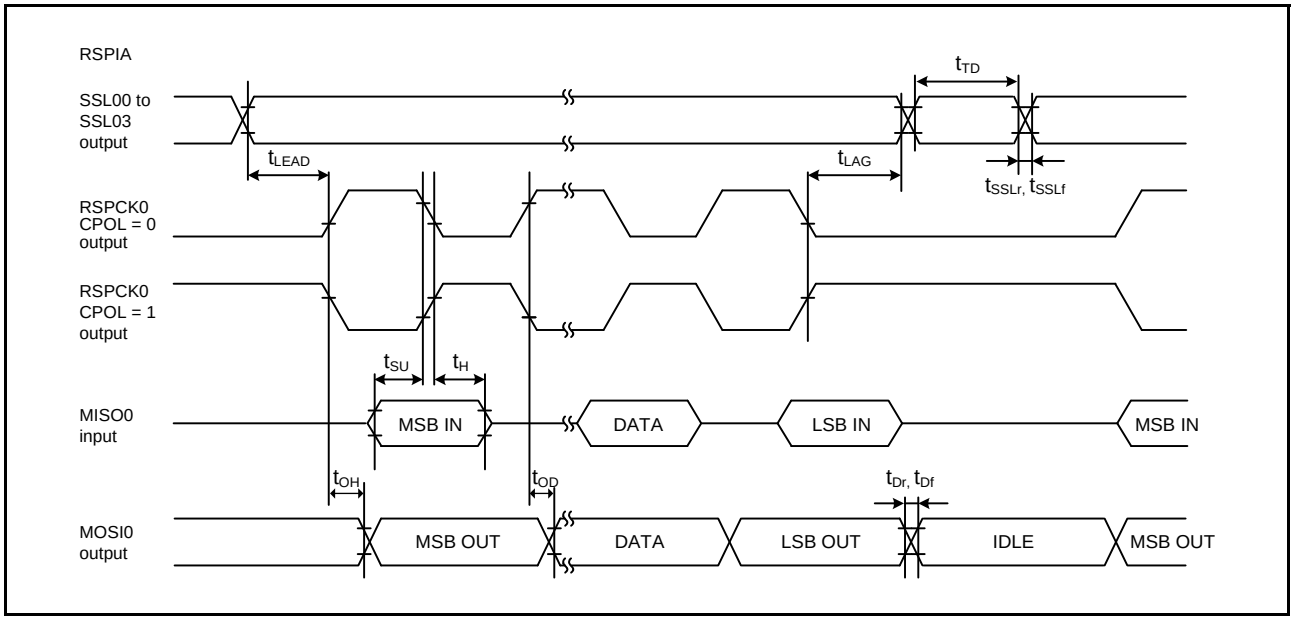


Figure 56.77 RSPIA Timing (Master, Motorola SPI, CPHA = 1)

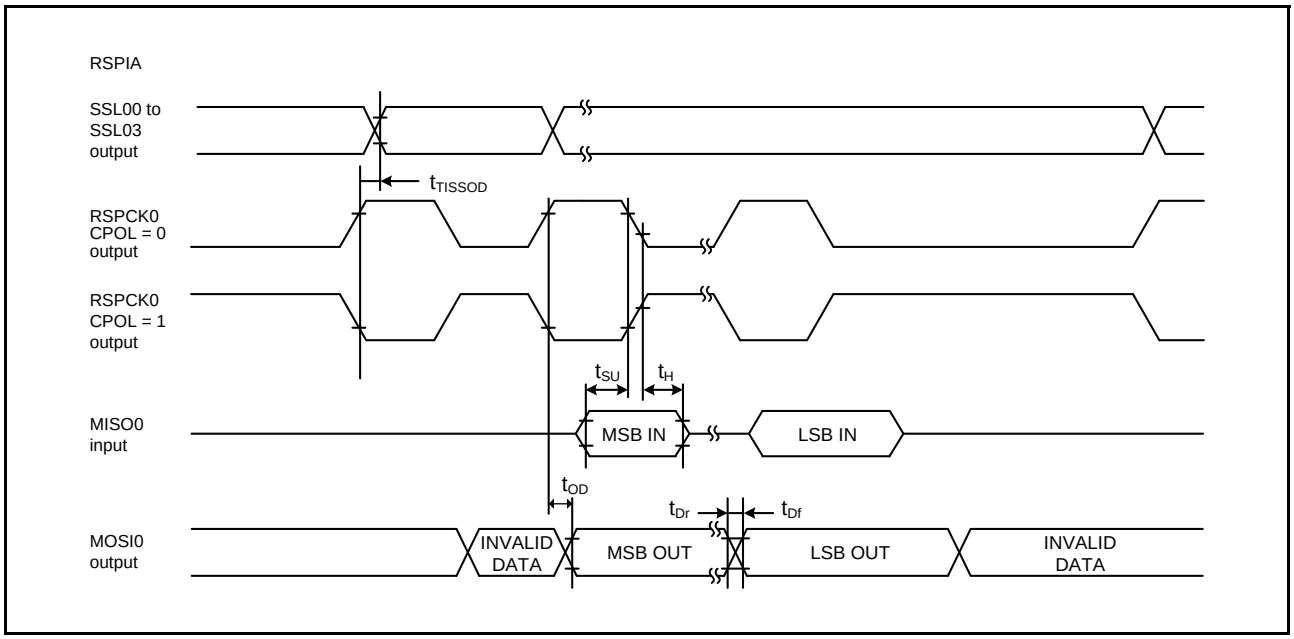


Figure 56.78 RSPIA Timing (Master, TI SSP)

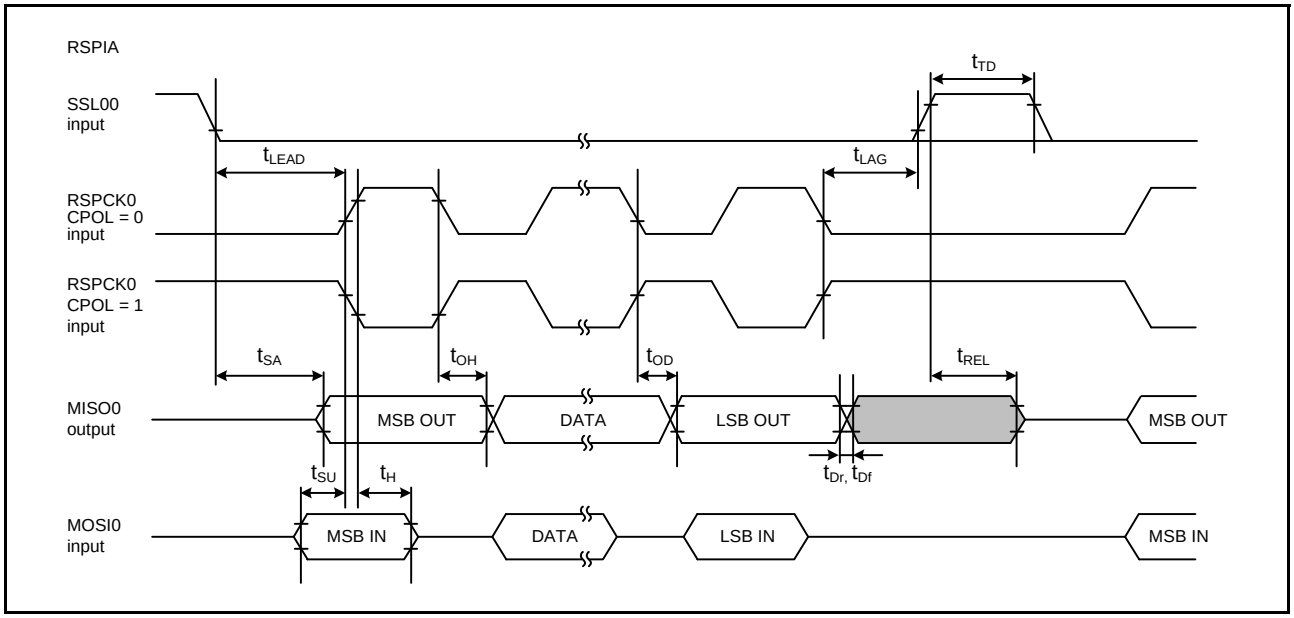


Figure 56.79 RSPIA Timing (Slave, Motorola SPI, CPHA = 0)

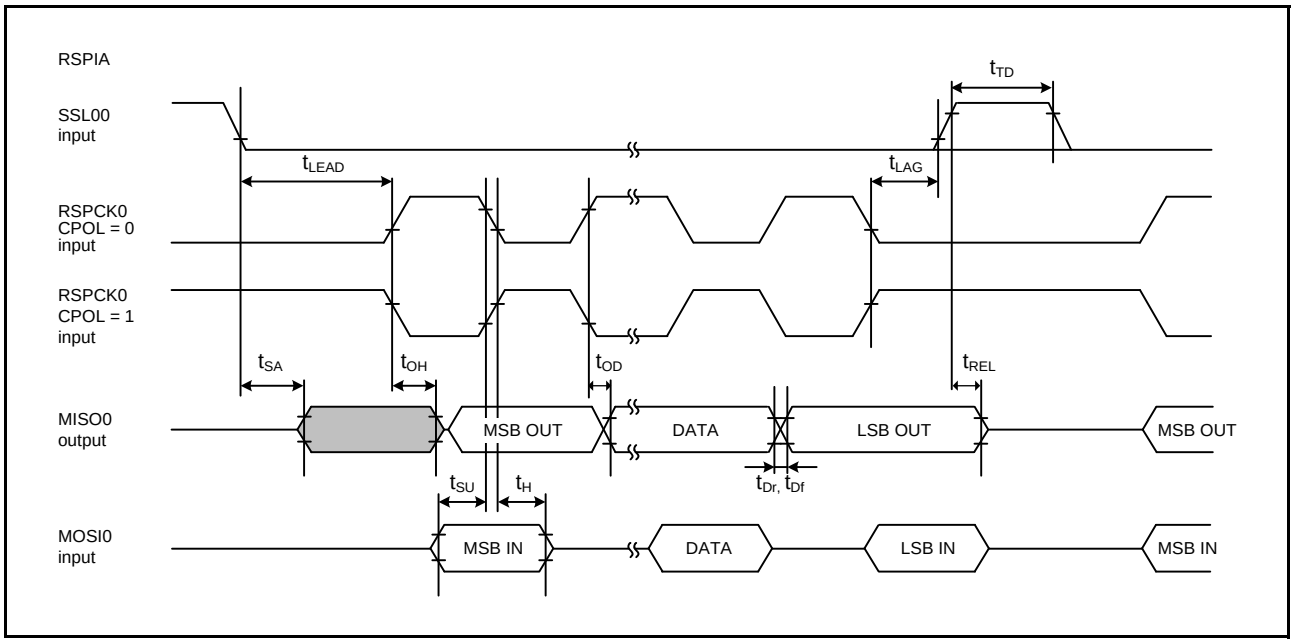


Figure 56.80 RSPIA Timing (Slave, Motorola SPI, CPHA = 1)

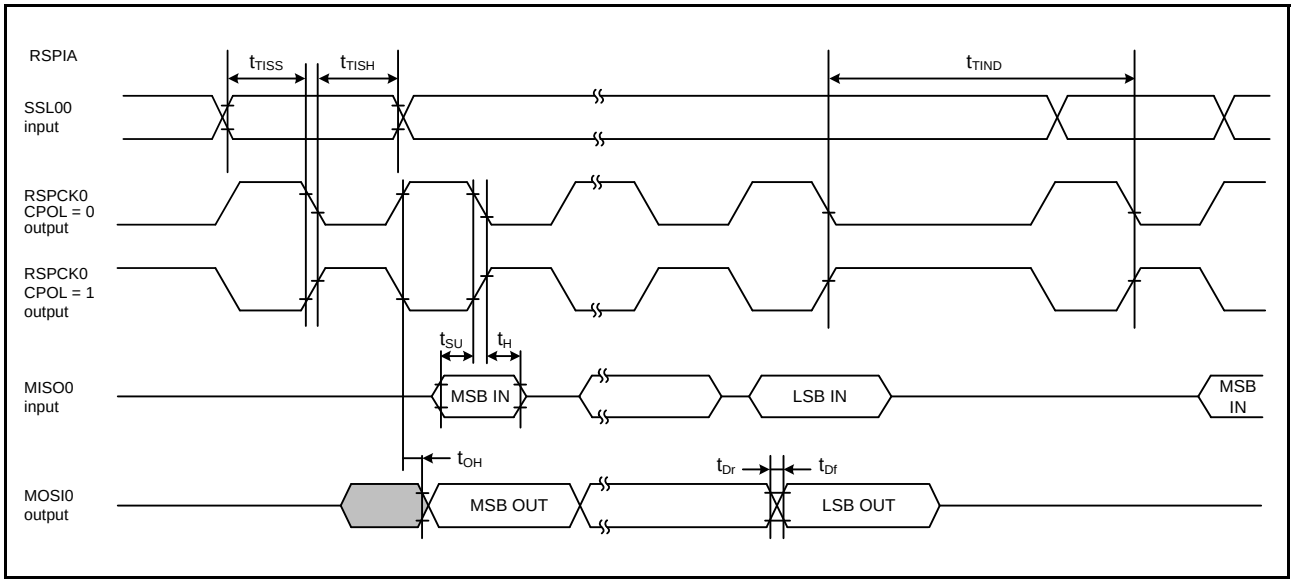


Figure 56.81 RSPIA Timing (Slave, TI SSP, Transmit with Delay between Frames)

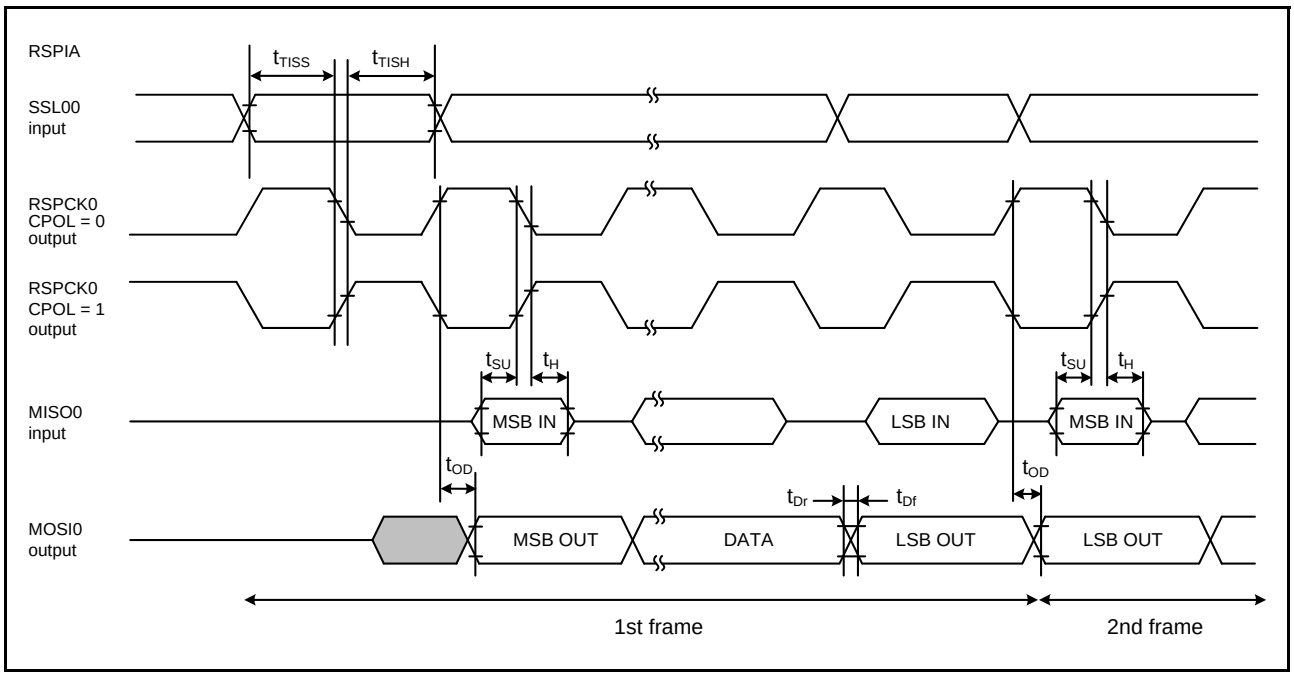


Figure 56.82 RSPIA Timing (Slave, TI SSP, Transmit with No Delay between Frames)

56.4.7.14 QSPIX

Table 56.45 QSPIX Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
ICLK = 8 to 120 MHz, PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-speed interface high-drive output is selected by the drive capacity control register.

Item		Symbol	Min.	Max.	Unit	Test Conditions
QSPIX	QSPCLK clock cycle	t_{QScyc}	2	48	t_{Icyc}*1	Figure 56.83
	QSPCLK clock high pulse width	t_{QSWH}	t_{QScyc} × 0.4	—	ns	
	QSPCLK clock low pulse width	t_{QSWL}	t_{QScyc} × 0.4	—	ns	
	Data input setup time	t_{Su}	8	—	ns	Figure 56.84
	Data input hold time	t_{IH}	0	—	ns	
	QSSL setup time	t_{LEAD}	(N + 0.5) × t_{QScyc} – 5*2	(N + 0.5) × t_{QScyc} + 100*2	ns	
	QSSL hold time	t_{LAG}	(N + 0.5) × t_{QScyc} – 5*3	(N + 0.5) × t_{QScyc} + 100*3	ns	
	Data output delay time	t_{OD}	—	4	ns	
	Data output hold time	t_{OH}	–3.3	—	ns	
	Successive transmission delay time	t_{TD}	1	16	t_{QScyc}	

Note 1. t_{Icyc}: ICLK cycle
Note 2. N: Value of the SPSSCR.SSSU bit (0 or 1)
Note 3. N: Value of the SPSSCR.SSHLD bit (0 or 1)

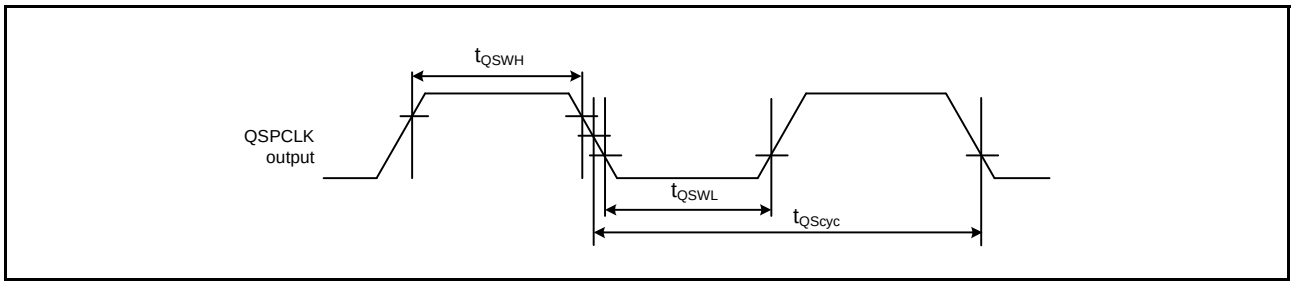


Figure 56.83 QSPIX Clock Timing

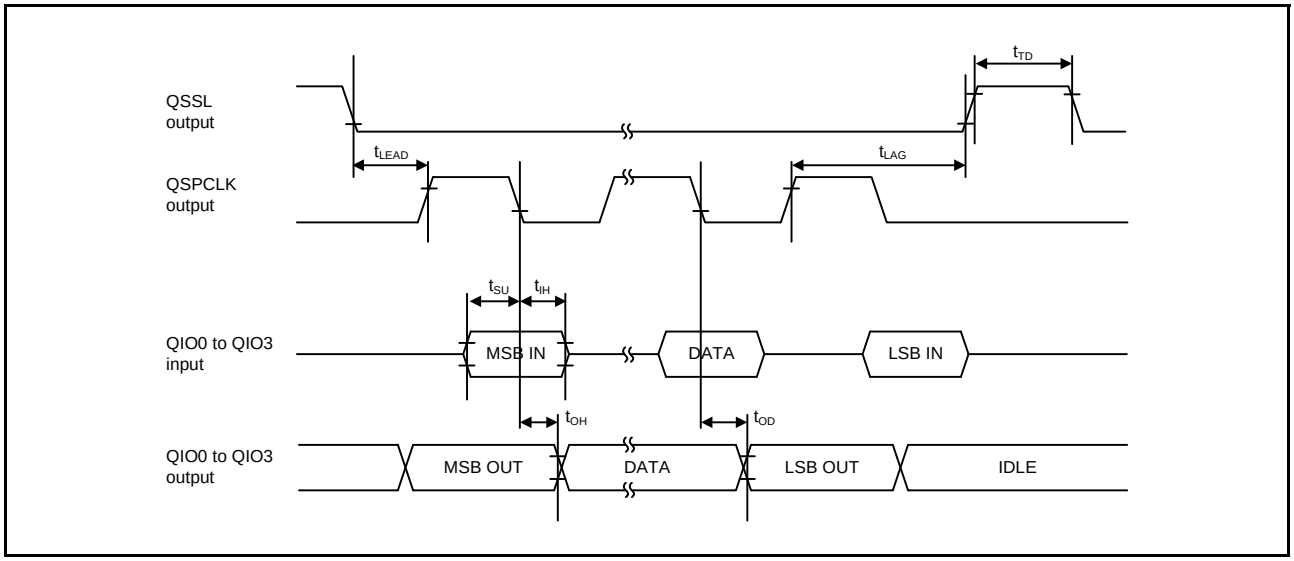


Figure 56.84 Transmit/Receive Timing

56.4.7.15 RIIC

Table 56.46 RIIC Timing (1)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$

Item		Symbol	Min.*1	Max.	Unit	Test Conditions
RIIC (Standard-mode, SMBus) ICFER.FMPE = 0	SCL input cycle time	t_{SCL}	6(12) × t_{IICcyc} + 1300	—	ns	Figure 56.85
	SCL input high pulse width	t_{SCLH}	3(6) × t_{IICcyc} + 300	—	ns	
	SCL input low pulse width	t_{SCLL}	3(6) × t_{IICcyc} + 300	—	ns	
	SCL, SDA input rise time	t_{Sr}	—	1000	ns	
	SCL, SDA input fall time	t_{Sf}	—	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	1(4) × t_{IICcyc}	ns	
	SDA input bus free time	t_{BUF}	3(6) × t_{IICcyc} + 300	—	ns	
	Start condition input hold time	t_{STAH}	t_{IICcyc} + 300	—	ns	
	Restart condition input setup time	t_{STAS}	1000	—	ns	
	Stop condition input setup time	t_{STOS}	1000	—	ns	
	Data input setup time	t_{SDAS}	t_{IICcyc} + 50	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b*2	—	400	pF	
RIIC (Fast-mode) ICFER.FMPE = 0	SCL input cycle time	t_{SCL}	6(12) × t_{IICcyc} + 600	—	ns	
	SCL input high pulse width	t_{SCLH}	3(6) × t_{IICcyc} + 300	—	ns	
	SCL input low pulse width	t_{SCLL}	3(6) × t_{IICcyc} + 300	—	ns	
	SCL, SDA input rise time	t_{Sr}	20 × (External pull-up voltage/5.5V)	300	ns	
	SCL, SDA input fall time	t_{Sf}	20 × (External pull-up voltage/5.5V)	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	1(4) × t_{IICcyc}	ns	
	SDA input bus free time	t_{BUF}	3(6) × t_{IICcyc} + 300	—	ns	
	Start condition input hold time	t_{STAH}	t_{IICcyc} + 300	—	ns	
	Restart condition input setup time	t_{STAS}	300	—	ns	
	Stop condition input setup time	t_{STOS}	300	—	ns	
	Data input setup time	t_{SDAS}	t_{IICcyc} + 50	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b*2	—	400	pF	

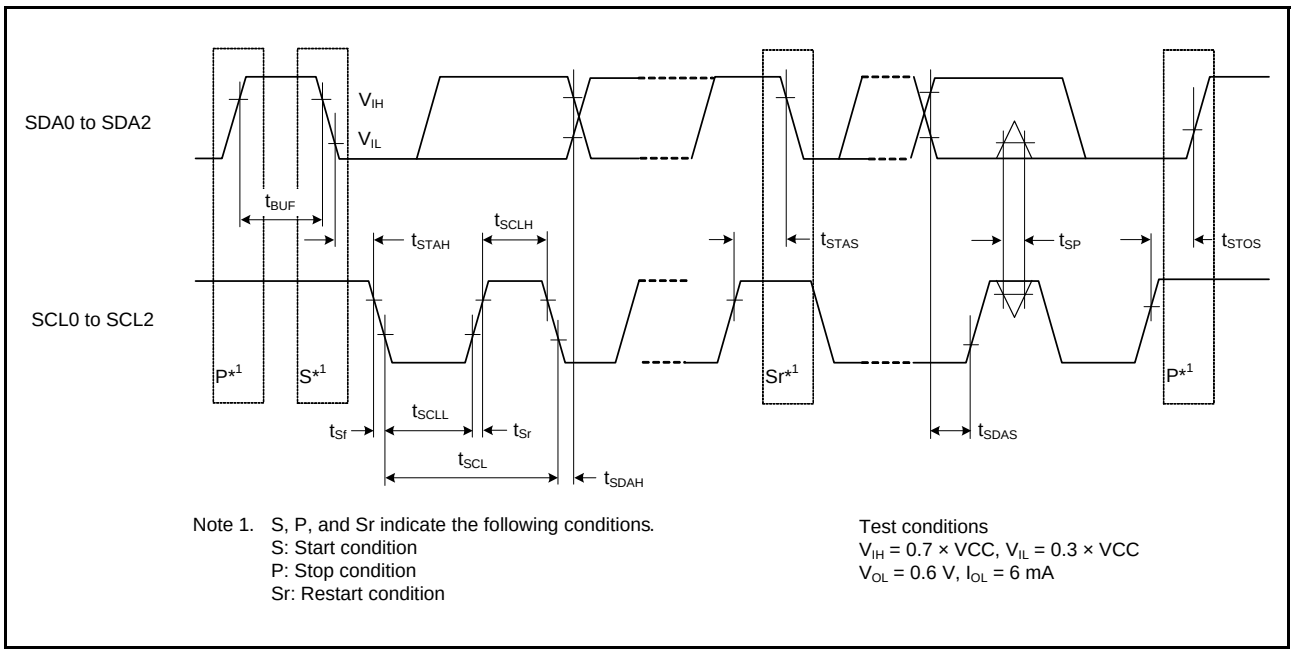
Note: t_{IICcyc}: RIIC internal reference clock (IICφ) cycle
Note 1. The value within parentheses is applicable when the value of the ICMR3.NF[1:0] bits is 11b while the digital filter is enabled by the setting ICFER.NFE = 1.
Note 2. C_b is the total capacitance of the bus lines.

Table 56.47 RIIC Timing (2)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$

Item		Symbol	Min.*1	Max.	Unit	Test Conditions
RIIC (Fast-mode+) ICFER.FMPE = 1	SCL input cycle time	t_{SCL}	6(12) × t_{IICcyc} + 240	—	ns	Figure 56.85
	SCL input high pulse width	t_{SCLH}	3(6) × t_{IICcyc} + 120	—	ns	
	SCL input low pulse width	t_{SCLL}	3(6) × t_{IICcyc} + 120	—	ns	
	SCL, SDA input rise time	t_{Sr}	—	120	ns	
	SCL, SDA input fall time	t_{Sf}	—	120	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	1(4) × t_{IICcyc}	ns	
	SDA input bus free time	t_{BUF}	3(6) × t_{IICcyc} + 120	—	ns	
	Start condition input hold time	t_{STAH}	t_{IICcyc} + 120	—	ns	
	Restart condition input setup time	t_{STAS}	120	—	ns	
	Stop condition input setup time	t_{STOS}	120	—	ns	
	Data input setup time	t_{SDAS}	t_{IICcyc} + 20	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b*2	—	550	pF	

Note: t_{IICcyc}: RIIC internal reference clock (IICφ) cycle
Note 1. The value within parentheses is applicable when the value of the ICMR3.NF[1:0] bits is 11b while the digital filter is enabled by the setting ICFER.NFE = 1.
Note 2. C_b is the total capacitance of the bus lines.



Note 1. S, P, and Sr indicate the following conditions.
S: Start condition
P: Stop condition
Sr: Restart condition

Test conditions
V_{IH} = 0.7 × VCC, V_{IL} = 0.3 × VCC
V_{OL} = 0.6 V, I_{OL} = 6 mA

Figure 56.85 RIIC Bus Interface Input/Output Timing

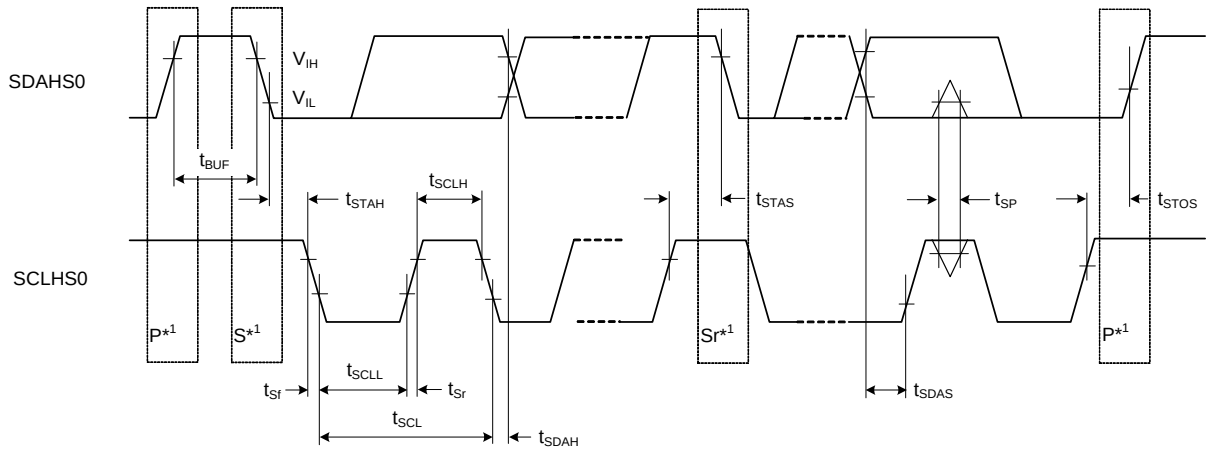
56.4.7.16 RIICHS

Table 56.48 RIICHS Timing (1)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$

Item		Symbol	Min.*1	Max.	Unit	Test Conditions
RIICHS (Standard-mode, SMBus) ICFER.FMPE = 0	SCLHS input cycle time	t_{SCL}	10(18) × t_{IICcyc} + 1300	—	ns	Figure 56.86
	SCLHS input high pulse width	t_{SCLH}	5(9) × t_{IICcyc} + 300	—	ns	
	SCLHS input low pulse width	t_{SCLL}	5(9) × t_{IICcyc} + 300	—	ns	
	SCLHS, SDAHS input rise time	t_{Sr}	—	1000	ns	
	SCLHS, SDAHS input fall time	t_{Sf}	—	300	ns	
	SCLHS, SDAHS input spike pulse removal time	t_{SP}	0	1(4) × t_{IICcyc}	ns	
	SDAHS input bus free time	t_{BUF}	5(9) × t_{IICcyc} + 300	—	ns	
	Start condition input hold time	t_{STAH}	t_{IICcyc} + 300	—	ns	
	Restart condition input setup time	t_{STAS}	1000	—	ns	
	Stop condition input setup time	t_{STOS}	1000	—	ns	
	Data input setup time	t_{SDAS}	t_{IICcyc} + 50	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCLHS, SDAHS capacitive load	C_b*2	—	400	pF	
RIICHS (Fast-mode) ICFER.FMPE = 0	SCLHS input cycle time	t_{SCL}	10(18) × t_{IICcyc} + 600	—	ns	
	SCLHS input high pulse width	t_{SCLH}	5(9) × t_{IICcyc} + 300	—	ns	
	SCLHS input low pulse width	t_{SCLL}	5(9) × t_{IICcyc} + 300	—	ns	
	SCLHS, SDAHS input rise time	t_{Sr}	20 × (External pull-up voltage/5.5V)	300	ns	
	SCLHS, SDAHS input fall time	t_{Sf}	20 × (External pull-up voltage/5.5V)	300	ns	
	SCLHS, SDAHS input spike pulse removal time	t_{SP}	0	1(4) × t_{IICcyc}	ns	
	SDAHS input bus free time	t_{BUF}	5(9) × t_{IICcyc} + 300	—	ns	
	Start condition input hold time	t_{STAH}	t_{IICcyc} + 300	—	ns	
	Restart condition input setup time	t_{STAS}	300	—	ns	
	Stop condition input setup time	t_{STOS}	300	—	ns	
	Data input setup time	t_{SDAS}	t_{IICcyc} + 50	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCLHS, SDAHS capacitive load	C_b*2	—	400	pF	
RIICHS (Fast-mode+) ICFER.FMPE = 1	SCLHS input cycle time	t_{SCL}	10(18) × t_{IICcyc} + 240	—	ns	
	SCLHS input high pulse width	t_{SCLH}	5(9) × t_{IICcyc} + 120	—	ns	
	SCLHS input low pulse width	t_{SCLL}	5(9) × t_{IICcyc} + 120	—	ns	
	SCLHS, SDAHS input rise time	t_{Sr}	—	120	ns	
	SCLHS, SDAHS input fall time	t_{Sf}	—	120	ns	
	SCLHS, SDAHS input spike pulse removal time	t_{SP}	0	1(4) × t_{IICcyc}	ns	
	SDAHS input bus free time	t_{BUF}	5(9) × t_{IICcyc} + 120	—	ns	
	Start condition input hold time	t_{STAH}	t_{IICcyc} + 120	—	ns	
	Restart condition input setup time	t_{STAS}	120	—	ns	
	Stop condition input setup time	t_{STOS}	120	—	ns	
	Data input setup time	t_{SDAS}	t_{IICcyc} + 20	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCLHS, SDAHS capacitive load	C_b*2	—	550	pF	

Note: t_{IICcyc}: RIICHS internal reference clock (IICφ) cycle
Note 1. The value within parentheses is applicable when the value of the ICICR.NF[3:0] bits is 0011b while the digital filter is enabled by the setting ICICR.NFE = 1.
Note 2. C_b is the total capacitance of the bus lines.



Note 1. S, P, and Sr indicate the following conditions.
S: Start condition
P: Stop condition
Sr: Restart condition

Test conditions
V_{IH} = 0.7 × VCC, V_{IL} = 0.3 × VCC
V_{OL} = 0.6 V, I_{OL} = 6 mA

Figure 56.86 RIICHS Bus Interface Input/Output Timing

Table 56.49 RIICHS Timing (2)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$

Item			Symbol	Min.*1	Max.	Unit	Test Conditions
RIICHS (Hs-mode) ICFER.HSME = 1	SCLHS input cycle time		t_{SCL}	10(12) × t_{IICcyc} + 80	—	ns	Figure 56.87
	SCLHS input high pulse width		t_{SCLH}	5(6) × t_{IICcyc}	—	ns	
	SCLHS input low pulse width		t_{SCLL}	5(6) × t_{IICcyc}	—	ns	
	SCLHS input rise time	C_b = 400pF	t_{SrCL}	—	80	ns	
		C_b = 100pF		—	40		
	SDAHS input rise time	C_b = 400pF	t_{SrDA}	—	160	ns	
		C_b = 100pF		—	80		
	SCLHS input fall time	C_b = 400pF	t_{SfCL}	—	80	ns	
		C_b = 100pF		—	40		
	SDAHS input fall time	C_b = 400pF	t_{SfDA}	—	160	ns	
		C_b = 100pF		—	80		
	SCLHS, SDAHS input spike pulse removal time		t_{SP}	0	1(1) × t_{IICcyc}	ns	Figure 56.86
	SDAHS input bus free time		t_{BUF}	5(6) × t_{IICcyc} + 40	—	ns	
	Start condition input hold time		t_{STAH}	t_{IICcyc} + 40	—	ns	Figure 56.87
	Restart condition input setup time		t_{STAS}	40	—	ns	
	Stop condition input setup time		t_{STOS}	40	—	ns	
	Data input setup time		t_{SDAS}	10	—	ns	
	Data input hold time	C_b = 400pF	t_{SDAH}	0	150	ns	
		C_b = 100pF		0	70		
	SCLHS, SDAHS capacitive load		C_b*2	—	400	pF	

Note: t_{IICcyc}: RIICHS internal reference clock (IICφ) cycle

Note 1. The value within parentheses is applicable when the value of the ICICR.NF[3:0] bits is 0011b while the digital filter is enabled by the setting ICICR.NFE = 1. Note that, in Hs-mode, the lower-order 2 bits of the NF[3:0] bits are ignored and the number of stages in the digital filter is single.

Note 2. C_b is the total capacitance of the bus lines.

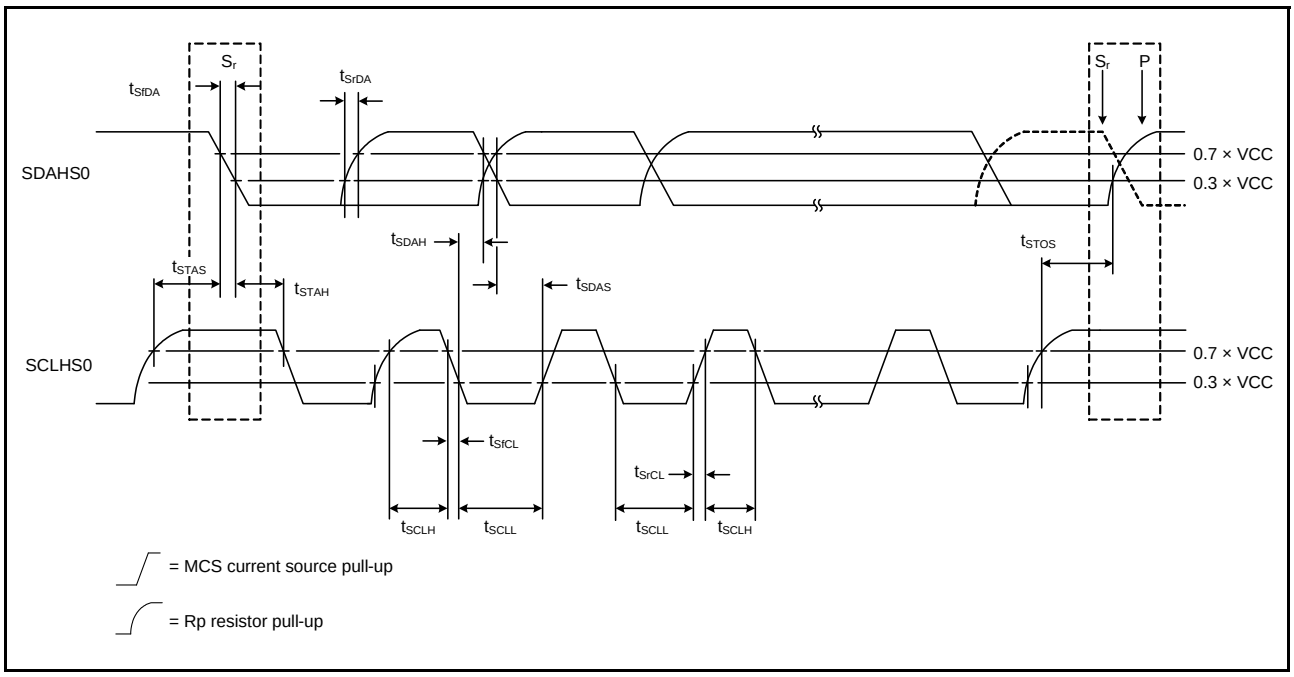


Figure 56.87 RIICHS Bus Interface Input/Output Timing (Hs-mode)

56.4.7.17 SDHI

Table 56.50 SDHI Timing

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the drive capacity control register.

Item		Symbol	Min.	Max.	Unit	Test Conditions*1
SDHI	SDHI_CLK pin output cycle time	$t_{PP(SD)}$	20	—	ns	Figure 56.88
	SDHI_CLK pin output high pulse width	$t_{WH(SD)}$	0.4 × $t_{PP(SD)}$	—	ns	
	SDHI_CLK pin output low pulse width	$t_{WL(SD)}$	0.4 × $t_{PP(SD)}$	—	ns	
	SDHI_CLK pin output rise time	$t_{TLH(SD)}$	—	3	ns	
	SDHI_CLK pin output fall time	$t_{THL(SD)}$	—	3	ns	
	Output data delay time (data transfer mode) for SDHI_CMD and SDHI_D0 to SDHI_D3 pins	$t_{ODLY(SD)}$	–6.5	4	ns	
	Input data setup time for SDHI_CMD and SDHI_D0 to SDHI_D3 pins	$t_{ISU(SD)}$	6	—	ns	
	Input data hold time for SDHI_CMD and SDHI_D0 to SDHI_D3 pins	$t_{IH(SD)}$	2	—	ns	

Note 1. When a letter “-A”, “-B”, etc. to indicate group membership is appended to the pin name, each pin is recommended to use in combination with the pins in the same group. All SDHI AC timings are measured in combination with the pins in the same group.

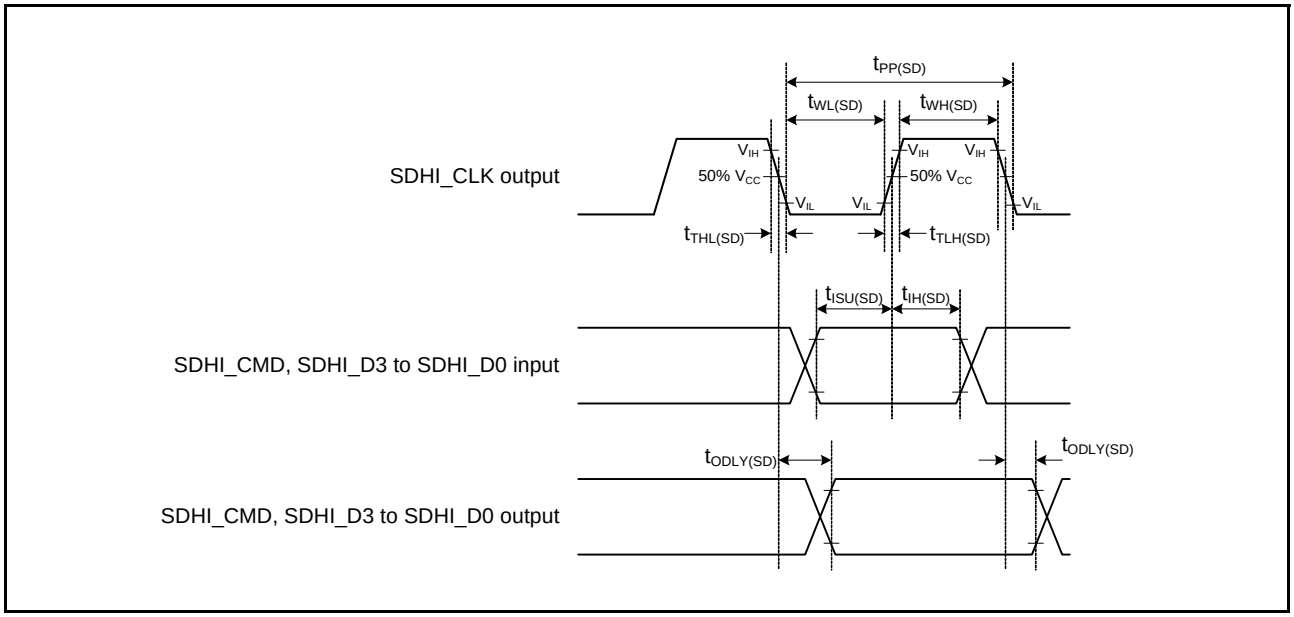


Figure 56.88 SD Host Interface Input/Output Signal Timing

56.5 USB Characteristics

Table 56.51 On-Chip USB Low Speed (Host Only) Characteristics (DP and DM Pin Characteristics)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 3.0 to 3.6 V, 3.0 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
UCLK = 48 MHz,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input characteristics	Input high level voltage	V_{IH}	2.0	—	—	V	
	Input low level voltage	V_{IL}	—	—	0.8	V	
	Differential input sensitivity	V_{DI}	0.2	—	—	V	| DP – DM |
	Differential common mode range	V_{CM}	0.8	—	2.5	V	
Output characteristics	Output high level voltage	V_{OH}	2.8	—	3.6	V	I_{OH} = –200 µA
	Output low level voltage	V_{OL}	0.0	—	0.3	V	I_{OL} = 2 mA
	Cross-over voltage	V_{CRS}	1.3	—	2.0	V	Figure 56.89
	Rise time	t_{LR}	75	—	300	ns	
	Fall time	t_{LF}	75	—	300	ns	
	Rise/fall time ratio	t_{LR} / t_{LF}	80	—	125	%	t_{LR}/ t_{LF}
Pull-down characteristics	DP/DM pull-down resistance (when the host controller function is selected)	R_{pd}	14.25	—	24.80	kΩ	

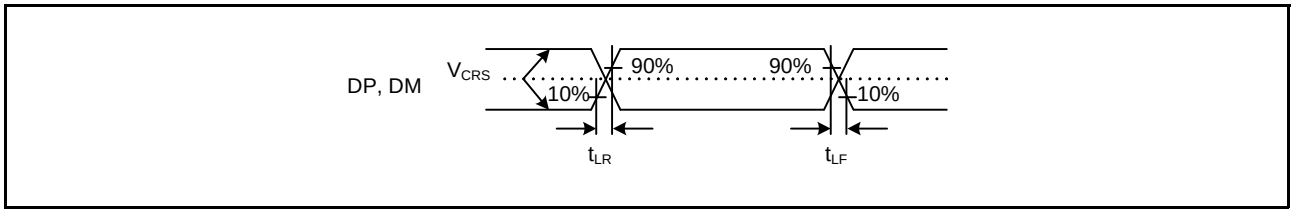


Figure 56.89 DP and DM Output Timing (Low Speed)

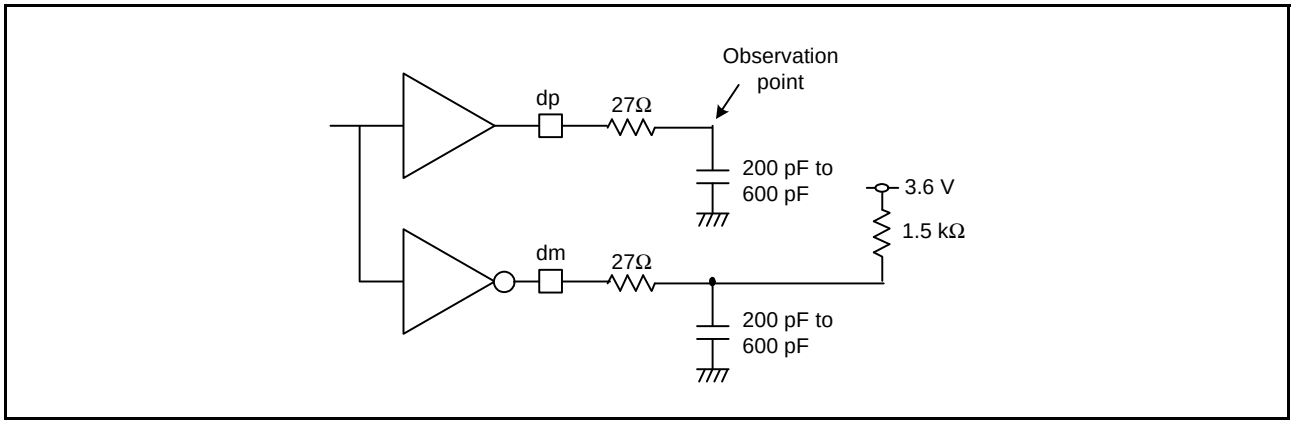


Figure 56.90 Test Circuit (Low Speed)

Table 56.52 On-Chip USB Full-Speed Characteristics (DP and DM Pin Characteristics)

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 3.0 to 3.6 V, 3.0 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
UCLK = 48 MHz,
PCLKA = 8 to 120 MHz, PCLKB = 8 to 60 MHz, $T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input characteristics	Input high level voltage	V_{IH}	2.0	—	—	V	
	Input low level voltage	V_{IL}	—	—	0.8	V	
	Differential input sensitivity	V_{DI}	0.2	—	—	V	| DP – DM |
	Differential common mode range	V_{CM}	0.8	—	2.5	V	
Output characteristics	Output high level voltage	V_{OH}	2.8	—	3.6	V	I_{OH} = –200 µA
	Output low level voltage	V_{OL}	0.0	—	0.3	V	I_{OL} = 2 mA
	Cross-over voltage	V_{CRS}	1.3	—	2.0	V	Figure 56.91
	Rise time	t_{FR}	4	—	20	ns	
	Fall time	t_{FF}	4	—	20	ns	
	Rise/fall time ratio	t_{FR} / t_{FF}	90	—	111.11	%	t_{FR}/ t_{FF}
	Output resistance	Z_{DRV}	28	—	44	Ω	Rs = 27 Ω included
Pull-up and pull-down characteristics	DP pull-up resistance (when the function controller function is selected)	R_{pu}	0.900	—	1.575	kΩ	Idle state
			1.425	—	3.090		At transmission and reception
	DP/DM pull-down resistance (when the host controller function is selected)	R_{pd}	14.25	—	24.80	kΩ	

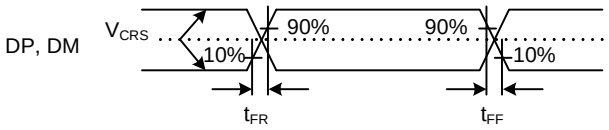


Figure 56.91 DP and DM Output Timing (Full-Speed)

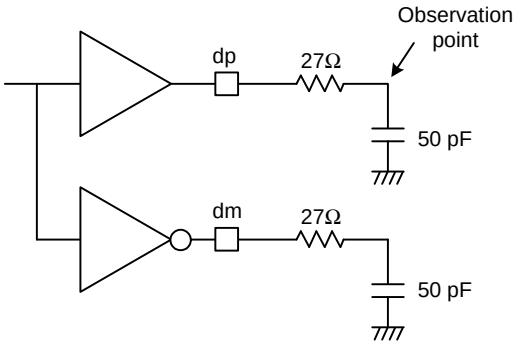


Figure 56.92 Test Circuit (Full-Speed)

56.6 A/D Conversion Characteristics

Table 56.53 12-Bit A/D (Unit 0) Conversion Characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKB = PCLKC = 1 MHz to 60 MHz, $T_a = T_{opr}$,
Source impedance = 1.0 kΩ

Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	8	—	12	Bit	
Analog input capacitance	—	—	30	pF	
Conversion time*1 (Operation at PCLKC = 60 MHz)	0.48 (0.267)*2	—	—	µs	Sampling in 16 states
Offset error	—	±1.0	±2.5	LSB	
Full-scale error	—	±1.0	±2.5	LSB	
Quantization error	—	±0.5	—	LSB	
Absolute accuracy	—	±2.5	±4.5	LSB	
DNL differential nonlinearity error	—	±0.5	±1.5	LSB	
INL integral nonlinearity error	—	±1.0	±2.5	LSB	

Note: The above specification values apply when there is no access to the external bus during A/D conversion. If access proceeds during A/D conversion, values may not fall within the above ranges.
Note 1. The conversion time includes the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.
Note 2. The value in parentheses indicates the sampling time.

Table 56.54 12-Bit A/D (Unit 1) Conversion Characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKB = PCLKD = 1 MHz to 60 MHz, $T_a = T_{opr}$,
Source impedance = 1.0 kΩ

Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	8	—	12	Bit	
Conversion time*1 (Operation at PCLKD = 60 MHz)	0.88 (0.633)*2	—	—	µs	Sampling in 38 states (ADSAM.SAM = 1)
Conversion time*1 (Operation at PCLKD = 30 MHz)	1 (0.500)*2	—	—	µs	Sampling in 15 states (ADSAM.SAM = 1)
Analog input capacitance	—	—	30	pF	
Offset error	—	±2.0	±3.5	LSB	
Full-scale error	—	±2.0	±3.5	LSB	
Quantization error	—	±0.5	—	LSB	
Absolute accuracy	—	±4.0	±6.0	LSB	
DNL differential nonlinearity error (Operation at PCLKD = 60 MHz)	—	±1.5	±4.0	LSB	
DNL differential nonlinearity error (Operation at PCLKD = 30 MHz)	—	±1.5	±2.5	LSB	
INL integral nonlinearity error (Operation at PCLKD = 60 MHz)	—	±2.0	±4.0	LSB	
INL integral nonlinearity error (Operation at PCLKD = 30 MHz)	—	±2.0	±3.5	LSB	

Note: The above specification values apply when there is no access to the external bus during A/D conversion. If access proceeds during A/D conversion, values may not fall within the above ranges.
Note 1. The conversion time includes the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.
Note 2. The value in parentheses indicates the sampling time.

Table 56.55 A/D Internal Reference Voltage Characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
PCLKB = PCLKD = 60 MHz, $T_a = T_{opr}$

Item	Min.	Typ.	Max.	Unit	Test Conditions
A/D internal reference voltage	1.13	1.18	1.23	V	

56.7 Temperature Sensor Characteristics

Table 56.56 Temperature Sensor Characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
$T_a = T_{opr}$

Item	Min.	Typ.	Max.	Unit	Test Conditions
Relative accuracy	—	±1	—	°C	
Temperature slope	—	4	—	mV/°C	
Output voltage	—	1.21	—	V	T_a = 25°C
Temperature sensor start time	—	—	30	µs	
Sampling time*1	4.15	—	—	µs	

Note 1. Set the S12AD1.ADSSTRT register such that the sampling time of the 12-bit A/D converter satisfies this specification.

56.8 CTSU Characteristics

Table 56.57 CTSU Characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
$T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
External capacitance connected to TSCAP pin	C_{tscap}	9	10	11	nF	
TS pin capacitive load	C_{base}	—	—	50	pF	
Total sum of the high-level output current*1	ΣI_{OH}	—	—	–40*2	mA	The mutual capacitance method is in use.

Note 1. Total sum of I_{OH} of the pins other than TSCAP, and TS0 to TS16

Note 2. In the mutual capacitance method, when the amount of current output from the I/O pins other than those of the CTSU is relatively large, the VCC voltage largely drops, affecting measurement by the CTSU. Accordingly, the total sum of I_{OH} of the other pins should be no more than the listed value when the CTSU is in use.

56.9 Power-on Reset Circuit and Voltage Detection Circuit Characteristics

Table 56.58 Power-on Reset Circuit and Voltage Detection Circuit Characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
$T_a = T_{opr}$

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Voltage detection level	Power-on reset (POR)	Low power consumption function disabled*1	V_{POR}	2.5	2.6	2.7	V	Figure 56.93
		Low power consumption function enabled*2		1.8	2.25	2.7		
	Voltage detection circuit (LVD0)		V_{det0_1}	2.84	2.94	3.04		Figure 56.94
			V_{det0_2}	2.77	2.87	2.97		
			V_{det0_3}	2.70	2.80	2.90		
	Voltage detection circuit (LVD1)		V_{det1_1}	2.89	2.99	3.09		Figure 56.95
			V_{det1_2}	2.82	2.92	3.02		
			V_{det1_3}	2.75	2.85	2.95		
	Voltage detection circuit (LVD2)		V_{det2_1}	2.89	2.99	3.09		Figure 56.96
			V_{det2_2}	2.82	2.92	3.02		
			V_{det2_3}	2.75	2.85	2.95		
Internal reset time	Power-on reset time		t_{POR}	—	4.6	—	ms	Figure 56.93
	LVD0 reset time		t_{LVD0}	—	0.70	—		Figure 56.94
	LVD1 reset time		t_{LVD1}	—	0.57	—		Figure 56.95
	LVD2 reset time		t_{LVD2}	—	0.57	—		Figure 56.96
Minimum VCC down time			t_{VOFF}	200	—	—	µs	Figure 56.93, Figure 56.94
Response delay time			t_{det}	—	—	200	µs	Figure 56.93 to Figure 56.96
LVD operation stabilization time (after LVD is enabled)			$t_{d(E-A)}$	—	—	10	µs	Figure 56.95, Figure 56.96
Hysteresis width (LVD1 and LVD2)			V_{LVH}	—	70	—	mV	

Note: The minimum VCC down time indicates the time when VCC is below the minimum value of voltage detection levels V_{POR}, V_{det1}, and V_{det2} for the POR/ LVD.
Note 1. The low power consumption function is disabled and DEEPCUT[1:0] = 00b or 01b.
Note 2. The low power consumption function is enabled and DEEPCUT[1:0] = 11b.

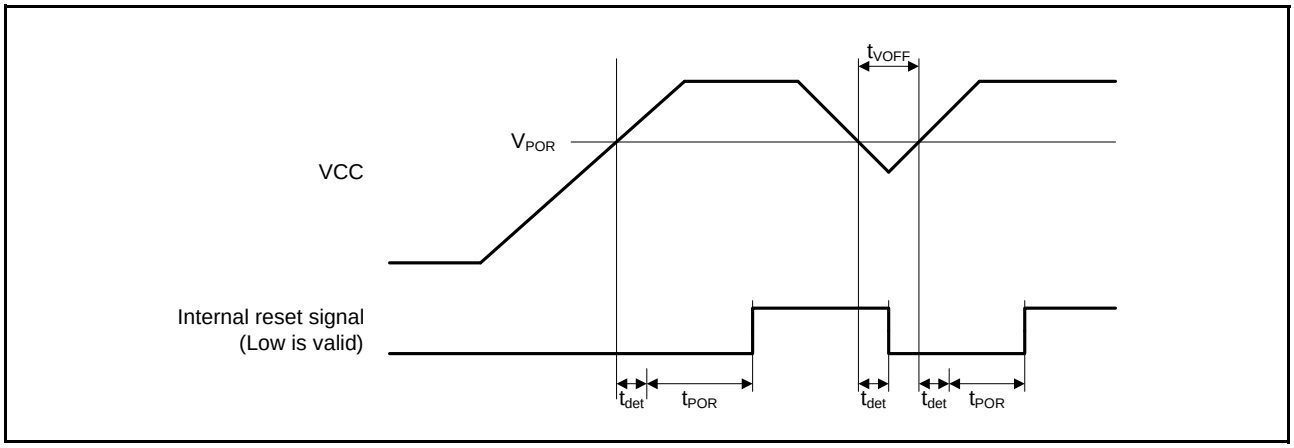


Figure 56.93 Power-on Reset Timing

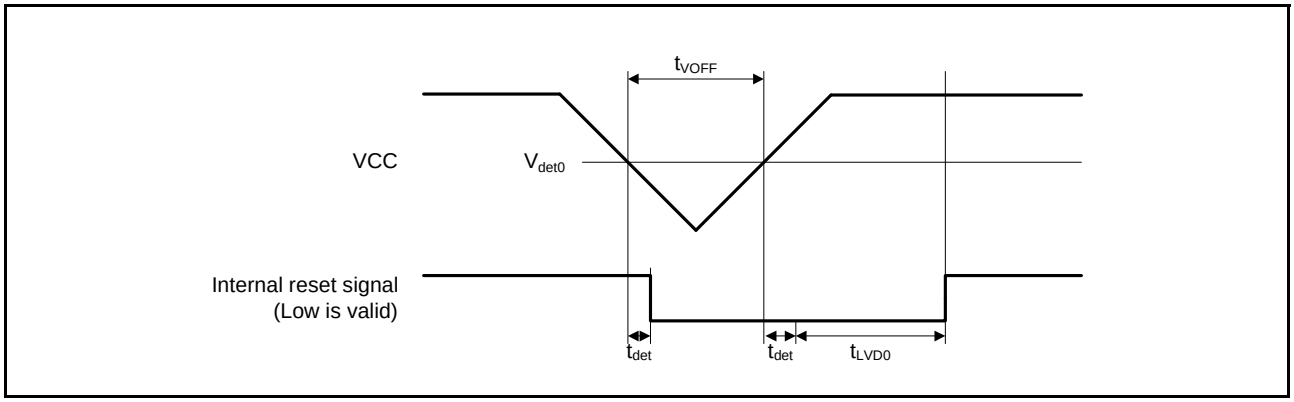


Figure 56.94 Voltage Detection Circuit Timing (V_{det0})

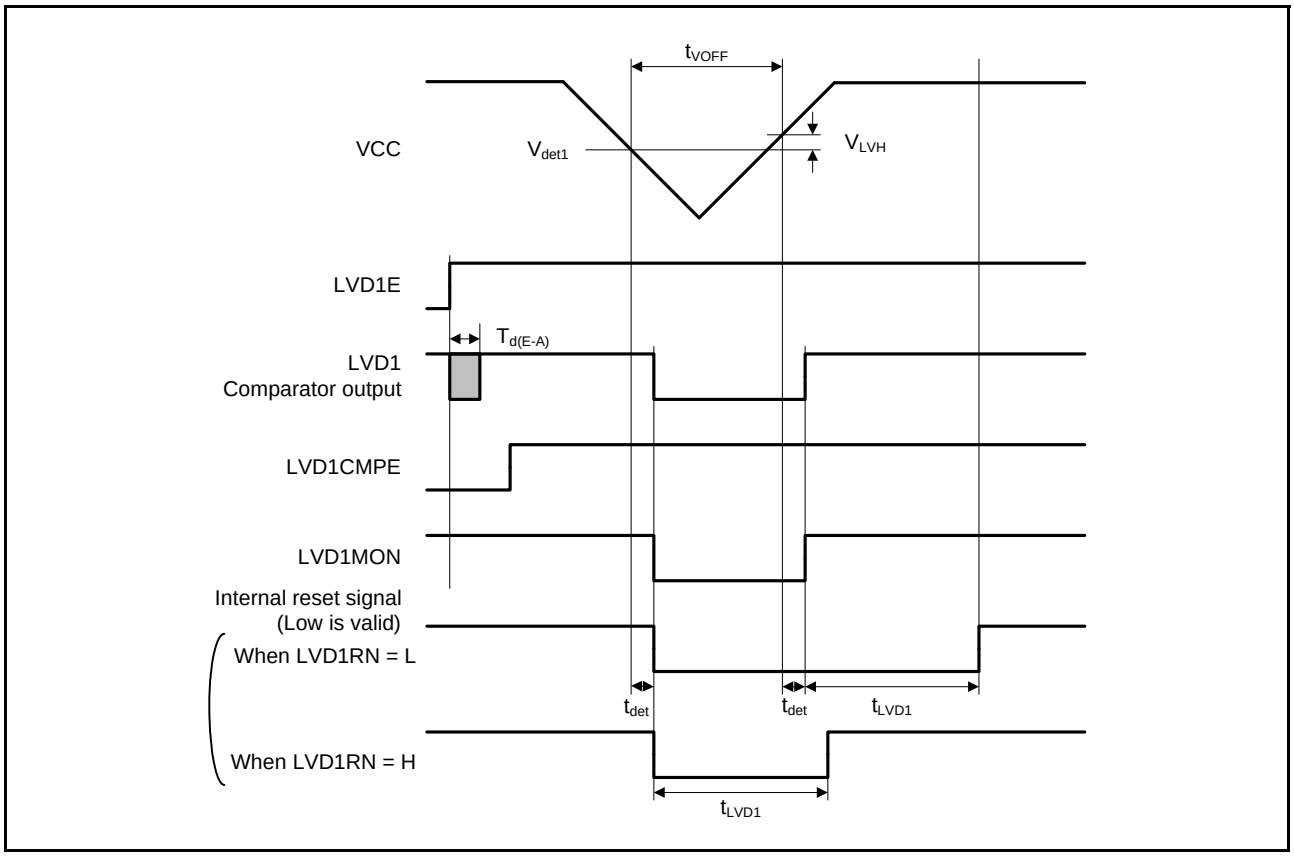


Figure 56.95 Voltage Detection Circuit Timing (V_{det1})

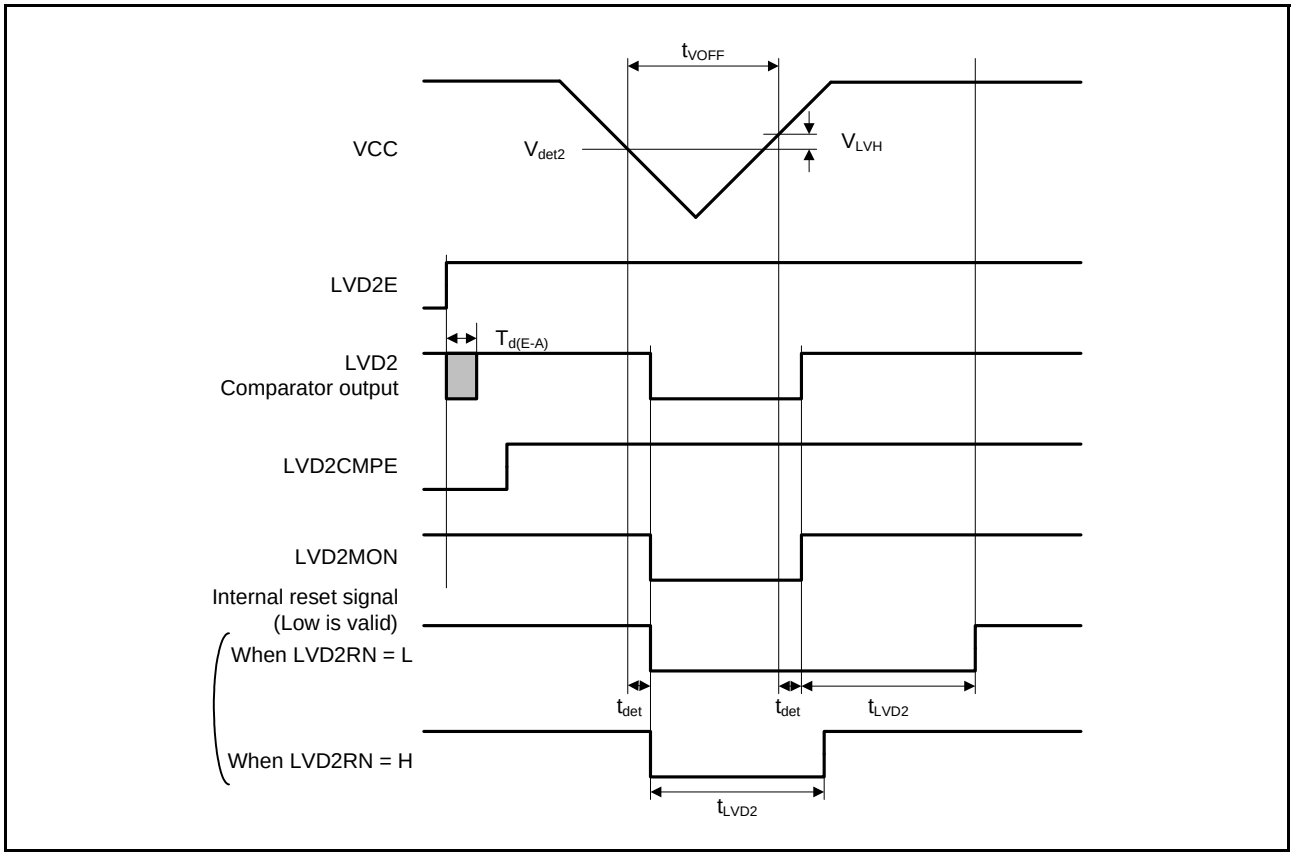


Figure 56.96 Voltage Detection Circuit Timing (V_{det2})

56.10 Oscillation Stop Detection Timing

Table 56.59 Oscillation Stop Detection Circuit Characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
$T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Detection time	t_{dr}	—	—	1	ms	Figure 56.97

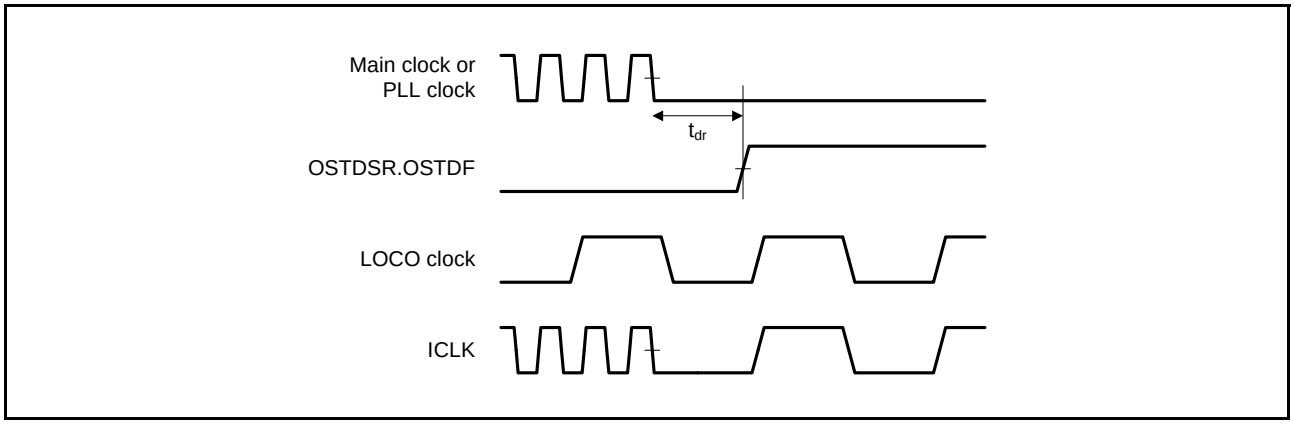


Figure 56.97 Oscillation Stop Detection Timing

56.11 Battery Backup Function Characteristics

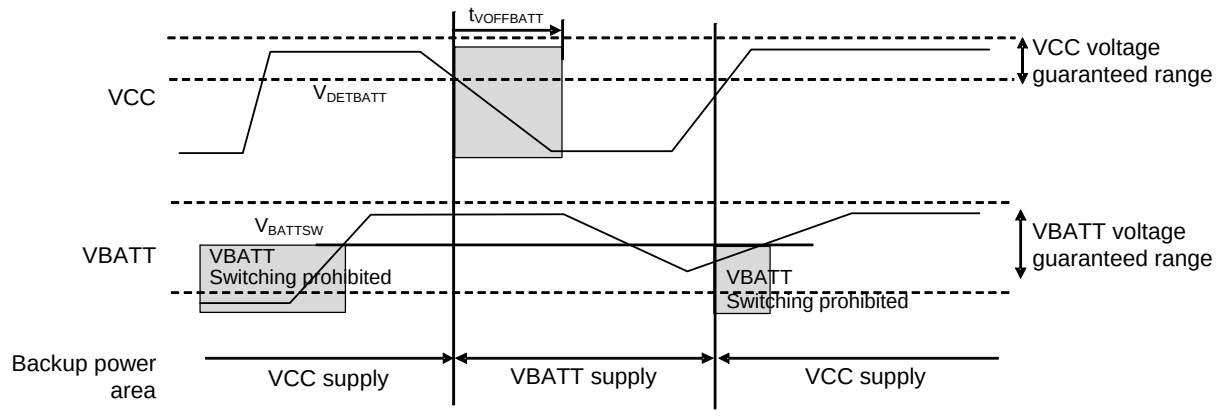
Table 56.60 Battery Backup Function Characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
V_{BATT} = 1.62 to 3.6 V, $T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Voltage level for switching to battery backup	$V_{DETBATT}$	2.50	2.60	2.70	V	Figure 56.98
Lower-limit V_{BATT} voltage for power supply switching due to VCC voltage drop	V_{BATTSW}	2.00	—	—	V	
VCC-off period for starting power supply switching*1	$t_{VOFFBATT}$	200	—	—	µs	
Backup domain power-down detection level	$V_{PDR(BKP)}$	1.45	1.5	1.55	V	Figure 56.99
Time delay in assertion of the reset signal for the backup domain*2	$t_{p(PDRL)}$	—	—	2000	µs	
Time delay in negation of the reset signal for the backup domain	$t_{p(PDRH)}$	—	—	1000	µs	
Temper input pulse width	$t_{w(TAMPI)}$	200	—	—	ns	Figure 56.100

Note 1. The VCC-off period for switching power supply indicates the period from VCC falling below the minimum value of the battery backup switching threshold voltage ($V_{DETBATT}$) until the source of supply is switched to VBATT. When the VCC recovers within this period, the source may not be switched to VBATT and supply from VCC is continued.

Note 2. When the V_{BKP} recovers within this period, the backup domain reset signal may not be generated.



Note. The VBATT voltage when the supplied power source switches from VCC to VBATT should not be lower than V_{BATTSW}, the lower-limit VBATT voltage for switching between power supplies due to a drop in the VCC voltage.

Figure 56.98 Battery Backup Function Characteristics

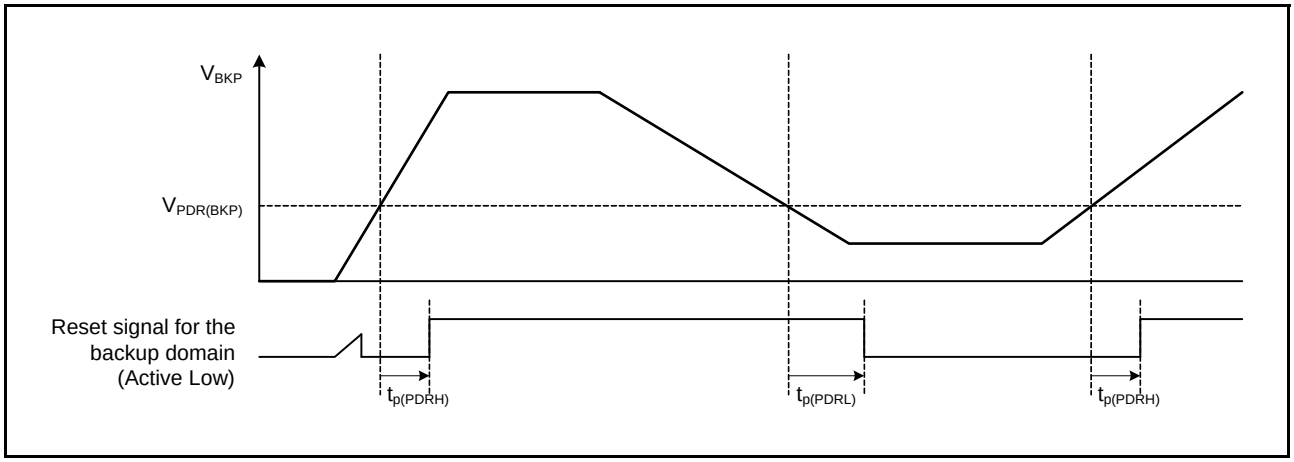


Figure 56.99 Backup Domain Reset Characteristics

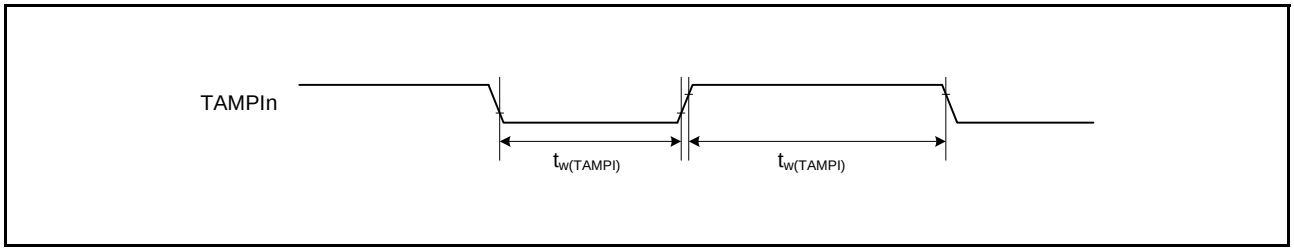


Figure 56.100 TAMPIn Input Timing

56.12 Flash Memory Characteristics

Table 56.61 Code Flash Memory Characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
Temperature range for programming/erasure: $T_a = T_{opr}$

Item		Symbol	FCLK = 4 MHz			FCLK = 15 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test Conditions
			Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.		
Programming time N_{PEC} ≤ 100 times	128 bytes	t_{P128}	—	0.75	13.2	—	0.38	6.6	—	0.34	6	ms	
	8 Kbytes	t_{P8K}	—	49	176	—	25	88	—	22	80	ms	
	32 Kbytes	t_{P32K}	—	194	704	—	97	352	—	88	320	ms	
Programming time N_{PEC} > 100 times	128 bytes	t_{P128}	—	0.91	15.8	—	0.46	8	—	0.41	7.2	ms	
	8 Kbytes	t_{P8K}	—	60	212	—	30	106	—	27	96	ms	
	32 Kbytes	t_{P32K}	—	234	848	—	117	424	—	106	384	ms	
Erasure time N_{PEC} ≤ 100 times	8 Kbytes	t_{E8K}	—	78	216	—	48	132	—	43	120	ms	
	32 Kbytes	t_{E32K}	—	283	864	—	173	528	—	157	480	ms	
Erasure time N_{PEC} > 100 times	8 Kbytes	t_{E8K}	—	94	260	—	58	158	—	52	144	ms	
	32 Kbytes	t_{E32K}	—	341	1040	—	208	632	—	189	576	ms	
Reprogramming/erasure cycle*1		N_{PEC}	10000 *2	—	—	10000 *2	—	—	10000 *2	—	—	Times	
Suspend delay time during programming		t_{SPD}	—	—	264	—	—	132	—	—	120	µs	
First suspend delay time during erasing (in suspend priority mode)		t_{SESD1}	—	—	216	—	—	132	—	—	120	µs	
Second suspend delay time during erasure (in suspend priority mode)		t_{SESD2}	—	—	1.7	—	—	1.7	—	—	1.7	ms	
Suspend delay time during erasure (in erasure priority mode)		t_{SEED}	—	—	1.7	—	—	1.7	—	—	1.7	ms	
Forced stop command		t_{FD}	—	—	32	—	—	22	—	—	20	µs	
Data hold time*3, *4		t_{DRP}	20	—	—	20	—	—	20	—	—	Year	T_a ≤ 85°C
			10	—	—	10	—	—	10	—	—		T_a ≤ 105°C

Note 1. Definition of program/erase cycle:
The program/erase cycle is the number of erasing for each block. When the number of program/erase cycles is n, each block can be erased n times. For instance, when 128-byte program is performed 64 times for different addresses in 8-Kbyte block and then the block is erased, the program/erase cycle is counted as one. However, the same address cannot be programmed more than once before the next erase cycle (overwriting is prohibited).

Note 2. Characteristics are degraded as the number of program/erase increases. This is the minimum value of program/erase cycles to guarantee all characteristics listed in this table.

Note 3. This shows the characteristic when the flash memory writer or self-programming library from Renesas Electronics is in use, and the number of times programming and erasure proceed does not exceed the specified value.

Note 4. These values are based on the results of reliability testing.

Table 56.62 Data Flash Memory Characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
Temperature range for programming/erasure: $T_a = T_{opr}$

Item		Symbol	FCLK = 4 MHz			FCLK = 15 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test Conditions
			Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.		
Programming time	4 bytes	t_{DP4}	—	0.36	3.8	—	0.18	1.9	—	0.16	1.7	ms	
Erasure time	64 bytes	t_{DP64}	—	3.1	18	—	1.9	11	—	1.7	10	ms	
	128 bytes	t_{DP128}	—	4.7	27	—	2.9	16	—	2.6	15	ms	
	256 bytes	t_{DP256}	—	8.9	50	—	5.4	31	—	4.9	28	ms	
Blank check time	4 bytes	t_{DBC4}	—	—	84	—	—	33	—	—	30	µs	
	64 bytes	t_{DBC64}	—	—	280	—	—	110	—	—	100	µs	
	2 Kbytes	t_{DBC2K}	—	—	6160	—	—	2420	—	—	2200	µs	
Reprogramming/erasure cycle*1		N_{DPEC}	100000 *2	—	—	100000 *2	—	—	100000 *2	—	—	Times	
Suspend delay time during programming		t_{DSPD}	—	—	264	—	—	132	—	—	120	µs	
First suspend delay time during erasure (in suspend priority mode)	64 bytes	—	—	—	216	—	—	132	—	—	120	µs	
	128 bytes	—	—	—	216	—	—	132	—	—	120	µs	
	256 bytes	—	—	—	216	—	—	132	—	—	120	µs	
Second suspend delay time during erasure (in suspend priority mode)	64 bytes	—	—	—	300	—	—	300	—	—	300	µs	
	128 bytes	—	—	—	390	—	—	390	—	—	390	µs	
	256 bytes	—	—	—	570	—	—	570	—	—	570	µs	
Suspend delay time during erasing (in erasure priority mode)	64 bytes	—	—	—	300	—	—	300	—	—	300	µs	
	128 bytes	—	—	—	390	—	—	390	—	—	390	µs	
	256 bytes	—	—	—	570	—	—	570	—	—	570	µs	
Forced stop command		t_{FD}	—	—	32	—	—	22	—	—	20	µs	
Data hold time*3, *4		t_{DDRP}	20	—	—	20	—	—	20	—	—	Year	$T_a \leq 85°C$
			10	—	—	10	—	—	10	—	—		$T_a \leq 105°C$

Note 1. Definition of program/erase cycle:
The program/erase cycle is the number of erasing for each block. When the number of program/erase cycles is n, each block can be erased n times. For instance, when 4-byte program is performed 512 times for different addresses in 2-Kbyte block and then the block is erased, the program/erase cycle is counted as one. However, the same address cannot be programmed more than once before the next erase cycle (overwriting is prohibited).

Note 2. Characteristics are degraded as the number of program/erase increases. This is the minimum value of program/erase cycles to guarantee all characteristics listed in this table.

Note 3. This shows the characteristic when the flash memory writer or self-programming library from Renesas Electronics is in use, and the number of times programming and erasure proceed does not exceed the specified value.

Note 4. These values are based on the results of reliability testing.

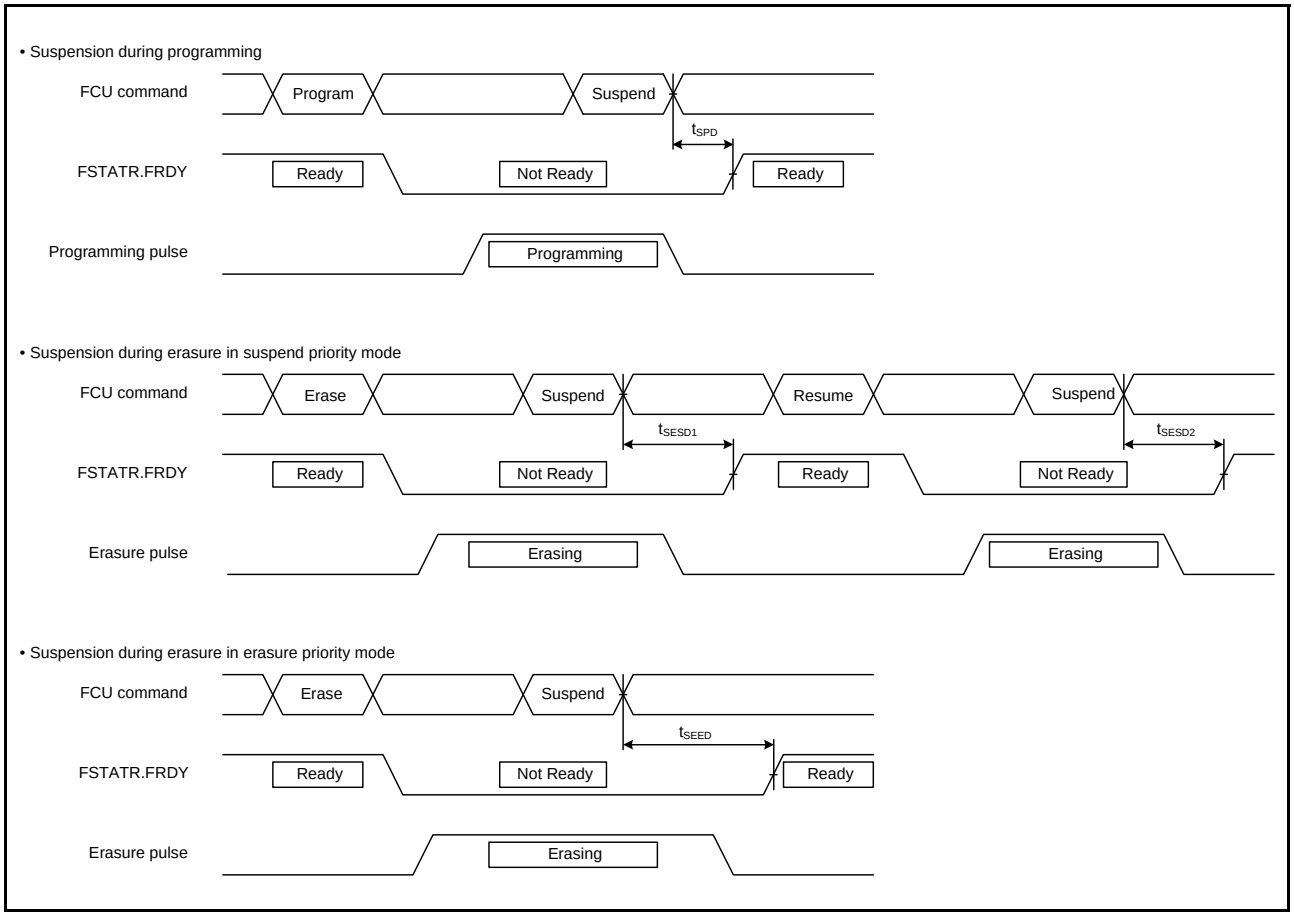


Figure 56.101 Flash Memory Programming/Erasure Suspension Timing

56.13 Boundary Scan

Table 56.63 Boundary Scan Characteristics

Conditions: VCC = AVCC0 = AVCC1 = VCC_USB = V_{BATT} = 2.7 to 3.6 V, 2.7 V ≤ VREFH0 ≤ AVCC0,
VSS = AVSS0 = AVSS1 = VREFL0 = VSS_USB = 0 V,
$T_a = T_{opr}$,
Output load conditions: V_{OH} = 0.5 × VCC, V_{OL} = 0.5 × VCC, C = 30 pF,
High-drive output is selected by the drive capacity control register.

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
TCK clock cycle time	t_{TCKcyc}	100	—	—	ns	Figure 56.102
TCK clock high pulse width	t_{TCKH}	45	—	—	ns	
TCK clock low pulse width	t_{TCKL}	45	—	—	ns	
TCK clock rise time	t_{TCKr}	—	—	5	ns	
TCK clock fall time	t_{TCKf}	—	—	5	ns	
TRST# pulse width	t_{TRSTW}	20	—	—	t_{TCKcyc}	Figure 56.103
TMS setup time	t_{TMSS}	20	—	—	ns	Figure 56.104
TMS hold time	t_{TMSH}	20	—	—	ns	
TDI setup time	t_{TDIS}	20	—	—	ns	
TDI hold time	t_{TDIH}	20	—	—	ns	
TDO data delay time	t_{TDOD}	—	—	40	ns	

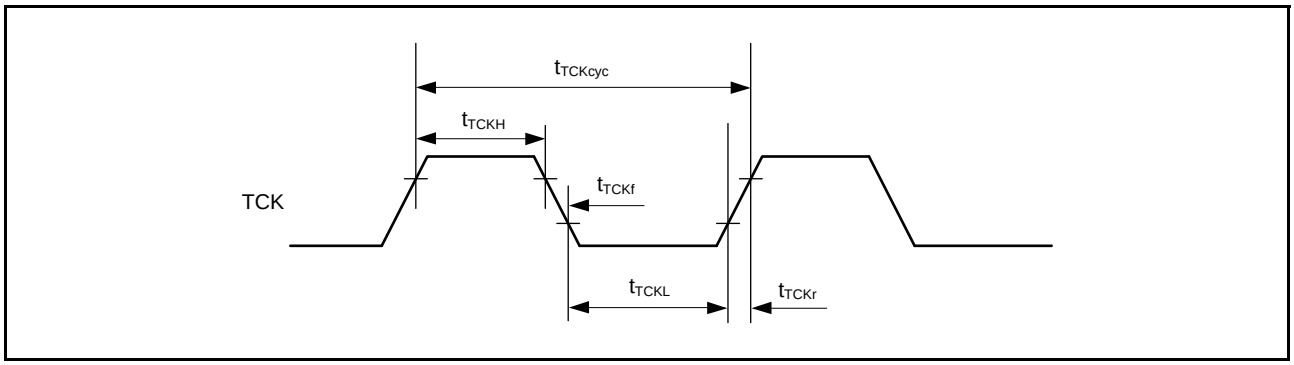


Figure 56.102 Boundary Scan TCK Timing

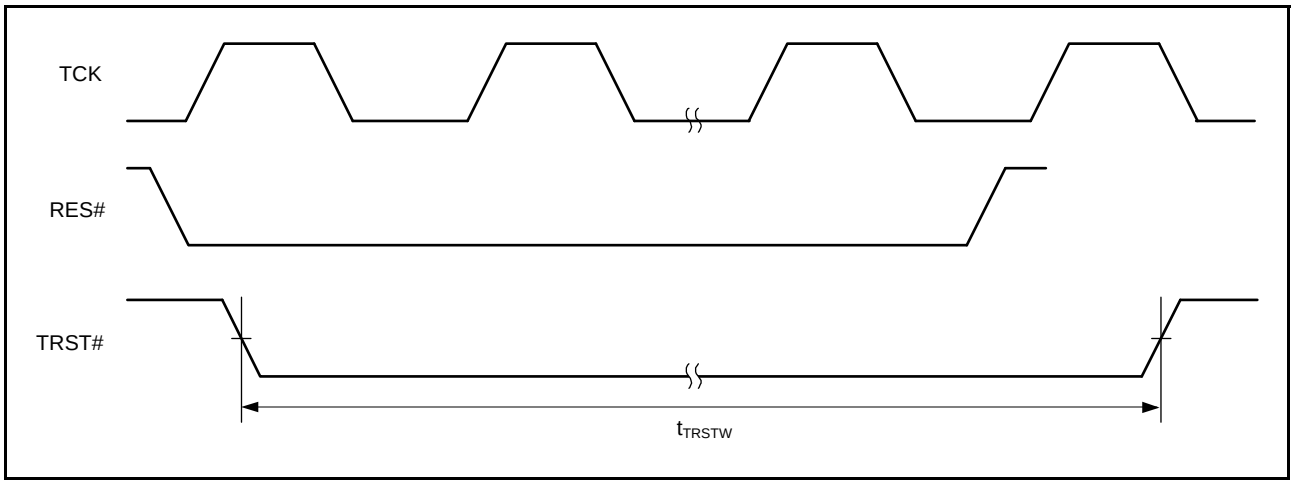


Figure 56.103 Boundary Scan TRST# Timing

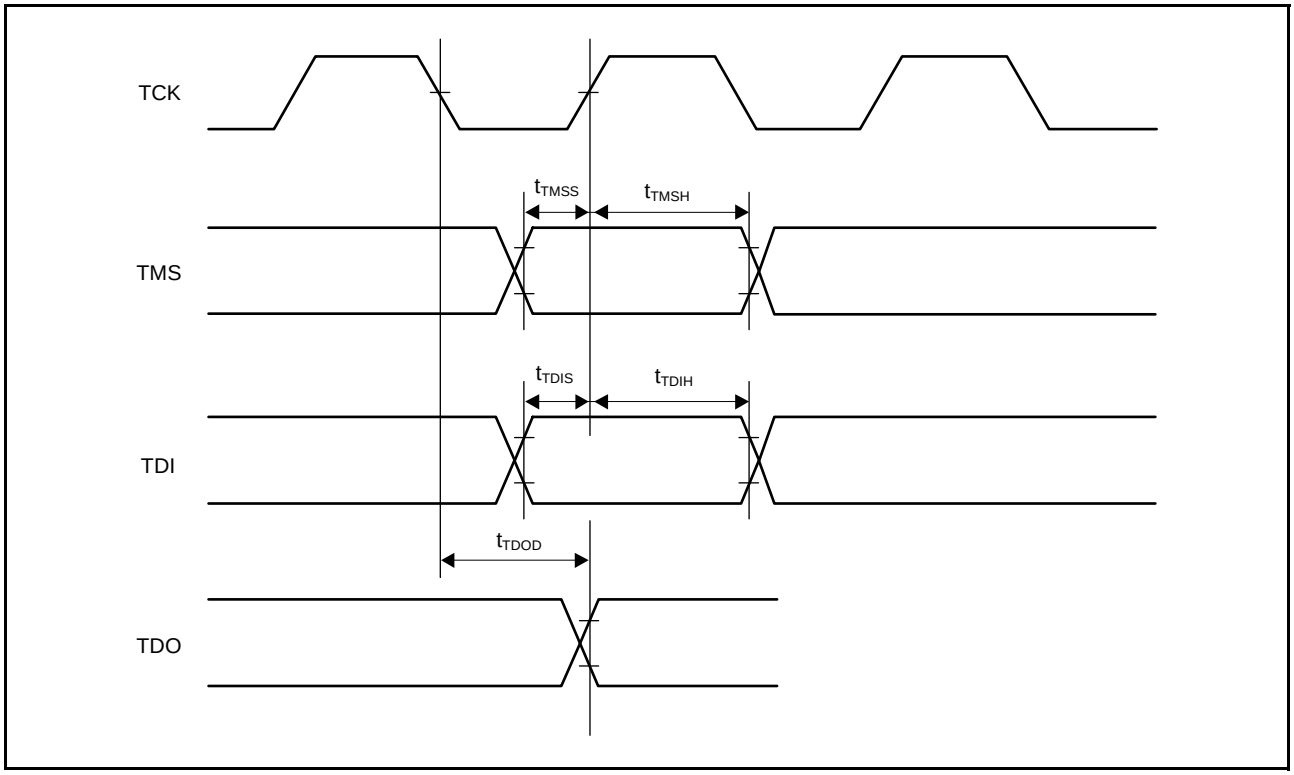


Figure 56.104 Boundary Scan Input/Output Timing

Appendix 1. Port States in Each Processing Mode

Table 1.1 Port States in Each Processing State (1 / 5)

Port Name Pin Name	Operating Mode According to Registers Setting	Reset	Software Standby Mode		Deep Software Standby Mode	After Deep Software Standby Mode is Canceled (Return to Start-up Mode)	
			OPE = 1	OPE = 0	IOKEEP = 1/0	IOKEEP = 1*1	IOKEEP = 0
P00/IRQ8, P01/IRQ9, P02/IRQ10, P03/IRQ11, P05/IRQ13, P07/IRQ15	All	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P12/IRQ2, P13/IRQ3, P14/IRQ4/USB0_ OVRCURA, P15/IRQ5/ CRX1-DS, P16/IRQ6/ SCL2-DS/ USB0_VBUS/ USB0_ OVRCURB, P17/IRQ7/ SDA2-DS	All	Hi-Z	Keep-O*2		Keep-O*3	Keep	Hi-Z
P20/IRQ8, P21/IRQ9, P22/IRQ15/ USB0_ OVRCURB, P23/IRQ3	All	Hi-Z	Keep-O*2		Keep-O*3	Keep	Hi-Z
P24/IRQ12/CS4#, P25/IRQ5/CS5#, P26/IRQ6/CS6#, P27/IRQ7/CS7#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended (EXBE = 1)		[CSn# output] H [Other than the above] Keep-O	[CSn# output] Hi-Z [Other than the above] Keep-O			
P30/IRQ0-DS/ RTCIC0/TAMPI0, P31/IRQ1-DS/ RTCIC1/TAMPI1, P32/IRQ2-DS/ RTCIC2/TAMPI2, P33/IRQ3-DS, 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-DS, P41/IRQ9-DS, P42/IRQ10-DS, P43/IRQ11-DS, P44/IRQ12-DS, P45/IRQ13-DS, P46/IRQ14-DS, P47/IRQ15-DS	All	Hi-Z	Keep-O*2		Keep-O*3	Keep	Hi-Z
P50/IRQ0/WR0#/ WR#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended (EXBE = 1)		[WR0#/WR# output] H [Other than the above] Keep-O	[WR0#/WR# output] Hi-Z [Other than the above] Keep-O			
P51/IRQ1/WR1#/ BC1#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O*2		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/IRQ2/RD#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended (EXBE = 1)		[RD# output] H [Other than the above] Keep-O	[RD# output] Hi-Z [Other than the above] Keep-O			
P53/IRQ3/BCLK/ PMC0-DS	All	Hi-Z	[Clock output] H [Other than the above] Keep-O*2		Keep*3	Keep	Hi-Z

Table 1.1 Port States in Each Processing State (2 / 5)

Port Name Pin Name	Operating Mode According to Registers Setting	Reset	Software Standby Mode OPE = 1	Software Standby Mode OPE = 0	Deep Software Standby Mode IOKEEP = 1/0	After Deep Software Standby Mode is Canceled (Return to Start-up Mode) IOKEEP = 1*1	After Deep Software Standby Mode is Canceled (Return to Start-up Mode) IOKEEP = 0
P54/IRQ4/ALE	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended (EXBE = 1)		[ALE output] L [Data output] Hi-Z [Other than the above] Keep-O	[ALE output] Hi-Z [Data output] Hi-Z [Other than the above] Keep-O			
P55/IRQ10	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				
P56/IRQ6	All	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P60/IRQ0/CS0#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended (EXBE = 1)		[CS0# output] H [Other than the above] Keep-O	[CS0# output] Hi-Z [Other than the above] Keep-O			
P61/IRQ1/CS1#/ SDCS#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended (EXBE = 1)		[CS1# output] H [Data output] Hi-Z*4 [Other than the above] Keep-O	[CS1# output] Hi-Z [Data output] Hi-Z*4 [Other than the above] Keep-O			
	Self-Refresh disabled (SDSELF. SFEN = 0)		[SDCS# output] H [Other than the above] Keep-O	[SDCS# output] Hi-Z [Other than the above] Keep-O			
	Self-Refresh enabled (SDSELF. SFEN = 1)		[SDCS# output] L [Other than the above] Keep-O				
P62/IRQ2/CS2#/ RAS#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended (EXBE = 1)		[CS2# output] H [Data output] Hi-Z*4 [Other than the above] Keep-O	[CS2# output] Hi-Z [Data output] Hi-Z*4 [Other than the above] Keep-O			
	Self-Refresh disabled (SDSELF. SFEN = 0)		[RAS# output] H [Other than the above] Keep-O	[RAS# output] Hi-Z [Other than the above] Keep-O			
	Self-Refresh enabled (SDSELF. SFEN = 1)		[RAS# output] L [Other than the above] Keep-O				
P63/IRQ3/CS3#/ CAS#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended (EXBE = 1)		[CS3# output] H [Data output] Hi-Z*4 [Other than the above] Keep-O	[CS3# output] Hi-Z [Data output] Hi-Z*4 [Other than the above] Keep-O			
	Self-Refresh disabled (SDSELF. SFEN = 0)		[CAS# output] H [Other than the above] Keep-O	[CAS# output] Hi-Z [Other than the above] Keep-O			
	Self-Refresh enabled (SDSELF. SFEN = 1)		[CAS# output] L [Other than the above] Keep-O				

Table 1.1 Port States in Each Processing State (3 / 5)

Port Name / Pin Name	Operating Mode According to Registers Setting	Reset	Software Standby Mode OPE = 1	Software Standby Mode OPE = 0	Deep Software Standby Mode IOKEEP = 1/0	After Deep Software Standby Mode is Canceled (Return to Start-up Mode) IOKEEP = 1*1	After Deep Software Standby Mode is Canceled (Return to Start-up Mode) IOKEEP = 0
P64/IRQ4/CS4#/WE#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[CS4# output] H [Data output] Hi-Z*4 [Other than the above] Keep-O	[CS4# output] Hi-Z [Data output] Hi-Z*4 [Other than the above] Keep-O			
	Self-Refresh disabled (SDSELF. SFEN = 0)		[WE# output] H [Other than the above] Keep-O	[WE# output] Hi-Z [Other than the above] Keep-O			
	Self-Refresh enabled (SDSELF. SFEN = 1)						
P65/IRQ13/CS5#/CKE	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[CS5# output] H [Other than the above] Keep-O	[CS5# output] Hi-Z [Other than the above] Keep-O			
	Self-Refresh disabled (SDSELF. SFEN = 0)		[CKE output] H [Other than the above] Keep-O	[CKE output] Hi-Z [Other than the above] Keep-O			
	Self-Refresh enabled (SDSELF. SFEN = 1)		[CKE output] L [Other than the above] Keep-O				
P66/IRQ14/CS6#/DQM0	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[CS6# output] H [Other than the above] Keep-O	[CS6# output] Hi-Z [Other than the above] Keep-O			
	Self-Refresh disabled (SDSELF. SFEN = 0)		[DQM0 output] DQM0 output retained [Other than the above] Keep-O	[DQM0 output] Hi-Z [Other than the above] Keep-O			
	Self-Refresh enabled (SDSELF. SFEN = 1)						
P67/CS7#/IRQ15/DQM1	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[CS7# output] H [Other than the above] Keep-O*2	[CS7# output] Hi-Z [Other than the above] Keep-O*2			
	Self-Refresh disabled (SDSELF. SFEN = 0)		[DQM1 output] DQM1 output retained [Other than the above] Keep-O*2	[DQM1 output] Hi-Z [Other than the above] Keep-O*2			
	Self-Refresh enabled (SDSELF. SFEN = 1)						
P70/IRQ0/SDCLK	All	Hi-Z	[Clock output] H [Other than the above] Keep-O*2		Keep	Keep	Hi-Z
P71/IRQ1/CS1#, P72/IRQ10/CS2#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[CSn# output] H [Address output] Address output retained [Other than the above] Keep-O	[CSn# output] Hi-Z [Address output] Hi-Z [Other than the above] Keep-O			

Table 1.1 Port States in Each Processing State (4 / 5)

Port Name Pin Name	Operating Mode According to Registers Setting	Reset	Software Standby Mode		Deep Software Standby Mode	After Deep Software Standby Mode is Canceled (Return to Start-up Mode)	
			OPE = 1	OPE = 0	IOKEEP = 1/0	IOKEEP = 1*1	IOKEEP = 0
P73/IRQ8/CS3#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended (EXBE = 1)		[CS3# output] H [Other than the above] Keep-O	[CS3# output] Hi-Z [Other than the above] Keep-O			
P74/IRQ12/CS4#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended (EXBE = 1)		[CS4# output] H [Address output] Address output retained [Other than the above] Keep-O	[CS4# output] Hi-Z [Address output] Hi-Z [Other than the above] Keep-O			
P75/IRQ13/CS5#, P76/IRQ14/CS6#, P77/IRQ7/CS7#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended (EXBE = 1)		[CSn# output] H [Other than the above] Keep-O	[CSn# output] Hi-Z [Other than the above] Keep-O			
P80/IRQ8, P81/IRQ9, P82/IRQ2, P83/IRQ3, P86/IRQ14, P87/IRQ15	All	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P90/IRQ0, P91/IRQ9, P92/IRQ10, P93/IRQ11	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 [Data output] Hi-Z [Other than the above] Keep-O	[Address output] Hi-Z [Data output] Hi-Z [Other than the above] Keep-O			
PA0/IRQ0, PA1/IRQ11	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	[Address output] Hi-Z [Other than the above] Keep-O			
PA2/IRQ10, PA3/IRQ6-DS, PA4/IRQ5-DS, PA5/IRQ5, PA6/IRQ14, PA7/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)		[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-DS, PB2/IRQ2, PB3/IRQ3/ PMC0-DS, PB4/IRQ4, PB5/IRQ13, PB6/IRQ6, PB7/IRQ15	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/IRQ10, PC3/IRQ11/ PMC0-DS, PC4/IRQ12/ CS3#, PC5/IRQ5/CS2#, PC6/IRQ13/ CS1#, PC7/IRQ14/CS0#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O*2		Keep*3	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended (EXBE = 1)		[Address output] Address output retained [CSn# output] H [Other than the above] Keep-O*2	[Address output] Hi-Z [CSn# output] Hi-Z [Other than the above] Keep-O*2			
PD0/IRQ0, PD1/IRQ1, PD2/IRQ2, PD3/IRQ3, PD4/IRQ4, PD5/IRQ5, PD6/IRQ6, PD7/IRQ7	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended (EXBE = 1)		[Data output] Hi-Z [Other than the above] Keep-O*2				

Table 1.1 Port States in Each Processing State (5 / 5)

Port Name Pin Name	Operating Mode According to Registers Setting		Reset	Software Standby Mode		Deep Software Standby Mode	After Deep Software Standby Mode is Canceled (Return to Start-up Mode)	
				OPE = 1	OPE = 0	IOKEEP = 1/0	IOKEEP = 1*1	IOKEEP = 0
PE0/IRQ8, PE1/IRQ9, PE2/IRQ7-DS, PE3/IRQ11, PE4/IRQ12, 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		[Data output] Hi-Z [Other than the above] Keep-O*2				
		16 bits in width of bus		[Data output] Hi-Z				
PF5/IRQ4	All		Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
PJ3/IRQ11, PJ5/IRQ13	All		Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
PH1/USB0_DP/ IRQ0	All		Hi-Z	Keep-O*2, *4		Hi-Z*3	Hi-Z	
PH2/USB0_DM/ IRQ1	All		Hi-Z	Keep-O*2, *4		Hi-Z*3	Hi-Z	
USB1_DM	All		Hi-Z	Keep-O*4		Hi-Z	Hi-Z	
USB1_DP	All		Hi-Z	Keep-O*4		Hi-Z	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.

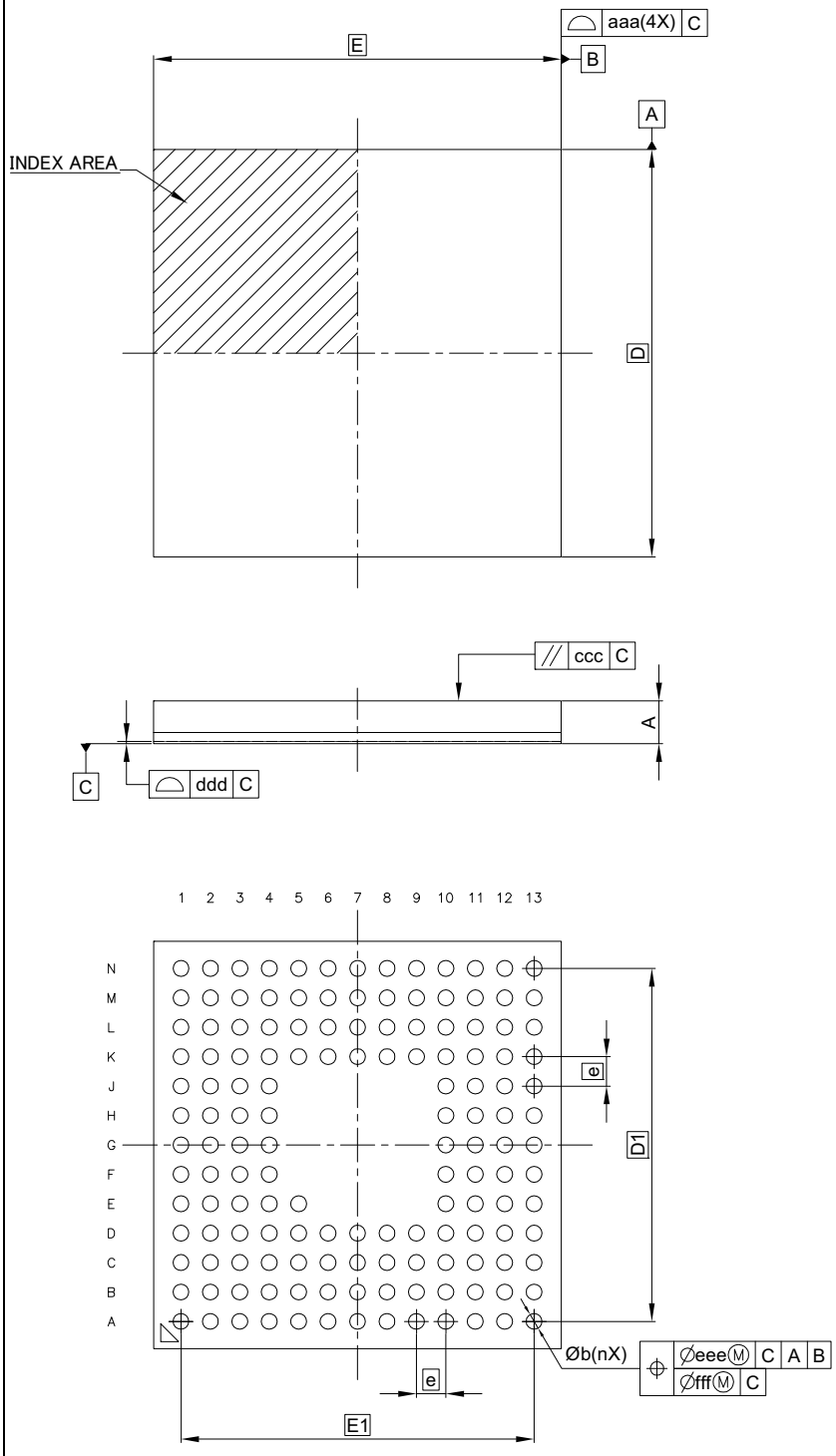
Hi-Z: High-impedance

Note 1. The I/O port state is retained 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 possible, but only when the pin is in use as a USB 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.

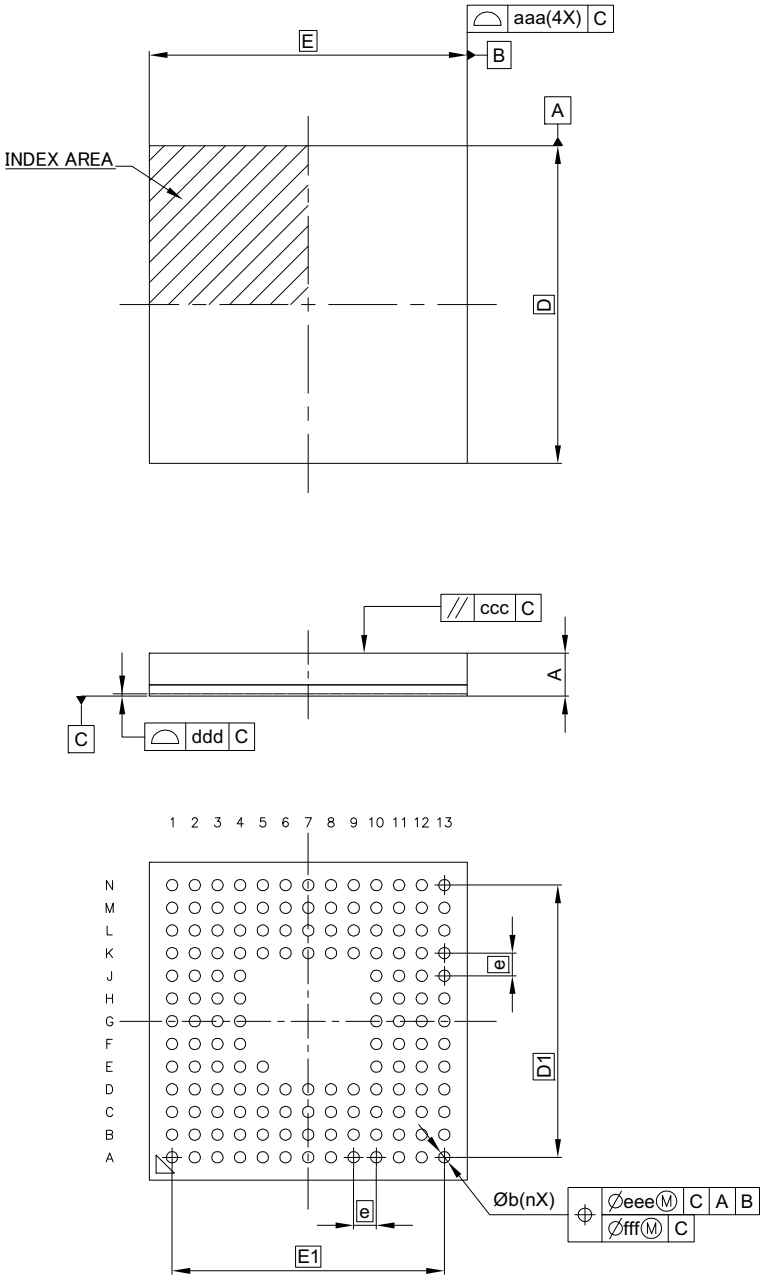
JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-TFLGA145-9×9-0.65	PTLG0145JC-A	0.16



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	—	9.00	—
E	—	9.00	—
D1	—	7.80	—
E1	—	7.80	—
A	—	—	1.01
b	—	0.35	—
e	—	0.65	—
aaa	—	—	0.10
ccc	—	—	0.10
ddd	—	—	0.10
eee	—	—	0.08
fff	—	—	0.08
n	—	145	—

Figure A 145-Pin TFLGA (PTLG0145JC-A)

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-TFLGA145-7×7-0.5	PTLG0145KB-A	0.09

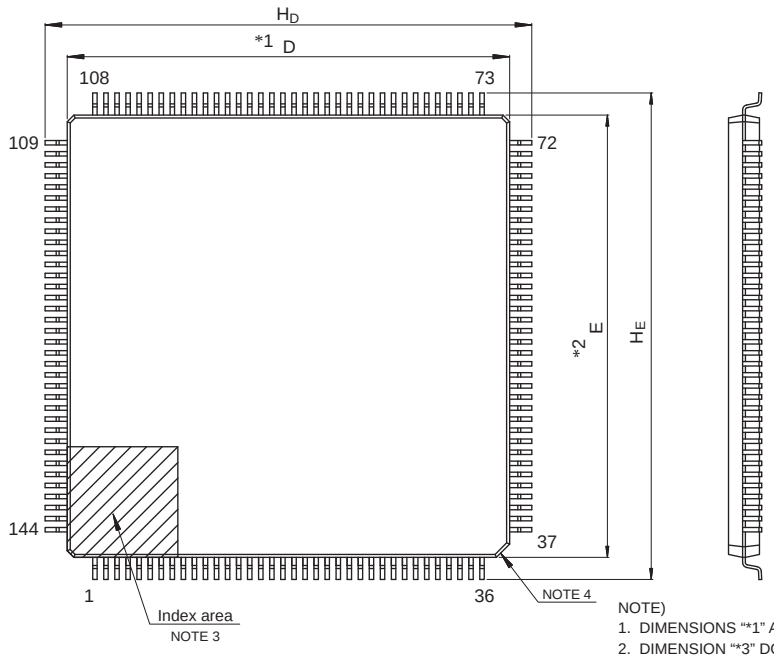


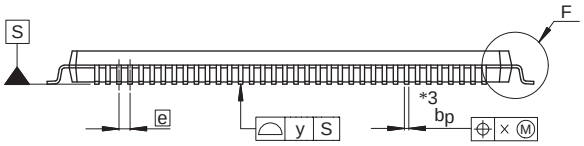
Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	—	7.00	—
E	—	7.00	—
D1	—	6.00	—
E1	—	6.00	—
A	—	—	1.01
b	—	0.25	—
e	—	0.50	—
aaa	—	—	0.10
ccc	—	—	0.10
ddd	—	—	0.08
eee	—	—	0.08
fff	—	—	0.05
n	—	145	—

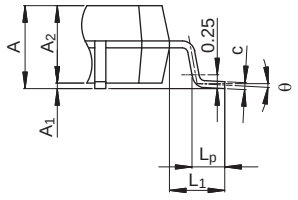
Figure B 145-Pin TFLGA (PTLG0145KB-A)

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP144-20x20-0.50	PLQP0144KA-B	—	1.2

Unit: mm







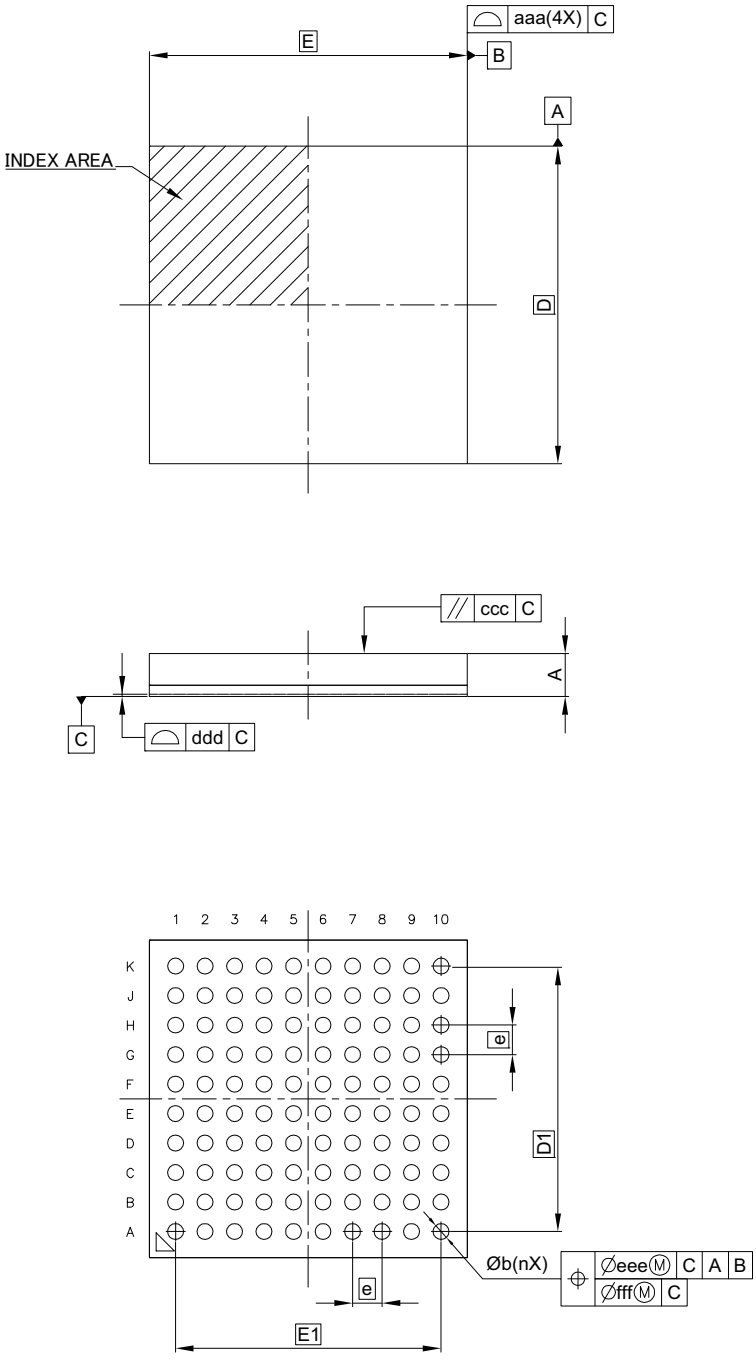
Detail F

NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

Reference Symbol	Dimensions 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	—	0.15
b_p	0.17	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.10
L_p	0.45	0.6	0.75
L_1	—	1.0	—

Figure C 144-Pin LFQFP (PLQP0144KA-B)

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-TFLGA100-7×7-0.65	PTLG0100JB-A	0.09

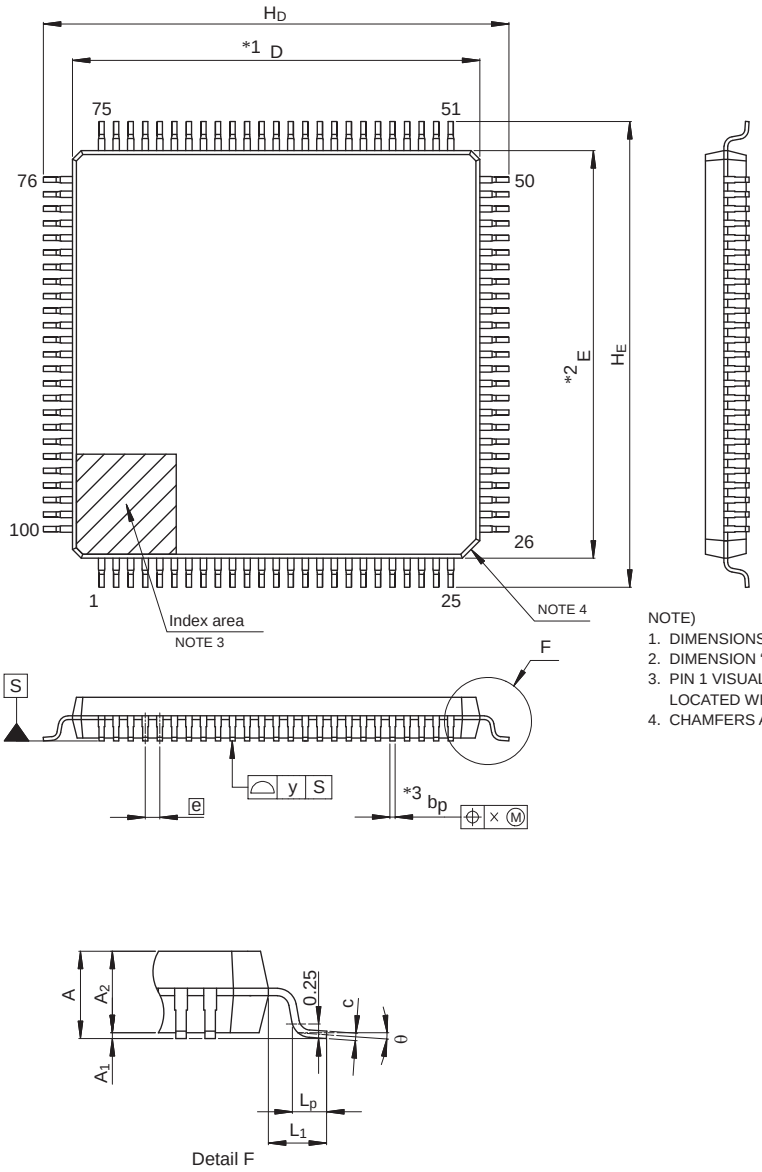


Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	—	7.00	—
E	—	7.00	—
D1	—	5.85	—
E1	—	5.85	—
A	—	—	1.01
b	—	0.35	—
e	—	0.65	—
aaa	—	—	0.10
ccc	—	—	0.10
ddd	—	—	0.08
eee	—	—	0.08
fff	—	—	0.08
n	—	100	—

Figure D 100-Pin TFLGA (PTLG0100JB-A)

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP100-14x14-0.50	PLQP0100KB-B	—	0.6

Unit: mm

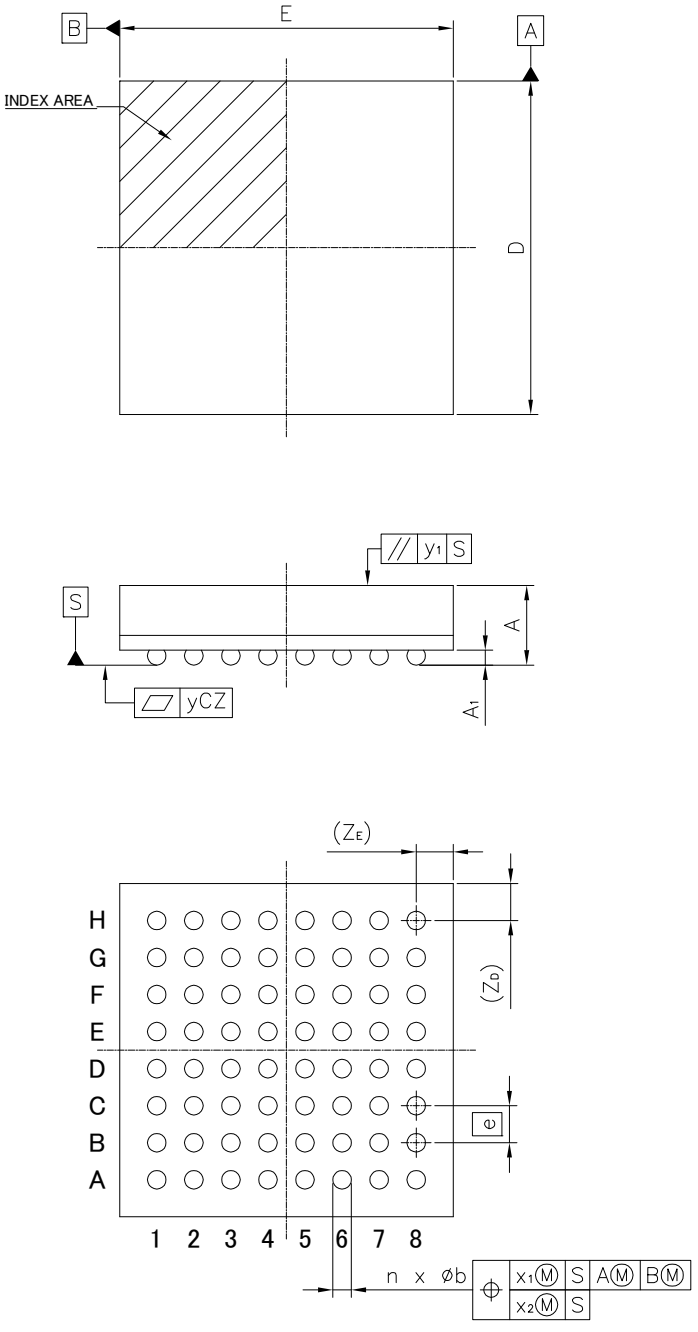


NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

Reference Symbol	Dimensions 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	—	0.15
b_p	0.15	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
L_p	0.45	0.6	0.75
L_1	—	1.0	—

Figure E 100-Pin LFQFP (PLQP0100KB-B)

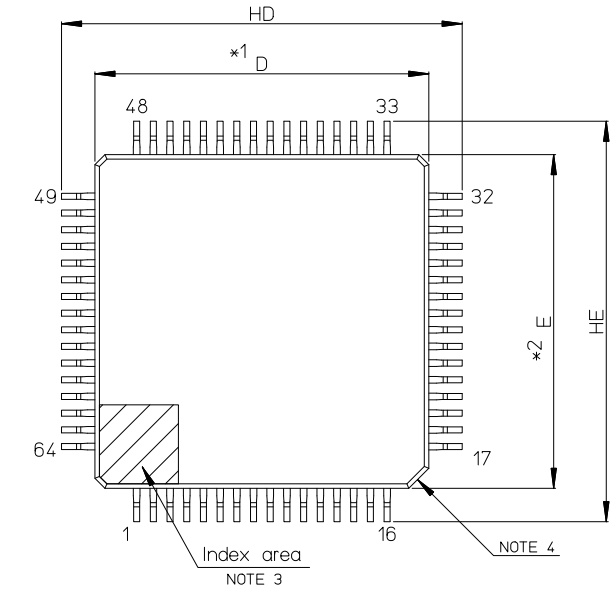
JEITA Package Code	RENESAS Code	MASS(Typ.)[g]
P-TFBGA64-4.5×4.5-0.50	PTBG0064KB-A	0.04

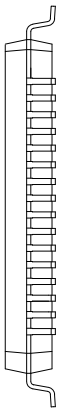


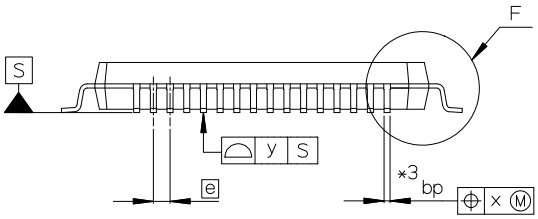
Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	4.42	4.50	4.58
E	4.42	4.50	4.58
A	—	—	1.20
A_1	0.15	0.20	0.25
e	—	0.50	—
b	0.20	0.25	0.30
x_1	—	—	0.15
x_2	—	—	0.05
y	—	—	0.08
y_1	—	—	0.20
n	—	64	—
Z_D	—	0.50	—
Z_E	—	0.50	—

Figure F 64-Pin TFBGA (PTBG0064KB-A)

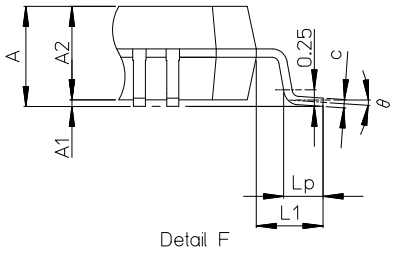
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP64-10x10-0.50	PLQP0064KB-C	——	0.3g







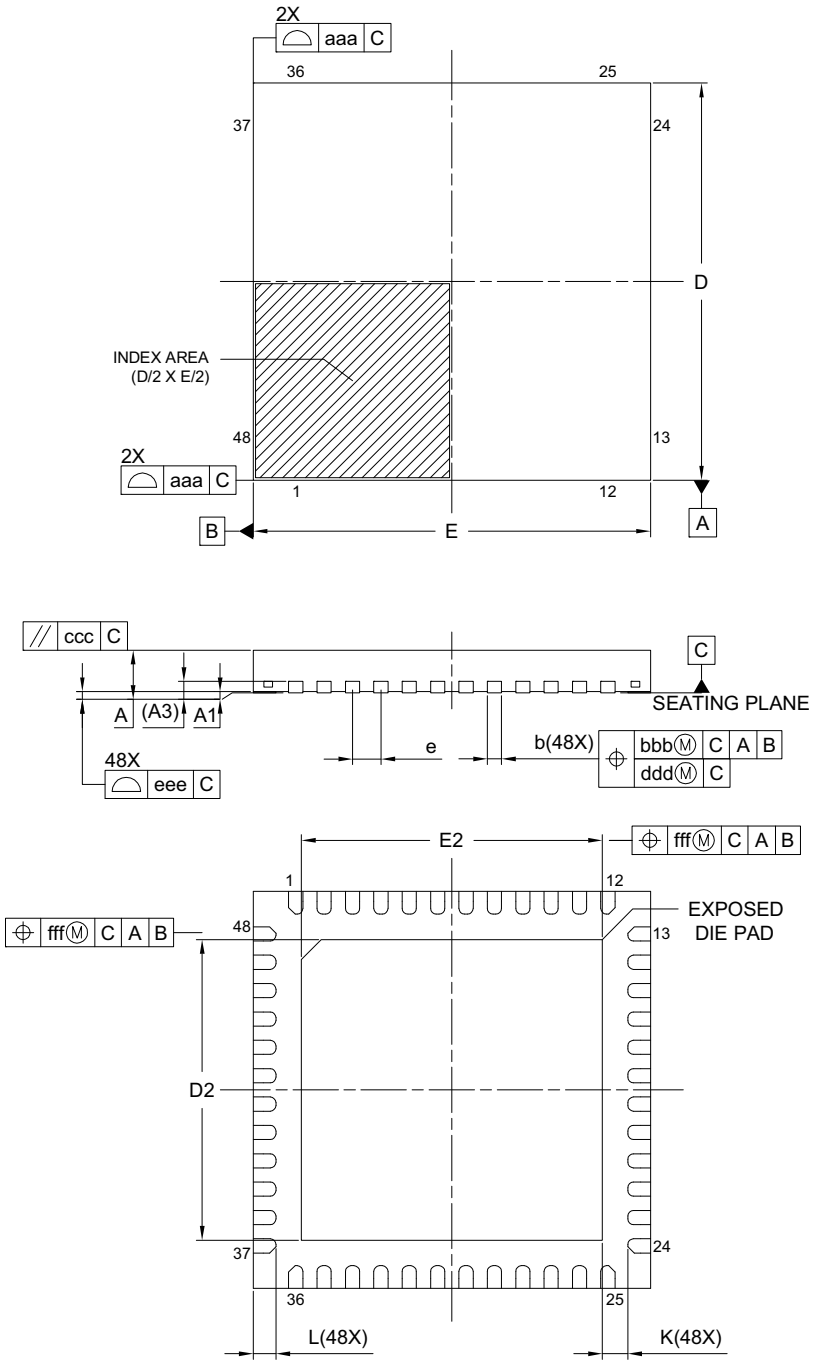
NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL; SIZE MAY VARY.



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	9.9	10.0	10.1
E	9.9	10.0	10.1
A2	——	1.4	——
HD	11.8	12.0	12.2
HE	11.8	12.0	12.2
A	——	——	1.7
A1	0.05	——	0.15
bp	0.15	0.20	0.27
c	0.09	——	0.20
θ	0°	3.5°	8°
e	——	0.5	——
x	——	——	0.08
y	——	——	0.08
Lp	0.45	0.6	0.75
L1	——	1.0	——

Figure G 64-Pin LFQFP (PLQP0064KB-C)

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN048-7x7-0.50	PWQN0048KC-A	0.13 g



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	0.02	0.05
A_3		0.203 REF.	
b	0.20	0.25	0.30
D		7.00 BSC	
E		7.00 BSC	
e		0.50 BSC	
L	0.30	0.40	0.50
K	0.20	—	—
D_2	5.25	5.30	5.35
E_2	5.25	5.30	5.35
aaa		0.15	
bbb		0.10	
ccc		0.10	
ddd		0.05	
eee		0.08	
fff		0.10	

Figure H 48-Pin HWQFN (PWQN0048KC-A)

REVISION HISTORY	RX671 Group User's Manual: Hardware

Classifications

- Items with Technical Update document number: Changes according to the corresponding issued Technical Update
- Items without Technical Update document number: Minor changes that do not require Technical Update to be issued

Rev.	Date	Description		Classification
		Page	Summary	
1.00	Mar 31, 2021	—	First edition, issued	
1.10	Apr 15, 2022	Features		
		69	Coremark value, changed	
		69	Useful functions for IEC60730 compliance, changed	
		1. Overview		
		76	Table 1.1 Outline of Specifications (7/10), changed	
		79	Table 1.1 Outline of Specifications (10/10), changed	
		82 to 84	Table 1.3 List of Products, changed	
		85	Figure 1.1 How to Read the Product Part Number, changed	
		86	Figure 1.2 Block Diagram, changed	
		91	Table 1.4 Pin Functions (5/8), changed	
		92	Table 1.4 Pin Functions (6/8), changed	
		94	Table 1.4 Pin Functions, Note changed	
		7. Option-Setting Memory (OFSM)		
		—	7.5.2 Setting Data for Programming the Option-Setting Memory, deleted	
		9. Clock Generation Circuit		
		319	9.2.5 PLL Control Register (PLLCR), changed	
		341	9.2.24 Sub-Clock Oscillator Control Register 2 (SOSCCR2), changed	
		343	9.2.26 High-Speed On-Chip Oscillator Trimming Registers n (HOCOTRRn) (n = 0 to 2), changed, Note added	
		360	Figure 9.15 Example Flowchart of Initialization when the Sub-Clock is to be Used as the Source to Drive Counting by the Realtime Clock, changed	TN-RX*-A0257A/E
		361	Figure 9.16 Example Flowchart for Initialization when the Sub-Clock is to be Used only as the System Clock, changed	
		362	Figure 9.17 Example Flowchart for when the Sub-Clock is not to be Used, Note 1 added	
		15. Interrupt Controller (ICUE)		
		453, 454	15.2.1 Interrupt Request Register n (IRn) (n = 016 to 255), changed	
		23. Multi-Function Pin Controller (MPC)		
		915	23.2.26 External Bus Control Register 1 (PFBCR1), changed	
		36. Serial Communications Interface (RSCI)		
		1770	Table 36.9 Relationship between N Setting in BRR[7:0] Bits and Bit Rate B, Explanatory notes, changed	
		1805 to 1810	36.2.17 Status Register (SSR), changed	
		38. High-Speed I^{2}C-bus Interface (RIICHS)		
		2085	38.2.22 Communication Status Interrupt Enable Register (ICCSIER), changed	
		2120	38.7.4 Hs-Mode Master Code Detection, added	
		2139	38.13 Interrupt Sources, added	
		2140	38.14 Event Link Function, added	
		55. Flash Memory (FLASH)		
		2844	Table 55.7 Methods of Programming, changed	
		2845	Table 55.8 List of Basic Functions, changed	
		2846	55.7.3 Security Functions, changed	
		2846	Table 55.9 Lists of Security Functions, changed	
		2932	55.17.1 Overview, changed	
		56. Electrical Characteristics		
		2949	Table 56.9 Normal Output Characteristics, added	
		2950	Table 56.10 Thermal Resistance Value (Reference), changed	

Rev.	Date	Description		Classification
		Page	Summary	
1.10	Apr 15, 2022	Appendix 2. Package Dimensions		
		3035	Figure A 145-Pin TFLGA (PTLG0145JC-A), added	
		3036	Figure B 145-Pin TFLGA (PTLG0145KB-A), added	
		3038	Figure D 100-Pin TFLGA (PTLG0100JB-A), added	
		3040	Figure F 64-Pin TFBGA (PTBG0064KB-A), added	
		3042	Figure H 48-Pin HWQFN (PWQN0048KC-A), added	

RX671 Group User’s Manual: Hardware

Publication Date: Rev.1.00 Mar 31, 2021
Rev.1.10 Apr 15, 2022

Published by: Renesas Electronics Corporation

RX671 Group

Renesas Electronics Corporation

R01UH0899EJ0110